Network Project
A Growing Network Model

CID: 01338402

March 24, 2020

Abstract: The degree distributions in growing network models were investigated analytically and numerically. The models considered were preferential attachment (BA model) and pure random attachment models. Both theoretical and numerical results were compared and found to agree with each other reasonably well. The BA model behaves like a power-law with an exponent of -3 for large degrees. Compared with the preferential attachment model, the degree distribution in pure random attachment model decays much more rapidly. In addition, an attachment model involving random walks was also simulated and studied. It was found that the degree distribution depended on the probability of continuing the random walk.

Word Count: 2100 words excluding front page, figure captions, table captions, and bibliography.

1 Introduction

The aim of the project is to investigate the degree distributions in simple models of a growing network. Firstly, the BA model which involves preferential attachment is considered. The degree distribution in the model is studied analytically and numerically. Comparing the theoretical and numerical results might reveal any limitations on the approximations made. A random attachment model is investigated in similar manner. An attachment model involving random walks is simulated numerically and the degree distribution is discussed.

1.1 Definition

The BA model is a growing network model which incorporates a preferential attachment mechanism. In this mechanism, the probability Π for choosing the existing vertex at on end of a new edge is proportional to its degree, k [1].

2 Phase 1: Pure Preferential Attachment Π_{pa}

2.1 Implementation

2.1.1 Numerical Implementation

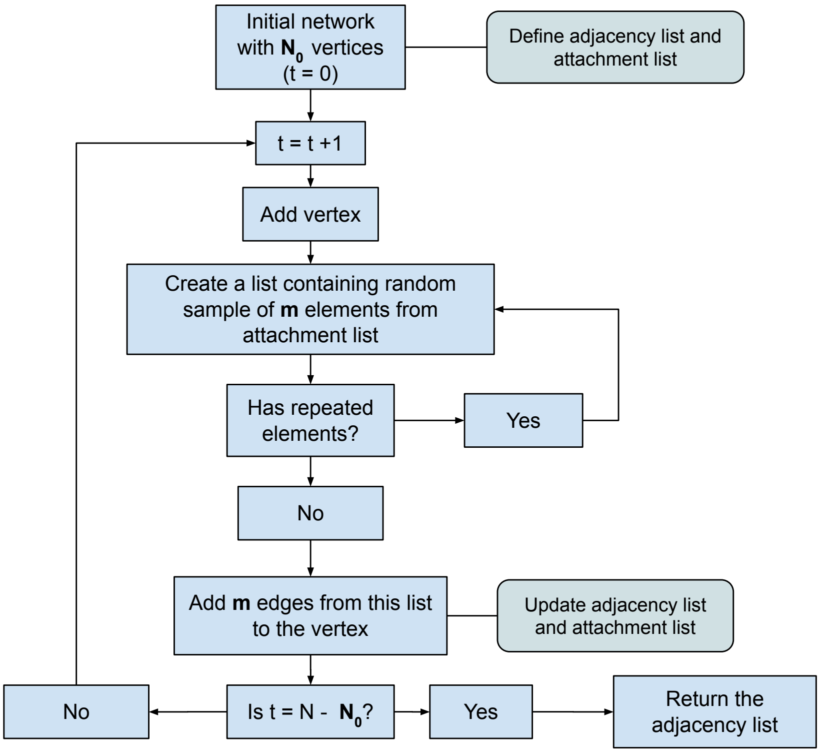

Initial network with $\mathbf{N_0}$ vertices (t = 0)
Define adjacency list and attachment list
t = t +1
Add vertex
Create a list containing random sample of **m** elements from attachment list
Has repeated elements?
Yes
No
Add **m** edges from this list to the vertex
Update adjacency list and attachment list
No
Is t = N - $\mathbf{N_0}$?
Yes
Return the adjacency list

Figure 1: The numerical implementation of BA model. N is the number of vertices needed to be in the final graph.

The numerical implementation of BA model is shown in figure 1. Python language was used. The whole procedure is made into a function to return the adjacency list of the completed network with N vertices and m edges added to each newly added vertex. To ensure the probability of attaching a new edge to the vertex is proportional to its degree, I defined an attachment list for the initial network. Every time an edge is connected to a vertex, the vertex is added to the attachment list. Therefore, the attachment list contains vertices with frequency equal to their number of degrees. Hence, this satisfies the probability condition.

