# Network Project
# A Growing Network Model

CID: 01351497

10th February 2020

**Abstract**: Using BA model, we found that degree distribution of pure preferential attachment model follow an approximately inverse cubic law and the maximum degree is directly proportional to $\sqrt{N}$ while those of random attachment model follows a more complicated relationship.

**Word Count**: 1494 words

# 1 Introduction

We aim to use BA model to explore about degree distribution and maximum degrees of freedom in both pure preferential attachment and random attachment model.

## 1.1 Definition

BA model incorporates two important features which are growth and preferential attachment such that the model could generate scale-free network. Thus, the system will have a power law degree distribution. BA model is used to explain man-made systems such as internet and social network. [**2 marks**]

# 2 Phase 1: Pure Preferential Attachment $\Pi_{\text{pa}}$

## 2.1 Implementation

### 2.1.1 Numerical Implementation

To implement the preferential attachment model, an attachment list is created to includes all the end points of edges. To move on, I create a target list which has m elements and each of them are selected from the attachment list uniquely and randomly which means if a node has more element in the list, it is more likely to be selected. After the target list have been filled, edges are added between the new node and the nodes in the target list. After that, a new node is added to the graph and this process keeps iterating until the number of nodes have reached the desired number. [**3 marks**]

### 2.1.2 Initial Graph

My initial graph is a complete graph with m+1 nodes and every node in the initial graph will have m as their degrees of freedom at the start. I used this as my initial graph as I would know that if any node has degrees of freedom less than m then the code is not working. [**3 marks**]

### 2.1.3 Type of Graph

Only simple graphs can be produced from this program as there are at most one edge between two nodes with no weight and direction. There is also no self-loops in the graph so it can be classified as a simple graph. [**4 marks**]

### 2.1.4 Working Code

According what I implemented above, each node should have at least m degrees of freedom. I wrote a function that checks the system satisfies the above condition if not error will be returned. Apart from that I check whether the 3 nodes added to the list are unique and the total number of nodes at the end. The program turns out to not return any error, so I am very confident that the program is working. [**2 marks**]

### 2.1.5 Parameters

The program requires m, initial number of nodes and final number of nodes as its parameters. The number of edges added is m and I set the initial number of nodes as m+1 for convenience. [**2 marks**]

## 2.2 Preferential Attachment Degree Distribution Theory

### 2.2.1 Theoretical Derivation

Using the BA model solution given below

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

The delta term could be ignored as it is a boundary term. The equation could then be expressed as

$$\frac{P_\infty(k)}{P_\infty(k-1)} = \frac{k-1}{k+2}$$

which can be further simplified using gamma function

$$P_\infty(k) = A\frac{\Gamma(k)}{\Gamma(k+3)}$$

I could express gamma function as factorial to express $P_\infty(k)$ as a function of $k$

$$P_\infty(k) = \frac{A}{k(k+1)(k+2)}$$

where A is a constant. To proceed, the fraction have to be expressed as 3 terms such that

$$\frac{1}{k(k+1)(k+2)} = \frac{C_1}{k} + \frac{C_2}{k+1} + \frac{C_3}{k+2}$$

where $C_1 = 0.5$, $C_2 = -1$, $C_3 = 0.5$. Then we could write the sum explicitly.

$$\sum_{k=m}^{\infty} \frac{C_1}{k} + \frac{C_2}{k+1} + \frac{C_3}{k+2} = \frac{C_1}{m} + \frac{C_2}{m+1} + \frac{C_3}{m+2} + \frac{C_1}{m+1} + \frac{C_2}{m+2} + \frac{C_3}{m+3} + ...$$

We found that any terms beginning from $m+2$ can be ignored as the sum of all the constants equal to zero.

$$\sum_{k=m}^{\infty} \frac{C_1}{k} + \frac{C_2}{k+1} + \frac{C_3}{k+2} = \frac{C_1}{m} + \frac{C_2+C_1}{m+1} + \frac{C_3+C_2+C_1}{m+2} + ...$$

Then we could use the property that probability should be normalised.

