Networks Project: Numerical analysis of the Barabási-Albert Model

Lingyi Hu
CID: 00919977

26th March, 2017

Abstract: In this report, we investigate the degree distribution of the Barabási-Albert (BA) Model and 2 other related variants, namely the random attachment and preferential attachment through random walks. We derive the degree distribution and the theoretical largest expected degree, k_1, for preferential and random attachment and compare it against numerical simulation. Agreement of the numerical results with the theoretical model is investigated using the Kolmogorov-Smirnov test. The degree distribution of the random walk model is compared with the scaled free behaviour of the BA model and also with the geometric degree distribution of the random attachment model.

Word count: 3013 words in report (excluding front page, figure captions, table captions, acknowledgement and bibliography).

1 Pure preferential attachment

1.1 Degree distribution

1.1.1 Theory

The master equation that describes the evolution of the BA model is given by

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

where k is the total degree of a vertex, $n(k,t)$ is the number of nodes at time t with total degree k, and the probability Π for choosing the existing vertex depends on the model.

In the pure preferential attachment model, we choose an existing edge with $\Pi_{pa} \propto k$, which after normalizing gives $\Pi_{pa} = k/2E(t)$ where $E(t)$ is the number of edges and $2E(t)$ is the normalization constant corresponding to the total degree of the network. Assuming $E(0) = mN(0)$, the number of edges at a given time t is given by $E(t) = mN(t)$, so we get $\Pi = k/2mnN(t)$. Since we are concerned with the degree distribution of the model at large t, we consider the long-time ansatz $n(k,t) \to N(t)p_\infty(k)$.

Substituting these terms into the master equation, we obtain

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

It is clear that $p_\infty(k < m) = 0$, since m edges are added at every stage. So there are 2 cases to consider when solving for the above equation: $k = m$ and $k > m$.

We first consider the case when $k > m$. In this case, $\delta_{k,m} = 0$ and we can rearrange Equation 2 to get

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

To solve this equation, we can substitute in a trial solution of the form

$$f(z) = A\frac{\Gamma(z+1+a)}{\Gamma(z+1+b)} \tag{4}$$

where $\Gamma(z)$ is the Gamma function, which is an extension of the factorial function, with its argument shifted by one, to all real and complex nnumbers except the non-positive integers. Its central property is that

$$\Gamma(z+1) = z\Gamma(z), \ \Gamma(1) = 1. \tag{5}$$

Substituting the trial solution in Equation 4 gives

$$\frac{A\Gamma(z+1+a)}{\Gamma(z+1+b)} \times \frac{\Gamma(z+b)}{A\Gamma(z+a)} \tag{6}$$

which indeed simplifies to give $(z+a)/(z+b)$, using the the property in Equation 5 that $\Gamma(z+a+1)/\Gamma(z+a) = z+a$.

Substituting $a = -1$ and $b = 2$, we get the solution for Equation 3 in terms of A and the Gamma function:

$$p_\infty(k) = A\frac{\Gamma(k)}{\Gamma(k+3)} \tag{7}$$

which simplifies to

$$p_\infty(k) = \frac{A}{k(k+1)(k+2)}. \tag{8}$$

For the second case of $k = m$, Equation 2 becomes

$$p_\infty(m) = \frac{1}{2}[(m+1)p_\infty(m-1) - mp_\infty(m)] + 1. \tag{9}$$

However, we already know that $p_\infty(k < m) = 0$, that is, $p_\infty(m-1) = 0$. Using this, and rearranging Equation 9, we get

$$p_\infty(m) = \frac{2}{m+2}. \tag{10}$$

Substituting $k = m$ and Equation 10 into Equation 8, we get

$$\frac{A}{m(m+1)(m+2)} = \frac{2}{m+2}, \tag{11}$$

giving us the constant A as

$$A = 2m(m+1). \tag{12}$$

For this constant to be physically reasonable, we need to check that the probability satisfies normalization, that is, we need to prove

$$\sum_{k=m}^{\infty} p_\infty(k) = 2m(m+1)\sum_{k=m}^{\infty}\frac{1}{k(k+1)(k+2)} = 1. \tag{13}$$

The term in the summation of Equation 13 can be expanded as a partial fraction:

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

By writing out the first few terms of each summation, we can see that most terms cancel:

$$\begin{array}{llllll}
\frac{1}{2m} - \frac{1}{m+1} + \cancel{\frac{1}{2(m+2)}} & & & \\
\quad + \frac{1}{2(m+1)} - \cancel{\frac{1}{m+2}} + \cancel{\frac{1}{m+3}} & & & \\
\quad\quad + \cancel{\frac{1}{2(m+2)}} - \cancel{\frac{1}{m+3}} + \frac{1}{2(m+4)} & & & \\
\quad\quad\quad + \cancel{\frac{1}{2(m+3)}} - \frac{1}{m+4} + \frac{1}{2(m+5)} & & & \\
\quad\quad\quad\quad + \ \dots & & &
\end{array} \tag{15}$$

and from the remaining terms we get the relation in Equation 13

$$\sum_{k=m}^{\infty} p_\infty(k) = 2m(m+1)\left(\frac{1}{2m} - \frac{1}{m} + \frac{1}{2(m+1)}\right) = 2m(m+1)\frac{1}{2m(m+1)} = 1 \tag{16}$$

Hence, we can confirm that the complete exact solution for the probability distribution in the long time limit is

$$p_\infty(k) = \frac{2m(m+1)}{k(k+1)(k+2)}. \tag{17}$$

1.1.2 Numerical analysis

To leverage its speed, `c++` code was used to generate graph data, while `python` was used for data analysis due to its wide range of data analysis tools, such as `numpy`, `scipy`, and `pandas`.