2.1.2 Initial Graph

The value of m must be smaller or equal to the number of vertices in the initial network, N_0. Hence, I created a complete graph which always has $m + 1$ vertices. The initial network produced was simple, undirected and unweighted. The graph was chosen to be complete since it was easier to produce the adjacency and attachment lists for a complete graph. Figure 2 shows an example of a complete graph used as the initial network.

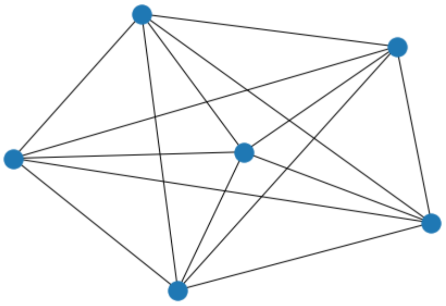


Figure 2: A complete graph with $N_0 = 6$.

2.1.3 Type of Graph

The graph produced is simple, undirected and unweighted by the nature of numerical implementation and the initial network. The graph is also random in a sense there is no order when choosing a vertex even though it follows the preferential attachment. The graph produced is also not multi-edged. Since each vertex in the initial network contains equal number of degrees, any preferential behaviour in the final graph should come from the "growing" process of adding vertices. This behaviour can be seen visually in small networks as shown in figure 3.

2.1.4 Working Code

I checked if there was any repeated neighbours and self-loops for each vertex to avoid multi-edges and there was none as expected. I produced network plots for small N to observe the preferential attachment visually as shown in figure 3. To further validate the code, I calculated the average degree in a simulated graph numerically. Theoretically, average degree is the ratio of total number of degrees to total number of vertices, i.e. $\langle k \rangle = 2E/N$ for an undirected graph. They both agreed to a high precision.

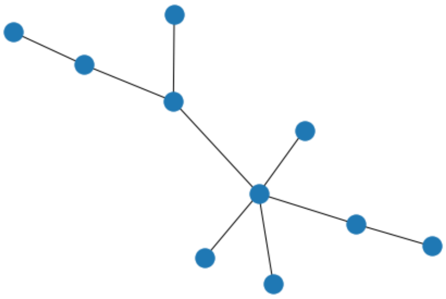


Figure 3: An example of final graph produced for $N = 10$ and $m = 1$. There is one vertex with the largest degree which displays preferential behaviour.

2.1.5 Parameters

The programme needs the number of vertices in the final network, N and the number of edges per vertex added, m. An optional seed number is also needed to initialise different realisation (default is 1). Different realisation is used every time a model is simulated for multiple runs.

2.2 Preferential Attachment Degree Distribution Theory

2.2.1 Theoretical Derivation

The master equation provided on the lectures 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 $n(k,t)$ is the number of vertices with degree k at time t, from [1]. Π here equals to $\Pi_{pa} = \frac{k}{2E}$ where $2E$ is the total number of degrees and E is the number of edges. At long time limit, $t \gg N(t=0)$ and $mt \gg E(t=0)$. Hence,

$$\lim_{t\to\infty} \frac{E(t)}{N(t)} = m. \tag{2}$$

The frequency distribution of degree $n(k,t)$ can be written as

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

where $p_\infty(k)$ is the probability distribution which is assumed to be stable in the long time limit. Substituting eq. (3) and the expression for Π into eq. (1), we have

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

Approximate $N(t) \approx t$ in the long time limit. Using eq. (2) and after some algebra,

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

Considering the case $k > m$ and rewriting eq. (5) to

$$2p_\infty(k) = (k-1)p_\infty(k-1) + kp_\infty(k) \tag{6}$$

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

From the problem sheet, we know

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

has the solution

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

where $\Gamma(n) = (n-1)!$ for $n \in \mathbb{Z}^+$ and A is a constant [2]. Using this result for eq (7), we get

$$p_\infty(k) = \frac{A}{k(k+1)(k+2)} \tag{10}$$

I assumed $\Pi(k,t)n(k,t) = 0$ for $k < m$ in the long time limit [1]. In fact, this is always true in the numerical implementation since the initial network will always have m edges per vertex. Considering this for k = m and using eq. (5), we get

$$p_\infty(m) = \frac{2}{2+m} \tag{11}$$

since $p_\infty(m-1) = 0$ from the assumption above. Equating eq. (10) and eq. (11), and solving for A, we get