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

and we could find $A$ as

$$A = 2m(m+1)$$

and $P_\infty(k)$ can be expressed as

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

[**6 marks**]

### 2.2.2 Theoretical Checks

We could check whether $P_\infty(k)$ is normalised by summing equation(1).

$$\sum_{k=m}^{\infty} P_\infty(k) = \frac{1}{2}[(m-1)P_\infty(m-1) - mP_\infty(m) + (m)P_\infty(m) - (m+1)P_\infty(m+1) + ...] + \delta_{m,m}$$

The first term inside the bracket vanishes as $P_\infty(m-1) = 0$ and the rest equal to zero as they cancel each other. The whole bracket vanishes and we could prove that $P_\infty(k)$ is normalised. [**6 marks**]

## 2.3 Preferential Attachment Degree Distribution Numerics

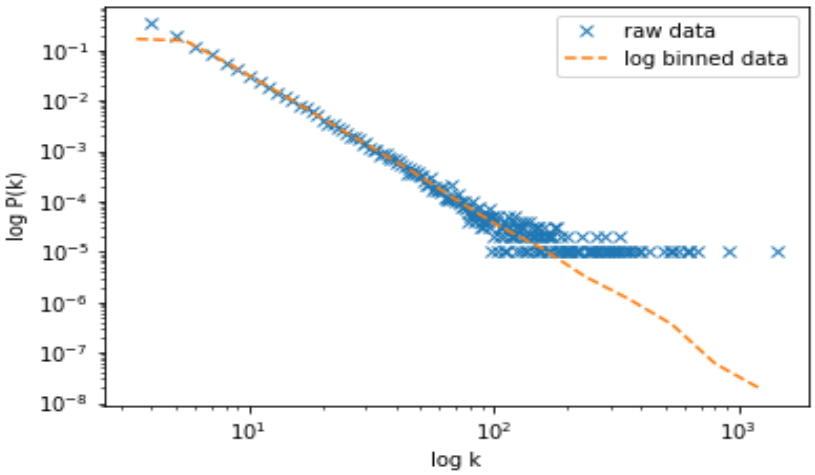
### 2.3.1 Fat-Tail

A fat-tailed distribution is one where you could see a roughly linear fall off on log-log plot. Fat-tailed distribution causes problems like few data points for large degrees. These data points for large degrees are crucial to understand the system. To deal with these probelms, we could use log binning where we define ith bin to range from $b_i$ to $b_{i+1}$ as shown below

$$\frac{b_{i+1}}{b_i} = exp(\Delta)$$

where $\Delta > 0$. Number of nodes with degree in each bin is counted and density will be plotted at the midpoint of each bin. After log binning the data points will be spaced out at intervals of $\Delta$ in a log log plot for $P(k)$ against $k$ as shown below.

Figure 1: Comparison between Raw Data and Log Binned Data

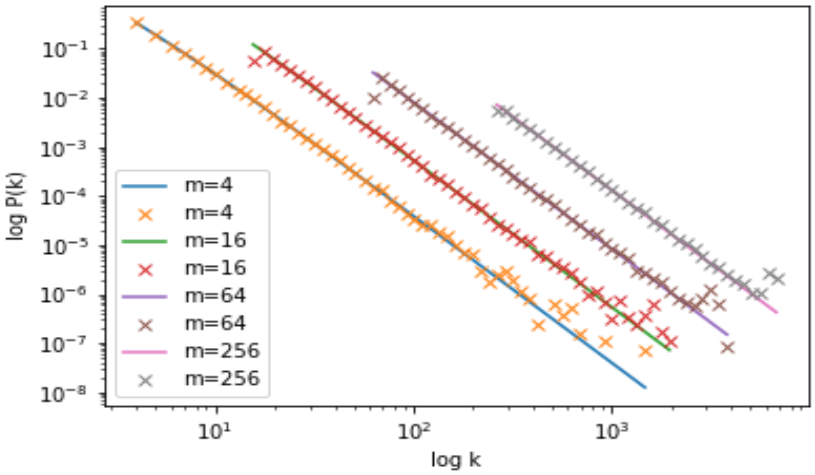



[**4 marks**]

### 2.3.2 Numerical Results

To compare theoretical results and numerical data, data is collected for $m = 4, 16, 64, 256$ and $N = 100000$. A log bin function is then used to process the data and theoretical results are calculated using the mid points of bins returned by the function. Theoretical results are represented by solid lines and numerical data are by crosses.