There were two main concerns when doing the numerical simulation:

1. What should the initial graph G_0 be?

2. Should self loops and multiple edges be allowed?

3. Of what order should m and N be respectively?

The initial graph was

Computational efficiency of our algorithm is important considering that bigger datasets give better and more reliable statistical results. The following algorithm was used In this algorithm, the array M holds the list of edges represented by pairs of vertices, for example, the vertices at $M[0]$ and $M[1]$ are connected, $M[2]$ and $M[3]$ are connected, and so on. In this list, the number of occurences of a vertex is equal to its degree, so it can be used as a sample pool to achieve preferential attachment. To choose m neighbours for each new vertex, we then sample from M. This is also equivalent to choosing an edge at random and then choosing a vertex at random from the edge.

Algorithm 1 Algorithm for preferential attachment

```
Require: number of vertices $N$, minimum degree $m$                ▷ $N > m$
Initialize our graph $g$
Initialize $M$ as an empty array of length $2Nm$
for $v$ in [0, ..., n-1] do
    g.addVertex()                                               ▷ add new vertex to $g$
    for $i$ in [0, ..., m-1] do
        $M[2(vm + i)] \leftarrow v$
        draw $r$ uniformly at random
        from $[0, ..., 2(vm + i)]$                              ▷ Choose random vertex from $M$
        $M[2(vm + i) + 1] \leftarrow M[r]$                      ▷ Add edge between vertex $M[r]$ and $v$

for $i$ in [0, ..., nm-1] do                    ▷ Add all edges stored in M into the graph
    Add edge $(M[2i], M[2i + 1])$ to graph $g$
```

Clearly this approach produces self loops and multiple edges, especially at the beginning before m vertices have been created. However, since we are concerned with the limit of $N \to \infty$, these effects are insignificant. Also, while unsatisfactory, multiple edges and self-loops do not affect our theoretical result. In the large N limit, the probability of getting multiple edges or self loops is small, and so for simplicity this algorithm was used without modification.

To check that the model was implemented correctly, the degree distribution generated by the model compared checked against the `networkx` implementation of the BA model.

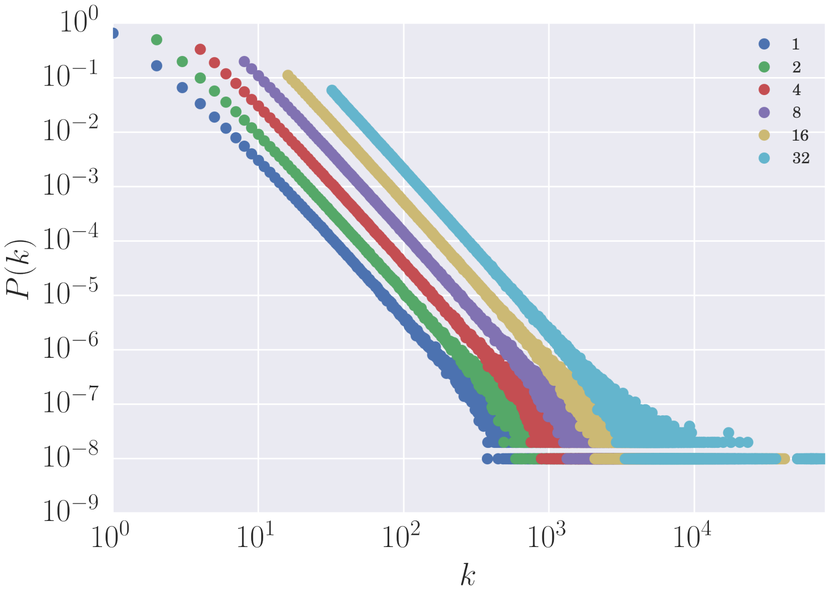

Figure 1: Raw degree distribution averaged over 100 runs for $m = 1, 2, 4, 8, 16, 32$. A noisy tail can still be seen at large k due to finite sized effects.

By using the same random seed, we could check that degree distribution between the two models were exactly the same. Graphs of fewer than 10 nodes were also generated and visualized to ensure that the points followed some basic constraints. This gives confidence that the algorithm is working as expected.

To investigate the degree distribution, the model was run for fixed N but varying m. Several numerical runs were performed for each set of values to improve statistics. It was found that generally above $N = 10^4$, finite size scaling effects are insignificant (see section 1.2.2). In order to reduce finite size effects, $N = 10^6$ was used.

As N is finite, there is bound to be a noisy tail at large k, where these degrees appeared once. To reduce the noise, 100 simulations were run for a single m and the degree distribution was averaged over all runs. This can be seen in Figure 1. The log-binning technique (*Complexity and criticality*) was adopted to to collapse the noisy data.

Visually, as can be seen from Figure 2, the numerical results seem to agree with the theoretical model after log-binning, until finite sized effects begin to kick in at large k. Alternatively, we can look at the complementary cumulative distribution function (ccdf) to observe the behaviour of the fat tail more clearly. This is shown in Figure 3. We can see that near the fat tail, the numerical ccdf goes slightly higher than the theoretical, before falling off. This is consistent across different m values, and the first bump increases in size with increasing m.