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

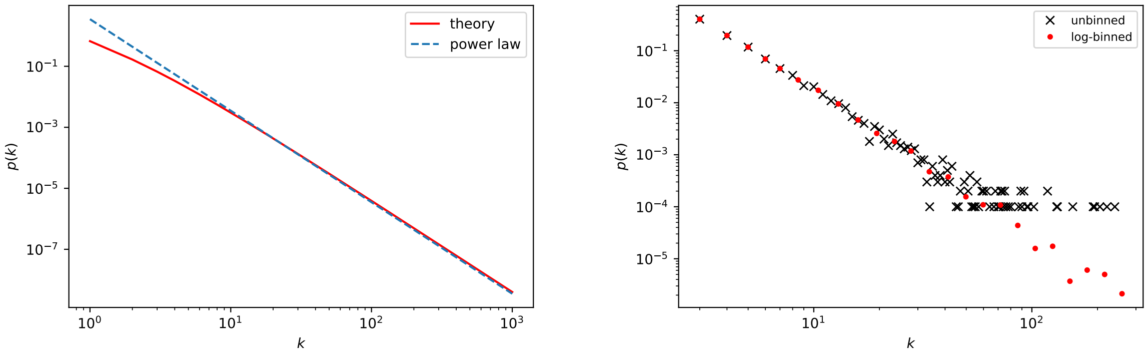
for degree distribution in the long time limit.

2.2.2 Theoretical Checks

Firstly, it was checked if the solution in eq. (12) is normalised. Using m as the lower limit for k,

$$\begin{aligned}
\sum_{k=m}^{\infty} p_\infty(k) &= \sum_{k=m}^{\infty} \frac{2m(m+1)}{k(k+1)(k+2)} && (13)\\
&= m(m+1)\sum_{k=m}^{\infty}\left(\frac{1}{k} - \frac{2}{k+1} + \frac{1}{k+2}\right) && (14)\\
&= m(m+1)\left[\sum_{k=m}^{\infty}\left(\frac{1}{k} - \frac{1}{k+1}\right) - \sum_{k=m}^{\infty}\left(\frac{1}{k+1} - \frac{1}{k+2}\right)\right] && (15)\\
&= m(m+1)\left[\frac{1}{m} - \frac{1}{m+1}\right] && (16)\\
&= 1 && (17)
\end{aligned}$$

From eq. (12), I observed that for $k \gg 1$, $p_\infty(k) \propto k^{-3}$. This agrees with the continuous solution found in the lectures. The solution in eq. (12) was plotted as a function of k for $m = 1$ and is shown in figure 4(a). It also approximates to a power-law with exponent of -3 for large k.



(a)

(b)

Figure 4: (a) Theoretical check: The solution in eq. (12) as a function of k for $m = 1$ on a log-log plot. The blue dashed line corresponds to the power-law with exponent of -3. (b) Fat-tail: Degree distribution for $N = 10000$ and $m = 3$. Raw data is grouped together at the "'"tail" region of the plot. The issue is solved when the data is log-binned with a scale of 1.2.

2.3 Preferential Attachment Degree Distribution Numerics

2.3.1 Fat-Tail

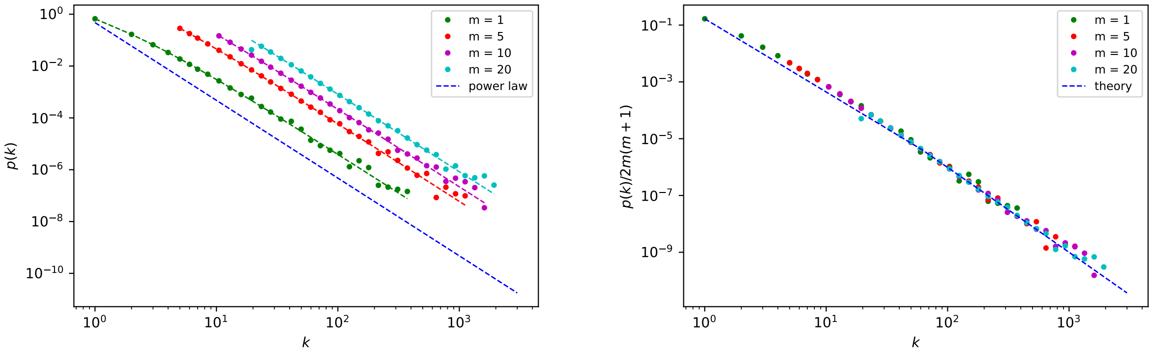
I ran the simulation for $N = 10000$ and $m = 3$ and plotted the degree distribution shown in figure 4(b). There are data points at the "tail" region of the plot that are grouped together. This is because in a fat-tail distribution, there exist a few vertices with very large k and they all represent a very small probability. Their occurrences are too rare that they are represented by only one probability in the plot [1]. To deal with this issue, I used the log binning code given during the complexity project. This gives the probability for a degree in a bin whose size scales exponentially. I chose a scale of 1.2 so the amount of information lost due to binning is minimal. This choice of scale will be applied for the remainder of the project.