Figure 2: Theoretical Result and Numerical Data for Degree Distribution




By comparing them visually, I am confident that I have a good fit of numerical data by the theoretical results. [**4 marks**]

### 2.3.3 Statistics

A chi-square test is ultilised in this part to determine whether the data are consistent with the theoretical predictions. To start the test, we have the hypothesis listed below

- $H_0$:The data are consistent with a specified distribution.
- $H_1$:The data are not consistent with a specified distribution.

$$(\chi^2) = \sum \frac{(Observed - Expected)^2}{Expected}$$

The formula is used to calculate $\chi^2$ and the results are as shown below

| m | 4 | 16 | 64 | 256 |
|---|---|---|---|---|
| $\chi^2$ | 0.000 | 0.078 | 0.053 | 0.000 |

The data have a degree of freedom and the confidence level is set at 5 percent for this test. Therefore I could reject $H_1$ at 5 percent confidence level.

[**4 marks**]

## 2.4 Preferential Attachment Largest Degree and Data Collapse

### 2.4.1 Largest Degree Theory

We expect at least one node will have degrees of freedom ranging from $k_1$ onwards among other nodes in the system. Therefore we could write this

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

With similar prodecures applied in 2.21 we could arrive at this expression

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

and we could express the above in terms of a quadratic equation

$$k_1^2 + k_1 - Nm(m+1) = 0$$

Using quadratic formula, we find $k_1$ to be

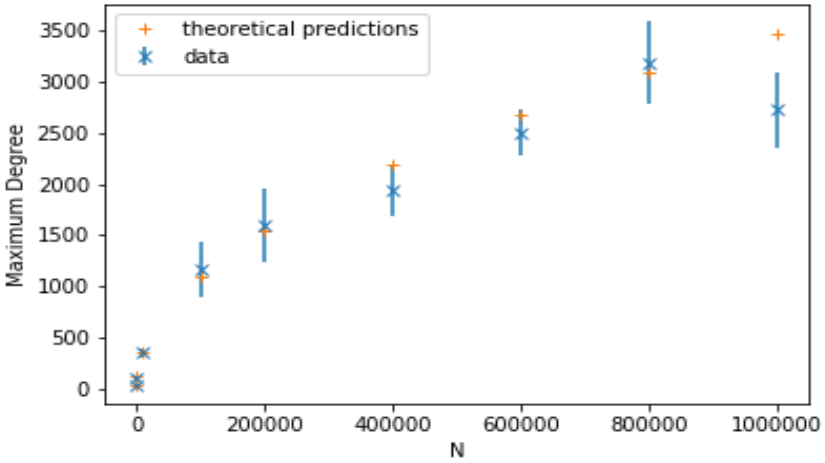
$$k_1 = \frac{-1 + \sqrt{1 + 4Nm(m+1)}}{2} \tag{3}$$

Therefore $k_1$ is proportional to $\sqrt{N}$ for systems with the same value of m. [**4 marks**]

### 2.4.2 Numerical Results for Largest Degree

Data is collected at $N = 100, 1000, 10000, 100000, 200000, 400000, 600000, 800000, 1000000$ and m is chosen to be 3 to reduce computing time significantly for large N values. Five samples of $k_1$ is taken at each $N$ and we take the mean as the results and the standard deviation as its uncertainty. To compare numerical data and theoretical results, both of them are plotted against $N$ as demonstrated below.