A Kolmogorov-Smirnov (KS) test was used to quantify the goodness-of-fit between theory and numerical results. It is a non-parametric test that measures how far apart two distributions are, and it returns a KS-statistic that can be converted into a p-value. A small p-value implies that the model is unlikely to have been the generating function for the data and should be rejected, while a large p-value implies that the differences between data and model may be attributed to statistical differences.

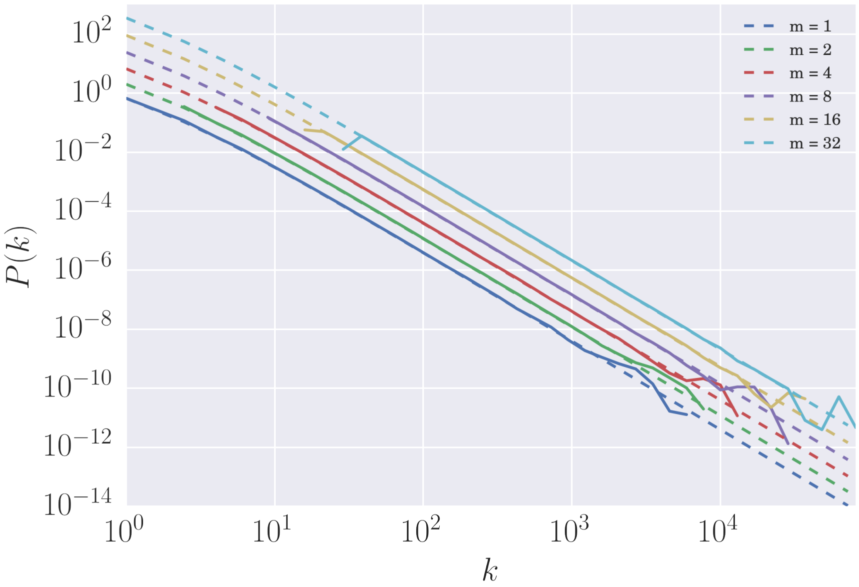

Figure 2: The solid lines show log-binned degree distributions for $m = 1, 2, 4, 8, 16, 32$. The dashed lines show the values predicted by the theoretical model. There is good agreement for small k finite sized effects kick in.

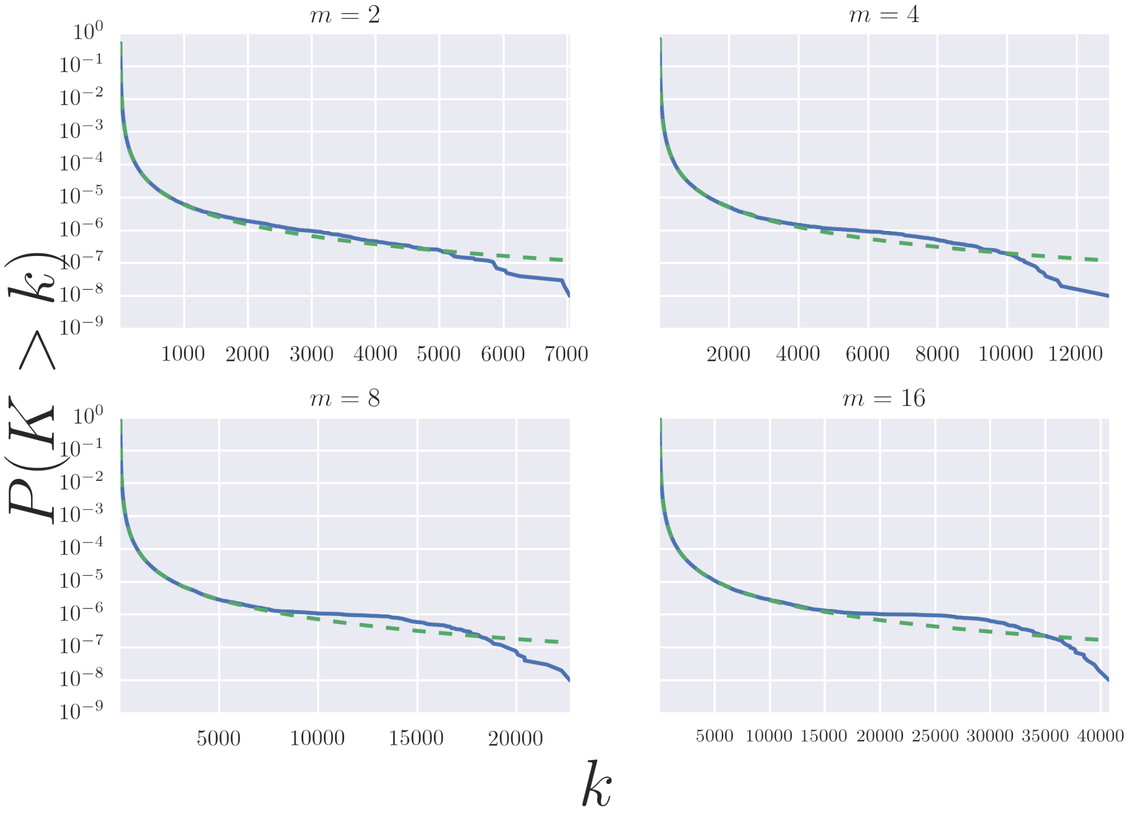

Figure 3: CCDF of $m = 2, 4, 8, 16$, where the solid line is the numerical CCDF and the dotted line is the CCDF predicted by theory.