2.3.2 Numerical Results

I ran the simulation for $m = 1$, 5, 10 and 20. The N chosen for each m was 100000 to ensure the distribution reaches a long time limit. Figure 5(a) shows the degree distribution for each m. Using the theoretical result in eq. (12), I fitted the theoretical functions for each values of m and plotted on top of the numerical results. Visually, the theoretical functions fit the numerical results very well. It also agrees for small k for $m = 1$. They all behave as a power-law for large k. Each has different vertical offset corresponding to their m values. To validate further, I performed a data collapse using the theoretical eq. (12). Figure 5(b) shows the plot for $p(k)/[2m(m + 1)]$ as a function of k. It agreed to a high degree.

2.3.3 Statistics

I used the Pearson's chi-squared test on the log-binned data to evaluate the goodness of fit of the theoretical results on the numerical ones. This is because I assumed the loss of information for log binned data is minimal. A more accurate method would be the KS-test for unbinned data. However, a chi-squared test still provides a valid result since the sample size is sufficiently large [3]. The chi square values obtained for the fit for each m is given on table 1. They are all small, indicating the theoretical results fit the numerical outcomes well.



(a) (b)

Figure 5: (a) Degree distribution for $N = 100000$ and $m = 1, 5, 10$ and 20. The dashed coloured lines are the theoretical predictions for corresponding values of m. The blue dashed line is the power-law with an exponent of -3 for comparison. (b) Data collapse on m: $p(k)/[2m(m + 1)]$ as a function of k. The blue dashed line is the theoretical prediction from eq. (12).

m	χ^2
1	7.395 $\times 10^{-5}$
5	4.928 $\times 10^{-5}$
10	5.164 $\times 10^{-5}$
20	0.0317

Table 1: The chi-square χ^2 values for the goodness-of-fit tests for each m using the theoretical predictions as null hypothesis, in preferential attachment model. The p-values obtained for all values of m are exactly 1, suggesting the likelihood of numerical observations matching the theory is very high.

2.4 Preferential Attachment Largest Degree and Data Collapse

2.4.1 Largest Degree Theory

I assumed there is only one vertex with the largest number of degree, k_1. Mathematically this is given by

$$\sum_{k=k_1}^{\infty} N p_\infty(k) = 1 \tag{18}$$

Using eq. (15) and eq. (18),

$$\sum_{k=k_1}^{\infty} p_\infty(k) = m(m+1) \left[\sum_{k=k_1}^{\infty} \left(\frac{1}{k} - \frac{1}{k+1} \right) - \sum_{k=k_1}^{\infty} \left(\frac{1}{k+1} - \frac{1}{k+2} \right) \right] \tag{19}$$