Figure 3: Theoretical Result and Numerical Data for $k_1$



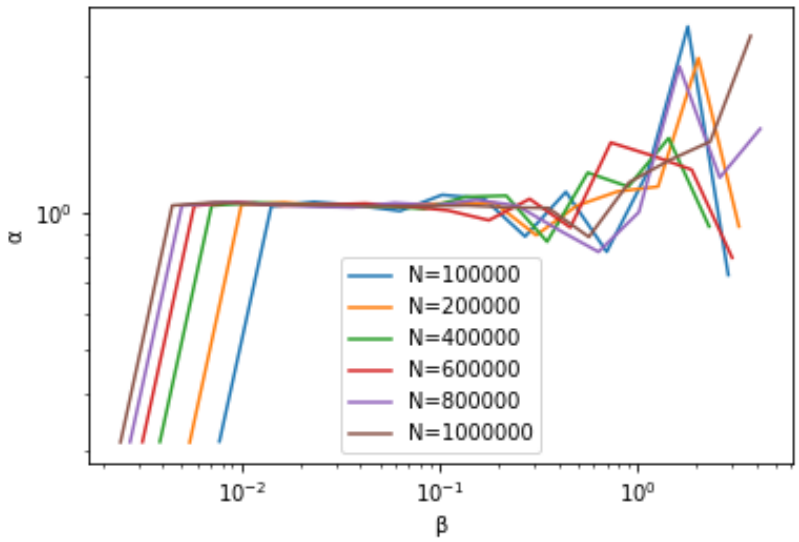

**[4 marks]**

where the blue bars are the uncertainty of the results and predictions match data nicely except for the last data point.

### 2.4.3 Data Collapse

To produce data collapse, we must first find out how $P(k)$ and $k_1(N)$ behave when $k >> 1$ and $N >> 1$. According to equation(3), $k_1(N)$ is directly proportional to $\sqrt{N}$ when $N >> 1$ and $P(k)$ is directly proportional to $P_\infty(k)$ when $k >> 1$. Therefore, we could produce data collapse by plotting $log\frac{P(k)}{P_\infty(k)}$ against $log\frac{k_1(N)}{\sqrt{N}}$.

Figure 4: Data Collapse




where $\alpha = log\frac{P(k)}{P_\infty(k)}$ and $\beta = log\frac{P(k)}{P_\infty(k)}$

**[4 marks]**

# 3 Phase 2: Pure Random Attachment $\Pi_{\text{rnd}}$

## 3.1 Random Attachment Theoretical Derivations

### 3.1.1 Degree Distribution Theory

The master equation for pure random attachment is stated below

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

To yield important relationship from this equation, we take time limit to infinity and use the relationship where $N(k,t)p(k,t) = n(k,t)$. Then we arrive at the results below

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

**[6 marks]**

### 3.1.2 Largest Degree Theory

To derive the expreesion for $k_1$, we follow the same procedures as shown in phase two. WE first expect at least one node has degrees of freedom from $k_1$ onwards.

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

By separating the terms,

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

By using the results of infinite geometric series and taking natural logs the final result is reached

$$k_1 = m - \frac{lnN}{ln\frac{m}{m+1}}$$

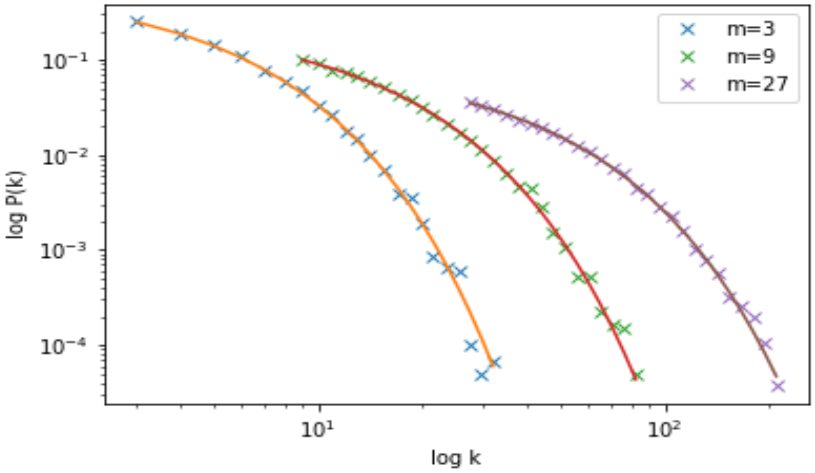
**[4 marks]**

## 3.2 Random Attachment Numerical Results

### 3.2.1 Degree Distribution Numerical Results

In this experiment, N is set to be 10000 and data are collected at $m$=3,9,27 . Data is then processed using the log bin function and they are plotted in logarithm scale with the theoretical prediction as below