The table below shows the results of the KS test on $m = 1$ to $m = 32$:

m	KS-statistic
1	0.000414
2	0.000978
4	0.000744
8	0.000662
16	0.001002
32	0.001132

The KS-statistic measures the distance between the numerical data and theoretical distribution, and a smaller value implies that the numerical data and theoretical distribution are more similar. The KS-statistic values can be easily converted to a p-value. However, these values are meaningless unless we know what kind of deviation from theoretical is considered acceptable, and at what value we should reject the null hypothesis. To answer that, we generate synthetic datasets governed by the theoretical distribution in Equation 17 to measure how far they fluctuate from the reference theoretical distribution in a similar way as described by Clauset et al. (2009), and compare the results with the simulated data.

The synthetic datasets were generated with a lower bound of m and an upper bound given by the maximum degree in the simulated dataset. We can then measure how far the synthetic datasets deviate from the theoretical distribution. If the simulated data is much further from the theoretical distribution than the typical synthetic data, then we would have grounds to reject the null hypothesis.

For each of the synthetic datasets, we compare it to the theoretical distribution and calculate its KS-statistic. Then we define our p-value as the fraction of the time the resulting statistic is larger than the value for the numerical data. We can then reject the null hypothesis if $p \leq 0.1$ (Clauset et al., 2009), that is, if there is a 1 in 10 probability or less that we would, by chance, get data that agree as poorly with the model as the current data.

Another issue to consider is the number of synthetic datasets to generate. Again, Clauset et al. (2009) suggests a useful rule of thumb: to have p-values accurate to within about ϵ, we need at least $\frac{1}{4}\epsilon^{-2}$ datasets. In this project, 100 datasets were generated for each m, to get an accuracy of about 0.05.

The resulting p-values are given in the table below:

m	p-value
1	0.71
2	0.45
4	0.50
8	0.58
16	0.71
32	0.46

Table 1: The list of p-values for each m for preferential attachment when compared with synthetic datasets.

Since the p-values for each m are all larger than 0.1, we can say that it is plausible that the numerical data was drawn from the theoretical distribution.

A significant point to note is that the KS test is used for testing continuous distribution, while our degree distribution is discrete. However, it was assumed that for large N, there will be values spanning a large range of k, hence the change in k can be considered small, and the distribution can be approximated as a continuous distribution.

A chi-squared test was also considered. The chi-squared test is a categorical test, suitable for discrete distributions, however, the test becomes invalid when the observed or expected frequencies for each category is too small, with a typical rule being that the frequencies should be at least 5 (Lawrence, 1997). In our numerical data, there many large values of k that only have a frequency of 1, and hence this test was determined to be not suitable.

1.2 Largest expected degree

1.2.1 Theory

The finite size of the system imposes a structural cutoff on the largest expected degree. For scale free networks, Aiello et al. (2001) defined the maximum degree to be approximately the value above which there is less than one vertex of that degree in the graph on average, that is, $N\sum_{k=k_1}^{\infty} p_\infty(k) = 1$.

Generally, it is shown (Boguñá et al., 2004) that for a scale free network with $p_\infty(k) \propto k^\gamma$, the largest expected degree will be

$$k_1(N) \sim N^{1/(\gamma-1)}. \tag{18}$$

In our case, this can be easily verified. Starting with the equation

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

we can see that this is almost identical to Equation 13, just with a different factor and lower limit. Hence we have

$$2m(m+1)\frac{1}{2k_1(k_1+1)} = \frac{1}{N}. \tag{20}$$

We can then rearrange this to give us an expression for k_1:

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

where the other negative solution is rejected as it is unphysical, and confirming that verifying that $k \propto N^{0.5}$.

1.2.2 Numerical analysis

As can be seen from the Figure 4, the numerical value seems to be consistently slightly lower than the theoretical k_1 values. This is reasonable since numerical simulations are

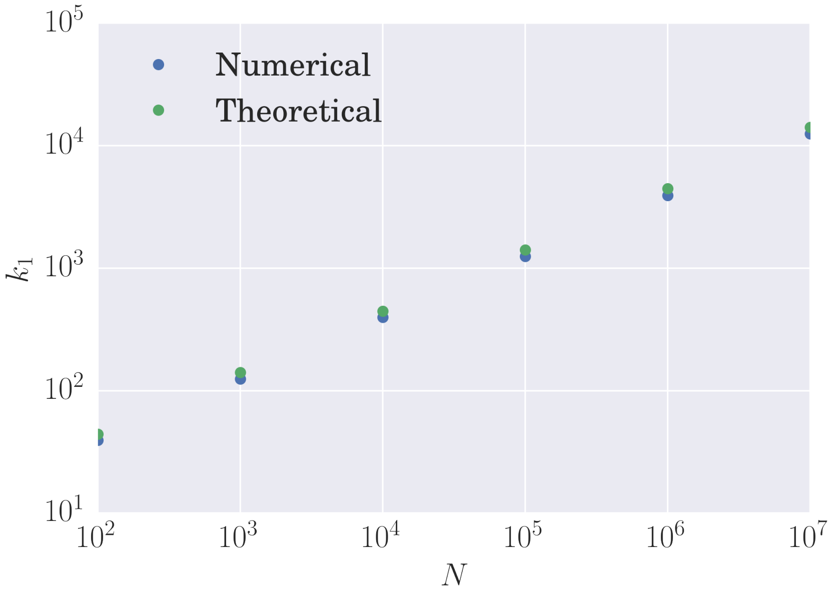

Figure 4: This shows the difference in k_1 for different values of N ranging from 100 to 10^7 for preferential attachment. There is a constant offset of the numerical values from the theoretical, indicating a systematic bias instead of statistical fluctuations.

for finite N, and hence there will be an upper limit to the possible degrees that a vertex can take, while in the theoretical derivation, there is no upper limit to the possible values of k that a vertex can have. By looking at the ratios of k_1(numerical)/k_1(theoretical) as shown in Figure 1.2.2, we can see that deviations are generally constant.