$$\frac{1}{N} = m(m+1) \left[\frac{1}{k_1} - \frac{1}{k_1+1} \right] \tag{20}$$

Rearranging gives

$${k_1}^2 + k_1 - Nm(m+1) = 0 \tag{21}$$

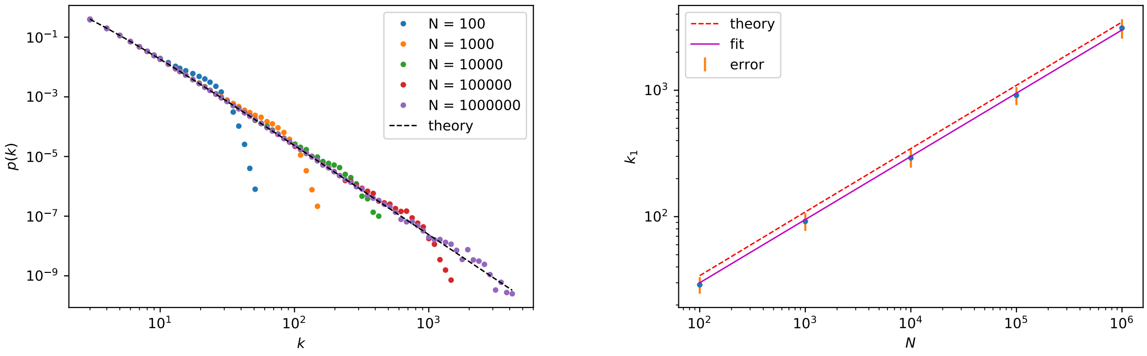
and taking the positive solution, we get

$$k_1 = \frac{-1 + \sqrt{1 + 4Nm(m+1)}}{2} \tag{22}$$

For $N \gg 1$, $k_1 \propto \sqrt{N}$.

2.4.2 Numerical Results for Largest Degree

I ran the simulation for $N = 100$, 1000, 10000, 100000 and 1000000; each with $m = 3$. I chose this value for m because a value larger than 1 represents a more connected graph. Each N is repeated for multiple runs to obtain good statistics; hence this value of m is also reasonable given the limited memory my computer had. The plot in figure 6(a) shows there are finite size effects present. This can be characterised by k_1, the largest degree size for a given N. I found k_1 for each run for each N and grouped them. The mean k_1 are plotted as a function of N in figure 6(b). The plot is fitted, and it gives an exponent of 0.50 ± 0.05 which agrees with eq. (22). However, there is a constant vertical offset for the numerical result from the theoretical prediction. This suggests the assumption, that there is only one degree with the largest size, might not be accurate entirely.



(a) (b)

Figure 6: (a) Degree distribution for $N = 100$ (10000 runs), 1000 (1000 runs), 10000 (100 runs), 100000 (100 runs) and 1000000 (10 runs), with $m = 3$. The black dashed line corresponds to the theoretical prediction for $m = 3$. (b) Largest expected degree k_1 as a function of N. The red dashed line is the theoretical prediction from eq. (22). The standard deviation in the distribution of k_1 was assumed to be its error, which seems to be constant for all N. The magenta coloured solid line is the line of best fit which is $k_1 \approx 3N^{0.50}$.

2.4.3 Data Collapse

Since figure 6(b) indicates how k_1 scales against N, I performed a data collapse on the data. For the horizontal collapse, I divided k by k_1 for each N since the "bump" in figure 6(a) is characterised by k_1. For the vertical collapse, I divided $p(k)$ by $p_\infty(k)$ so I get a horizontal line. In figure 7, it can clearly be seen that $p(k)/p_\infty(k) = 1$ until k approaches k_1, and then it decays rapidly. This signifies the degree distribution is limited by the finite size of the network, which makes sense.

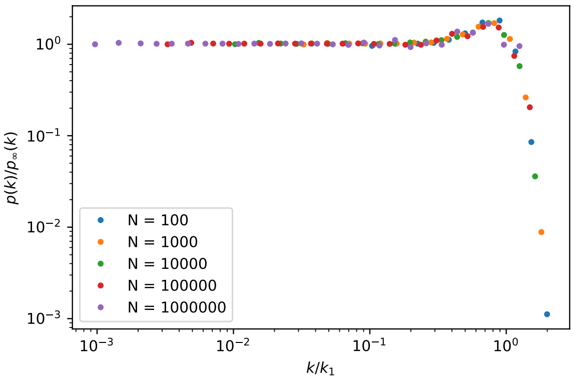


Figure 7: Data collapse for different values of N.

3 Phase 2: Pure Random Attachment Π_{rnd}

3.1 Random Attachment Theoretical Derivations

3.1.1 Degree Distribution Theory

The probability of choosing a vertex in pure random attachment is $\Pi = 1/N(t)$. Substituting this and using the same approximations in the preferential attachment derivation, we get a new master equation,

$$N(t+1)p_\infty(k) = N(t)p_\infty(k) + \frac{m}{N(t)}N(t)p_\infty(k-1) - \frac{m}{N(t)}N(t)p_\infty(k) + \delta_{k,m} \tag{23}$$