Figure 5: Degree Distribution in Pure Random Attachment




where the solid lines are the theoretical prediction. Chi-squared test is used to test whether the data follows another distribution and we obtain these values in the table.

| m | 3 | 9 | 27 |
|---|---|---|---|
| $\chi^2$ | 0.001 | 0.001 | 0.000 |

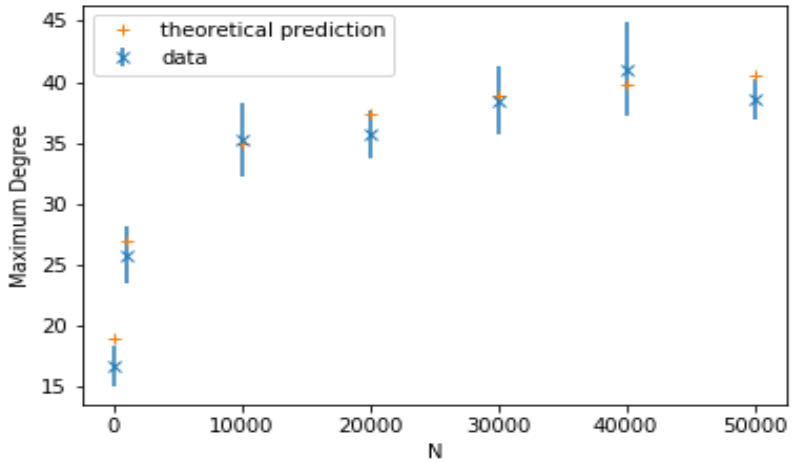
Thus we could reject the hypothesis that suggests that the data follows another distribution and the data is consistent with the theoretical prediction at 5% confidence level

[**4 marks**]

### 3.2.2 Largest Degree Numerical Results

Data is collected when $m$=3 and $N$=100,1000,10000,20000,30000,40000,50000. For each data point, ten samples are collected and the data points are the mean and the uncertainties are the standard deviation of these 10 samples.

Figure 6: Data Collapse




[**4 marks**]

where $\chi^2$=0.553, where we could confidently reject the hypothesis that suggests that the data follows another distribution and the data is consistent with the theoretical prediction at 5% confidence level

# 4 Phase 3: Random Walks and Preferential Attachment

## 4.1 Implementation

The random walk program is implemented in a way similar to preferential attachment. To start, I select a node at random from the graph and I set the system to return 0 with probability of $q$ and 1 with probability of $1-q$. If the system return one, the program will pick neighbouring nodes of the original one at random. However if the system returns 0, the last node will be the selected node. Process where system returns 0 will continue until it returns one and the last node will be the selected node. The selected node will then be checked whether it is unique in the target list. If yes, the list will be filled when there are m elements and m edges will be added from the new node to these selected nodes. Finally a new node is added to the graph. The process continue until the system reaches the number of nodes specified in the program. [**4 marks**]

## 4.2 Numerical results

[**6 marks**]

## 4.3 Discussion of Results

[**2 marks**]

# 5 Conclusions

Although failing to produce any results for phase 3, I have enjoyed doing this project as I see networks as one of new branches in physics that have a lot of applications in real life such as internet and social network and disease spreading.

# References


[1] T.S. Evans, *Amazing Paper*, Journal of Astounding Results, **7** (2019) 1234.

[2] K.Christensen and N.Maloney, *Complexity and Criticality*, Imperial College Press, London, 2005.