By repeating numerical simulations for 100 times, we can estimate the error on k_1 by calculating the standard deviation for the sample and using the following formula to estimate population standard deviation:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(k_{1,i} - \bar{k}_1)^2} \tag{22}$$

where n is the number of repeats. In this case, $n = 100$.

The values can be seen in Figure 1.2.2,

N	k_1^{theory}	$k_1^{\text{numerical}}$	$k_1^{\text{numerical}}/k_1^{\text{theory}}$
10^2	45	39.3 ± 0.1	0.889
10^3	141	124.9 ± 0.3	0.886
10^4	447	396 ± 1	0.887
10^5	1414	1248 ± 3	0.883
10^6	4472	3924 ± 9	0.878
10^7	14142	12558 ± 30	0.888

Table 2: This table shows the theoretical and numerical values for the largest expected degree as defined in Equation 21 for preferential attachment. The errors on k_1 are rounded off to 1 significant figure.

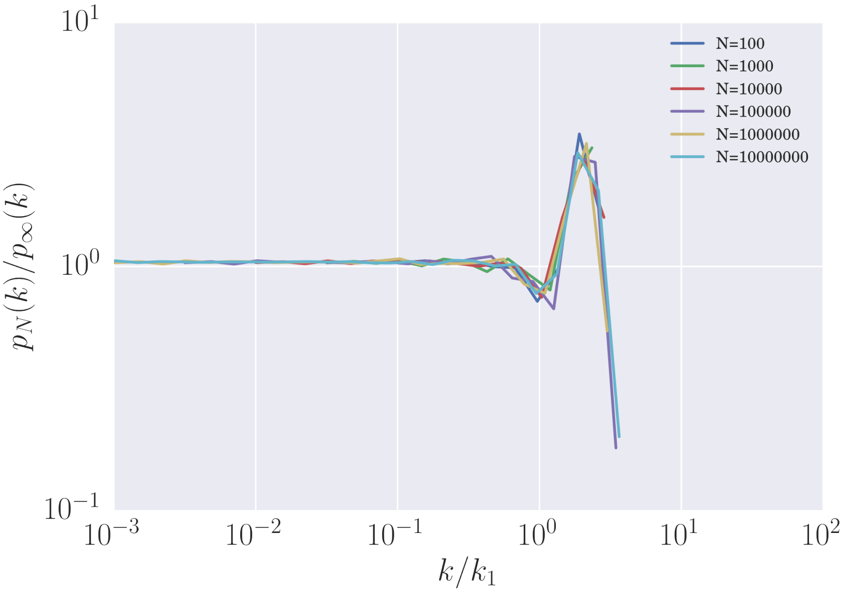

Figure 5: Data collapse of the degree distribution for networks of size $N = 10^2, 10^3, 10^4, 10^5, 10^6, 10^7$

To produce a data collapse, we need to find the function f such that

$$p_N(k) = f(k)\mathcal{G}\left(k/k_1\right) \tag{23}$$

To find $f(k)$, we know that in the limit of $N \to \infty$, p must have no dependence on L, and in the large N limit, $p_N(k) = f(k)$ which is just $p_\infty(k)$. This implies

$$\frac{p_N(k)}{p_\infty(k)} = G\left(k/k_1\right) \tag{24}$$

which means that plotting $p_N(k)/p_\infty(k)$ against k/k_1 will produce a data collapse, as shown in Figure 5.

2 Pure random attachment

2.1 Degree distribution

2.1.1 Theory

The pure random attachment model can be seen as a limiting case of the BA model. In this model, all existing vertices are chosen with equal probability, i.e. $\Pi = \Pi_{rnd} \propto 1$. This preserves growth but removes preferential attachment.

Similar to the previous section, we start from the master equation in Equation 1, and instead of $\Pi = k/2E(t)$ as in preferential attachment, we use $\Pi_{rnd} = 1/N(t)$. Again, we consider the long-time ansatz $n(k,t) \to N(t)p_\infty(k)$. Substituting these terms into Equation 1, we have

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

Considering the case of $k > m$, we obtain the recurrence relation

$$p_\infty(k) = \left(\frac{m}{m+1}\right) p_\infty(k-1) = ... = \left(\frac{m}{m+1}\right)^{k-m} p_\infty(m) \tag{26}$$

Now we consider $k = m$. Substituting $k = m$ into Equation 25 and remembering that $p_\infty(k < m) = 0$, we get

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

giving us

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

Combining this result with Equation 26, we get the following formula for $p_\infty(k)$:

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

For normalization, we need to check that

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

The terms in the summation form a converging geometric series, with the starting term being zero and common ratio being $m/(m+1)$. Hence we have

$$\sum_{k=m}^{\infty}\left(\frac{m}{m+1}\right)^{k-m} = \frac{1}{1-[m/(m+1)]}. \tag{31}$$

By substituting this back into the Equation 30, we can see that normalization is satisfied.