Rearranging,

$$(1+m)p_\infty(k) = mp_\infty(k-1) + \delta_{k,m} \tag{24}$$

For $k > m$,

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

and

$$p_\infty(k+1) = \frac{m}{m+1}p_\infty(k) = \left(\frac{m}{m+1}\right)^2 p_\infty(k-1). \tag{26}$$

Therefore, by induction,

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

since m is the lower limit of k. For $k = m$, eq. (24) becomes

$$(1+m)p_\infty(m) = mp_\infty(m-1) + 1 \tag{28}$$

but the first term on RHS is zero since there are no vertices with degrees fewer than m in the long time limit. Thus,

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

Using this in eq. (27), we get

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

As a check, the solution in eq. (30) is shown to be normalised too.

3.1.2 Largest Degree Theory

Using the same assumption as in eq. (18) and eq. (30),

$$\sum_{k=k_1}^{\infty} p_\infty(k) = \sum_{k=k_1}^{\infty} \frac{m^{k-m}}{(m+1)^{k-m+1}} = \frac{1}{N} \tag{31}$$

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

Rearranging and changing the index on the sum on RHS,

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

Using geometric sum on RHS,

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

Taking the log of both sides and rearranging for k_1 gives

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

For $N \gg 1$, $k_1 \propto \ln N$ in pure random attachment mechanism.

3.2 Random Attachment Numerical Results

3.2.1 Degree Distribution Numerical Results

As in the preferential attachment, I ran the simulation for m = 1, 5, 10 and 20, each with N = 100000. Figure 8 shows the resulting degree distribution. The theoretical equation for degree distribution in eq. (30) was plotted on top of the numerical results. Compared to preferential attachment, the degree distribution in random attachment model decays exponentially, i.e. more rapidly. The statistics test result is shown in table 2. Both visually and statistically, theoretical and numerical results agree to a high degree.

m	χ^2
1	1.695×10^{-4}
5	7.095×10^{-5}
10	6.018×10^{-5}
20	0.0125

Table 2: The chi-square χ^2 values for the goodness-of-fit tests for each m using the theoretical predictions as null hypothesis, in random attachment model. The p-values obtained for all values of m are exactly 1, suggesting the likelihood of numerical observations matching the theory is very high.

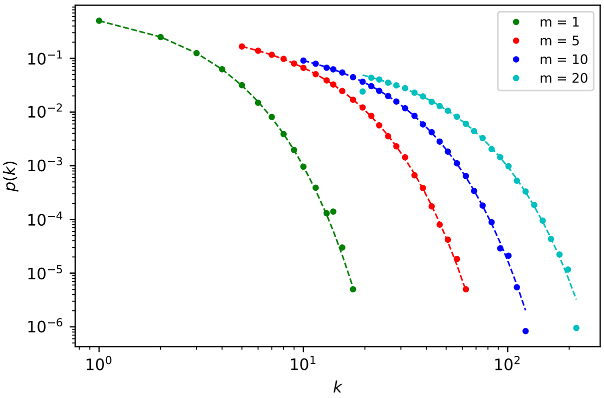
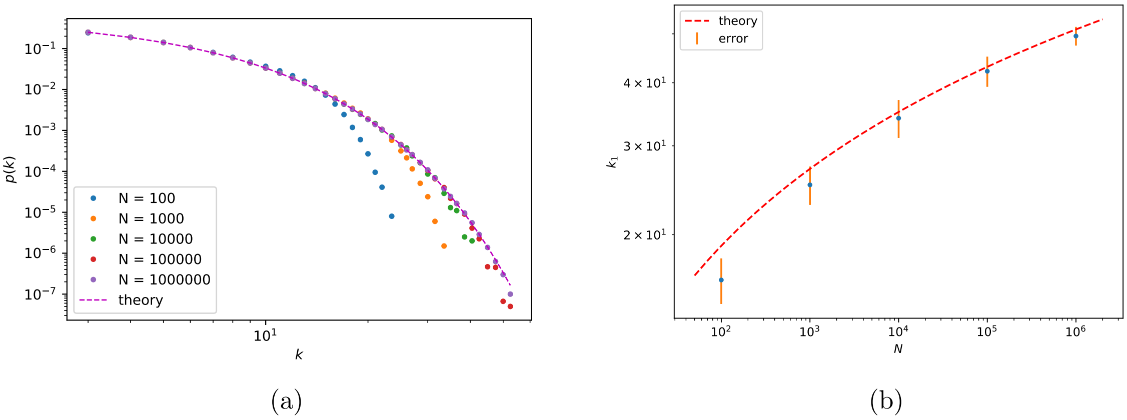


Figure 8: Degree distribution for $N = 100000$ and $m = 1, 5, 10$ and 20. The dashed coloured curves are the theoretical predictions for corresponding values of m using eq. (30).

3.2.2 Largest Degree Numerical Results

The simulation is run for N = 100, 1000, 10000, 100000 and 1000000, each with $m = 3$ as previously. Figure 9(a) shows the resulting degree distribution. It also decays exponentially but for large k, finite size effects can be seen which can be characterised by its largest degree size, k_1. k_1 was calculated numerically for multiple runs. Figure 9(b) shows k_1 as a function of N along with its theoretical prediction. The theory fits the data better for larger N, suggesting the assumption might not be accurate for small N. The chi-square value obtained from comparing the numerical result with theoretical prediction is 0.621 with a p-value of 0.96. Hence, the theory fits the numerical data reasonably well for larger k.



(a)

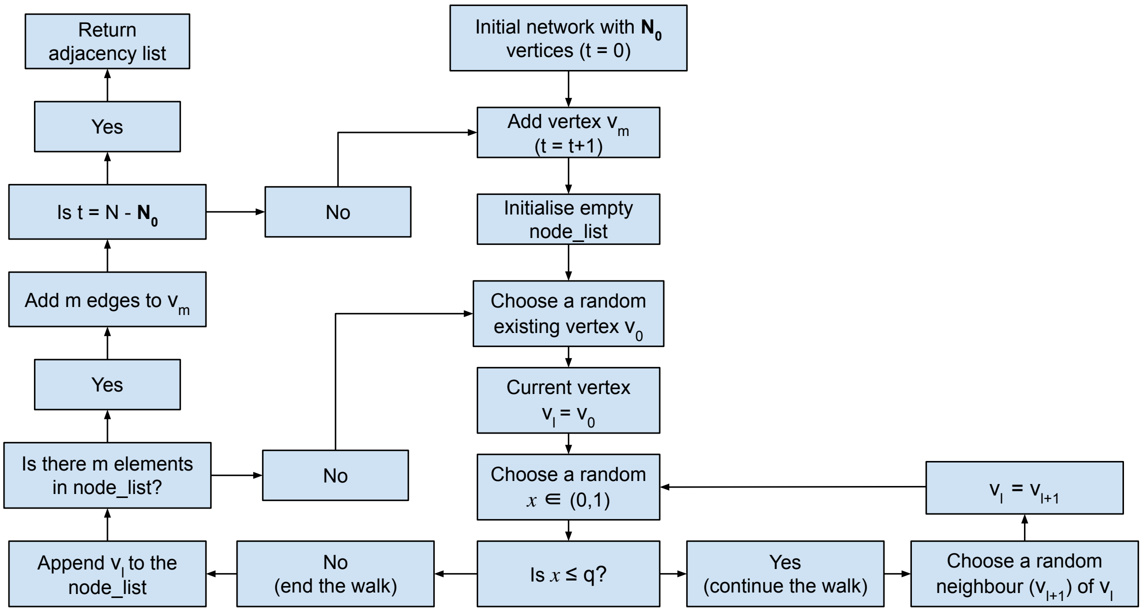
(b)

Figure 9: (a) Degree distribution for $N = 100$ (10000 runs), 1000 (1000 runs), 10000 (100 runs), 100000 (100 runs) and 1000000 (10 runs), with $m = 3$. The purple dashed curve corresponds to the theoretical prediction for $m = 3$ using eq. (30). (b) Largest expected degree k_1 as a function of N. The red dashed curve is the theoretical prediction from eq. (35) with $m = 3$. The standard deviation in the distribution of k_1 was assumed to be its error, which seems to be bigger for smaller N.

4 Phase 3: Random Walks and Preferential Attachment

4.1 Implementation

The summary of the numerical implementation of the model is shown in figure 10. The initial network was the same as in previous sections.