As we can see from Equation 29, the resulting degree distribution in this limit is geometric (Peköz et al., 2013), indicating that growth alone is not sufficient to produce a scale free structure.

2.1.2 Numerical analysis

Numerical simulations confirmed that growth alone is not sufficient to produce a scale free structure. Figure 6 shows the raw degree distribution of numerical simulations for $N = 10^6$ and different values of m. The fat tail was reduced by taking the average of multiple simulations. Already, we can see that it does not follow a power law. After log binning, it follows the theoretical geometric simulation very well for small k, as can be seen in Figure 7.

Goodness of fit was tested in the same way as in the previous section. When tested against the null hypothesis of a geometric distribution governed by Equation 29 p-values obtained are listed below, showing that it is plausible that the simulated data follow the theoretical distribution. While the p-values fluctuate between 0.4 and 0.7, they are all safely above the threshold of 0.1.

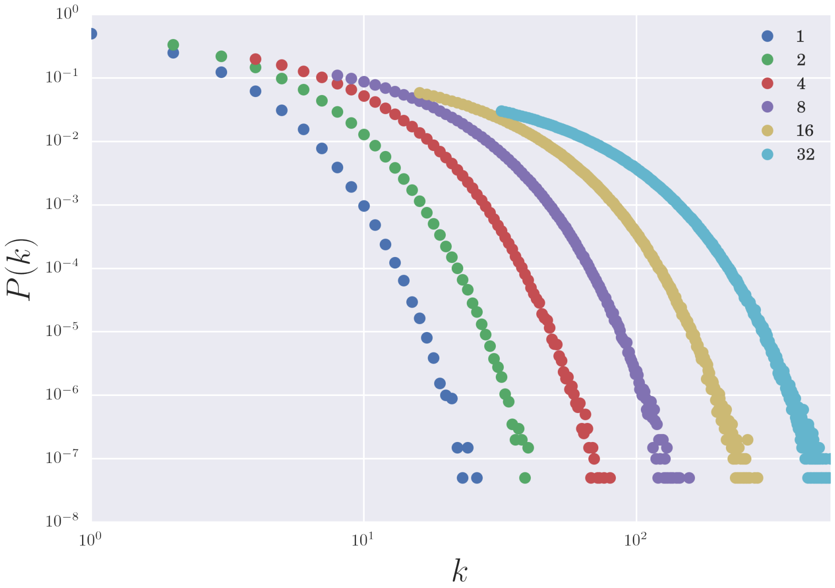

Figure 6: Raw degree distribution for random attachment for $N = 10^6$ and $m = 1, 2, 4, 8, 16, 32$.

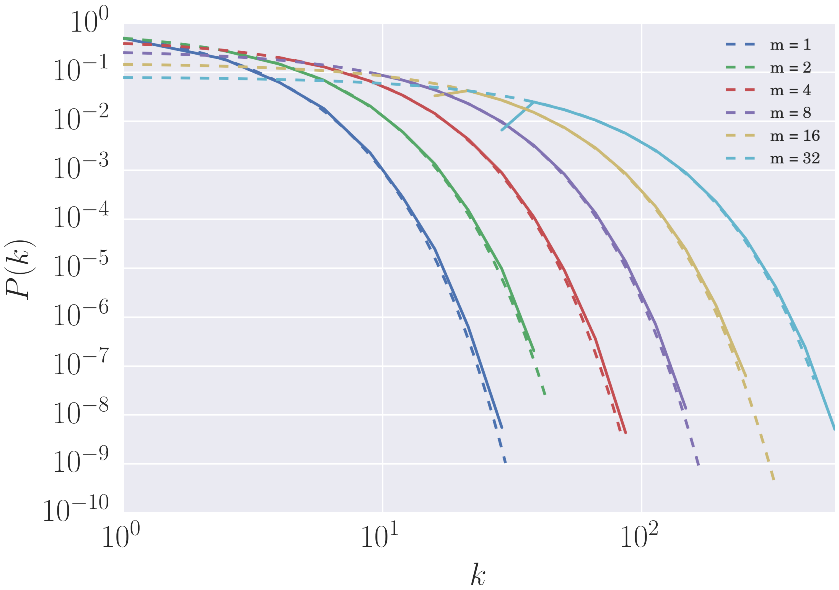

Figure 7: Data collapse of the degree distribution for networks of size $N = 10^2, 10^3, 10^4, 10^5, 10^6, 10^7$

m	p-value
1	0.714
2	0.446
4	0.502
8	0.706
16	0.456

Table 3: The list of p-values for each m for random attachment when compared with synthetic datasets.

To demonstrate whether this test is effective, random attachment simulation data was also tested against the power law distribution, and we obtained result of $p = 0$, to an accuracy of ± 0.05, for all m. This shows that we can reject the power law hypothesis as expected.

2.2 Largest expected degree

2.2.1 Theory

Using the same definition from the previous section for largest expected degree, we need

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

Again, the summation is similar to the previous Equation 31 just with a different lower limit. Applying the geometric series summation formula to the terms in the summation, we get

$$\frac{N}{m+1} \left(\frac{m}{m+1} \right)^{k_1 - m} \frac{1}{1 - (m/(m+1))} = 1 \tag{33}$$

Rearranging this and taking logarithm of both sides, we obtain an expression for k_1:

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

From this, we know that the largest degree grow logarithmically with N, as opposed to a power law like for preferential attachment.