Return adjacency list
Initial network with $\mathbf{N_0}$ vertices (t = 0)
Yes
Add vertex v_m (t = t+1)
Is t = N - $\mathbf{N_0}$
No
Initialise empty node_list
Add m edges to v_m
Choose a random existing vertex v_0
Yes
Current vertex $v_l = v_0$
Is there m elements in node_list?
No
Choose a random $x \in (0,1)$
$v_l = v_{l+1}$
Append v_l to the node_list
No (end the walk)
Is $x \leq q$?
Yes (continue the walk)
Choose a random neighbour (v_{l+1}) of v_l

Figure 10: The numerical implementation of a random walk. q is the probability of continuing the walk.

4.2 Numerical results

I ran the simulation for $q = 0$, $q = 0.5$ and $q = 0.9$, for $N = 1000000$ with $m = 4$. Figure 11 shows the resulting degree distributions. For $q = 0$, the plot decays exponentially, like a pure random attachment model. In fact, I plotted the random attachment theoretical equation and calculated its chi-square value, which was very close to zero suggesting it agreed too a high degree. For $q = 0.9$, the plot looks linear, indicating a power-law. The exponent fitted was -3.07 ± 0.05, which matches the preferential attachment model (BA model). For $q = 0.5$, the distribution decays exponentially in the beginning and after $k \approx 30$, it decays like a power-law. But this power-law decays much more quickly than the one for $q = 0.9$. Final few points were fitted to a power-law and the exponent obtained was -5.0 ± 0.2.

4.3 Discussion of Results

When $q = 0$, the random walk ends at the first vertex chosen randomly from existing vertices. This is exactly like a random attachment as seen. When q is close to 1, the random walk will continue for a considerably longer time. The probability of attachment depends on how well connected the final vertex of the walk is. This indicates preferential attachment since the probability is proportional to the degree of the final vertex. For $q = 0.5$, the probability of attachment is a mixed form of both the mechanisms as seen above.

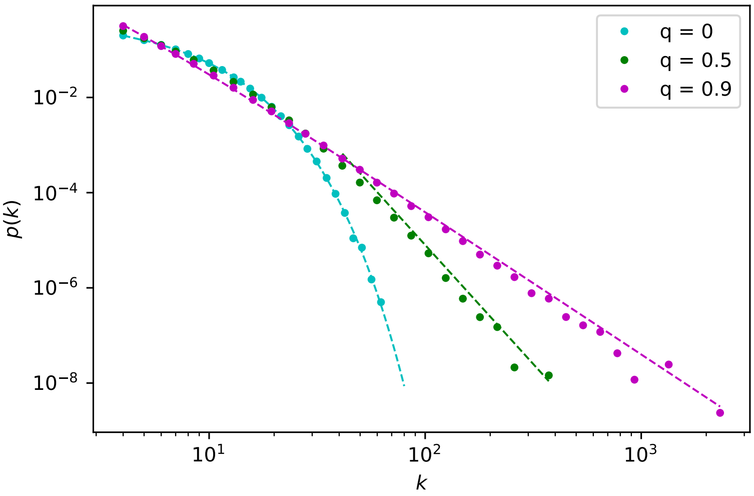


Figure 11: Degree distributions for $N = 1000000$ with $m = 4$; for $q = 0$, 0.5 and 0.9. The data was log-binned as before. The cyan-coloured dashed curve for $q = 0$ corresponds to the theoretical prediction from random attachment model using eq. (30). The green dashed line is the power-law with an exponent of -5. The magenta-coloured line corresponds to the theoretical prediction for preferential attachment model using eq. (12).

5 Conclusions

The degree distributions in preferential attachment (BA model) and pure random attachment models were investigated analytically and numerically. Both theoretical and numerical results were compared and found to agree with each other reasonably well. The BA model behaves like a power-law with an exponent of -3 for large k. Compared with the preferential attachment model, the degree distribution in pure random attachment model decays much more rapidly. In addition, an attachment model involving random walks was also simulated and studied. It was found that the degree distribution depended on the probability of continuing the random walk.

References

[1] T.S. Evans, *Networks Lecture Course Notes*, Complexity and Networks course, Physics Department, Imperial College London; (2019).

[2] T.S. Evans, *Networks Problem Sheet 2: Random Networks*, Complexity and Networks course, Physics Department, Imperial College London; (2019).

[3] Mitchell, Bruce. *A Comparison of Chi-Square and Kolmogorov-Smirnov Tests.* Area, vol. 3, no. 4, 1971, pp. 237–241. JSTOR, www.jstor.org/stable/20000590. Accessed 23 Mar. 2020.