2.2.2 Numerical analysis

The numerical largest degree was calculated in the same way as described in the previous section. Figure ?? shows the discrepancy between the numerical and theoretical largest degree. The ratio between k_1^{theory} and $k_1^{\text{numerical}}$ is largely constant. From Figure 8 we can see that the difference between the numerical and theoretical values decrease as N increases, which is expected.

From the following tables of values, we can also see that the values get closer to 1 for larger N. Similarly, the standard error was estimated from the sample standard deviation, governed by Equation 22.

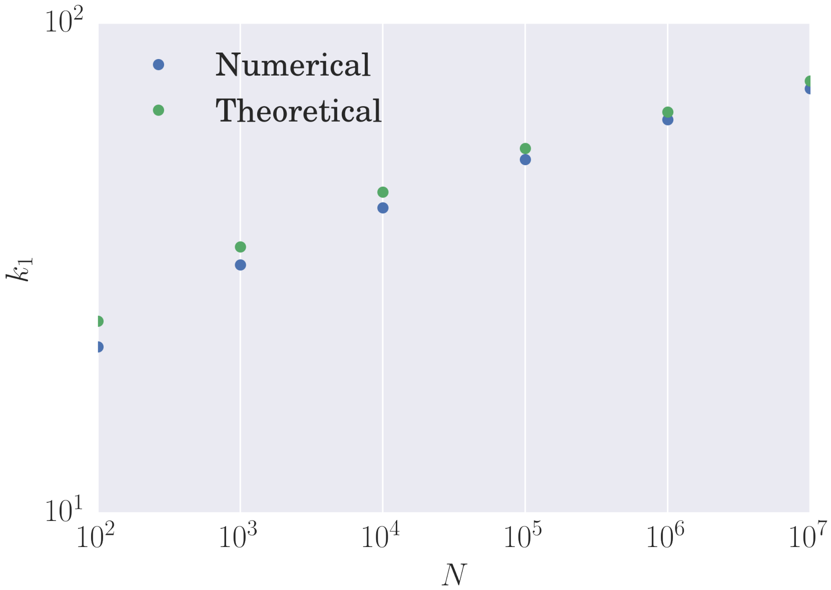

Figure 8: This shows the difference in k_1 for different values of N ranging from 100 to 10^7 for random attachment. From the graph we can see that the difference between the theoretical and numerical values decrease as N increases.

N	k_1^{theory}	$k_1^{\text{numerical}}$	$k_1^{\text{numerical}}/k_1^{\text{theory}}$
10^2	25	21.85 ± 0.01	0.887
10^3	35	32.15 ± 0.03	0.920
10^4	46	42.05 ± 0.03	0.929
10^5	56	52.75 ± 0.02	0.949
10^6	66	63.55 ± 0.02	0.964
10^7	76	73.55 ± 0.03	0.965

Table 4: This table shows the theoretical and numerical values for the largest expected degree as defined in Equation 34 for random attachment. The errors on k_1 are rounded off to 1 significant figure.

3 Random walks and preferential attachment

3.1 Theoretical degree distribution

One of the weaknesses of the BA model and its generalizations is that this implicitly requires a knowledge of the total degree and a calculation across existing vertices on the graph. This requirements then destroys the potential for this model to exhibit emergent properties based on local behaviour. In real-world networks, such as social networks or webpages, the new 'vertices' that join rarely have a global knowledge of the other network vertices. The attachment by performing a random walk is a solution proposed by Saramäki and Kaski (2004). In this model, a vertex is chosen at random from existing vertices and then executes a random walk of length L from that vertex. The new vertex then attaches to the destination vertex.

Preferential attachment then follows from the fact that the random walker is more likely to end up at a more highly connected vertex. This models real-world models such as interconnected webpages better, since we are likely to click on links to webpages from webpages we are already visiting.

This model was thought to be able to reproduce the BA degree distribution even for $L = 1$ (Saramäki and Kaski, 2004; J.P.Saramaki and T.S.Evans, 2004). While this is the case for large L, Cannings and Jordan (2013) later showed that the $L = 1$ and $m = 1$ degree sequence converges to a degenerate limiting solution in which almost every vertex has degree 1, instead of a power law distribution, and demonstrated that this model is fundamentally different from the BA model, unless we allow an indefinite length for the random walk. For $L = 0$, this reduces to the random attachment model.

3.2 Numerical simulation

A few considerations were taken into account when writing the numerical simulation for the random walk:

1. We are adding m edges to each new vertex. Should we reset the random walk to find each new destination vertex, or should we continue an L-step random walk from the current destination vertex to the next destination vertex till we get m edges?

2. Is the walker allowed to trace his steps backwards, that is, is the walk self-avoiding?

For point 1, we followed the convention in Saramäki and Kaski (2004) where we continue the random walk from the previous destination vertex, until we reach m destination vertices, that is, the random walk only resets when a new vertex is added. For point 2, it was decided that the walk will not be self-avoiding, otherwise it would get stuck easily.

For $L = 0$, the degree distribution reduces to that of random attachment, as expected.

The raw degree distribution generated by $L = 1$ is shown in Figure 9, with the dotted lines showing the theoretical BA distribution for the same m. We can see quite clearly that the case of $m = 1$ does not follow a power law distribution of preferential attachment.

To test whether it follows a power law distribution, we can use the same technique in section 1.1.2, that is, compare its KS-statistic against those of synthetic data, measured against a reference theoretical distribution. This was first done for $L = 1$, $N = 10^5$, and various values of m, and the following p-values were obtained:

m	p-value
1	0.00
2	0.00
4	0.029
8	0.768
16	0.648

As we can see from the table, for small m. Indeed, this was verified by Cannings and Jordan (2013) who showed that in this case, the degree distribution converges to

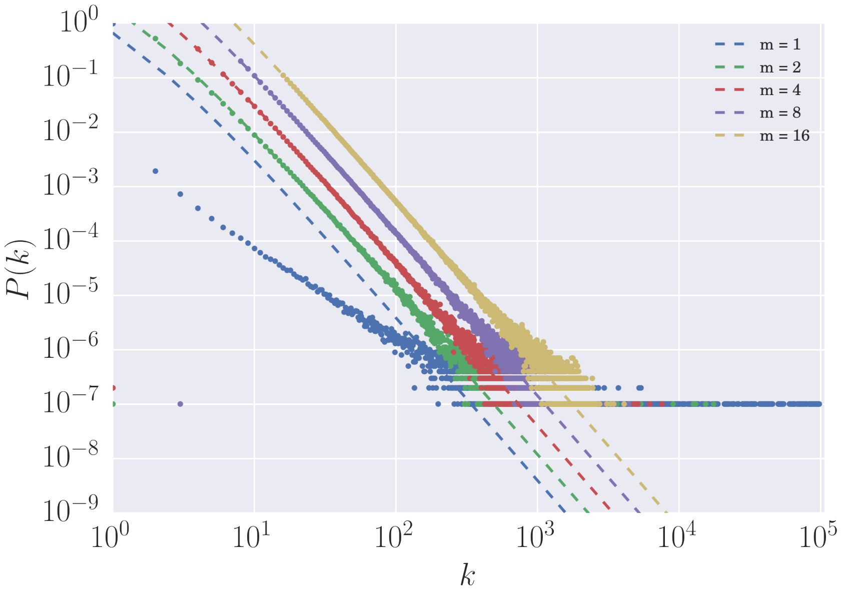

Figure 9: Degree distribution of the random walk model for $N = 10^5$, $L = 1$, and varying m

a degenerate limiting solution in which almost every vertex has degree 1. At $m = 8$, however, it is already highly plausible that the degree distribution follows that of the preferential attachment model.

Numerical simulations were done for fixed N and fixed m but varying L from 1 to 8, with the m chosen to be 8 from the previous results and $N = 10^5$. The same technique was used to test whether it follows a power law distribution, and we obtain the result of $p = 0.64$ for all L. This shows that it is plausible that the degree distribution for the random walk attachment follows a power law, and this holds regardless of L.

References

Aiello, William et al. (2001). "A Random Graph Model for Power Law Graphs". In: *Experimental Mathematics* 10.1, pp. 53–66. ISSN: 1058-6458. DOI: 10.1080/10586458.2001.10504428. URL: http://www.tandfonline.com/doi/abs/10.1080/10586458.2001.10504428.

Boguñá, M., R. Pastor-Satorras, and A. Vespignani (2004). "Cut-offs and finite size effects in scale-free networks". In: *European Physical Journal B* 38.2, pp. 205–209. ISSN: 14346028. DOI: 10.1140/epjb/e2004-00038-8. arXiv: 0311650 [cond-mat].

Cannings, Chris and Jonathan Jordan (2013). "Random walk attachment graphs". In: *Electronic Communications in Probability* 18, pp. 1–8. ISSN: 1083589X. DOI: 10.1214/ECP.v18-2518. arXiv: 1303.1052.

Christensen, Kim [author]. *Complexity and criticality.* Ed. by Nicholas R Moloney. Imperial College Press advanced physics texts volume 1. London: Imperial College Press. ISBN: ISBN: 1860945171.

Clauset, Aaron, Cosma Rohilla Shalizi, and M E J Newman (2009). "Power-Law Distributions in Empirical Data". In: *SIAM Review* 51.4, pp. 661–703. ISSN: 19417330. DOI: 10.1214/13-AOAS710. arXiv: arXiv:0706.1062v2.

J.P.Saramaki and T.S.Evans (2004). "Scale Free Networks from Self-Organisation". In: *Physical Review E* 72.2, pp. 1–33. ISSN: 1539-3755. DOI: 10.1103/PhysRevE.72.026138. arXiv: 0411390v2 [cond-mat]. URL: http://link.aps.org/doi/10.1103/PhysRevE.72.026138.

Lawrence, John (1997). *A guide to Chi-squared testing.* DOI: 10.1016/S0378-3758(97)00101-8.

Peköz, Erol A., Adrian Röllin, and Nathan Ross (2013). "Total variation error bounds for geometric approximation". In: *Bernoulli* 19.2, pp. 610–632. ISSN: 1350-7265. DOI: 10.3150/11-BEJ406. URL: http://projecteuclid.org/euclid.bj/1363192040.

Saramäki, Jari and Kimmo Kaski (2004). "Scale-free networks generated by random walkers". In: *Physica A: Statistical Mechanics and its Applications* 341.1-4, pp. 80–86. ISSN: 03784371. DOI: 10.1016/j.physa.2004.04.110. arXiv: 0404088 [cond-mat].