Department of Electrical and Computer Engineering

ECE 232E: Large Scale Social and Complex Networks: Design and Algorithms
Spring, 2020

Project 1: Random Graphs and Random Walks

Presented to:

Prof. Vwani Roychowdhury,
UCLA, Department of ECE

Group Members:

Samurdhi Karunaratne (305430927)
Shashank Narayana Gowda (504617384)
Vijay Ravi (805033666)

Contents

List of Figures

List of Tables

1 Generating Random Networks

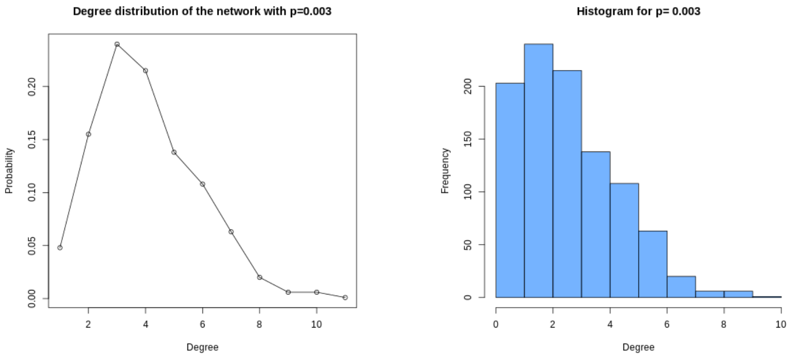
1.1 Create random networks using Erdos-Renyi model

1.1(a) Create undirected random networks with $n = 1000$ nodes, and the probability p for drawing an edge between two arbitrary vertices $0.003, 0.004, 0.01, 0.05$ and 0.1. Plot the degree distributions. What distribution is observed? Explain why? Also, report the mean and variance of the degree distributions and compare them to the theoretical values.

The Erdos-Renyi (ER) model is a mathematical model in the field of graph theory for generating random graphs or networks. In this project, the $G(n, p)$ variant of ER model was used for graph generation. The $G(n, p)$ model constructs a graph by connecting the nodes randomly. Each edge is included in the graph with probability p independent from every other edge. Thus, the higher the value of p, the denser the network. All graphs with n nodes and M edges have an equal probability of existing, given by:

$$p^M (1-p)^{\binom{n}{2}-M} \tag{1}$$

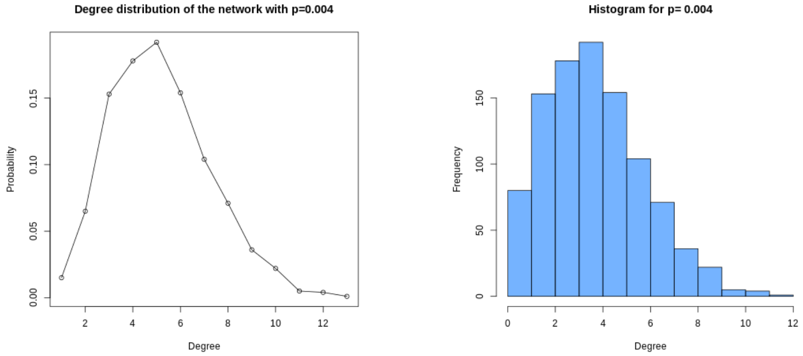
We created undirected random networks using the Erdos-Renyi model with the number of nodes, n= 1000 and the probability p for drawing an edge between two vertices is 0.003, 0.004, 0.01, 0.05 and 0.1. The plots for the degree distribution and the histogram are shown below in Figures 1-5.



(a)

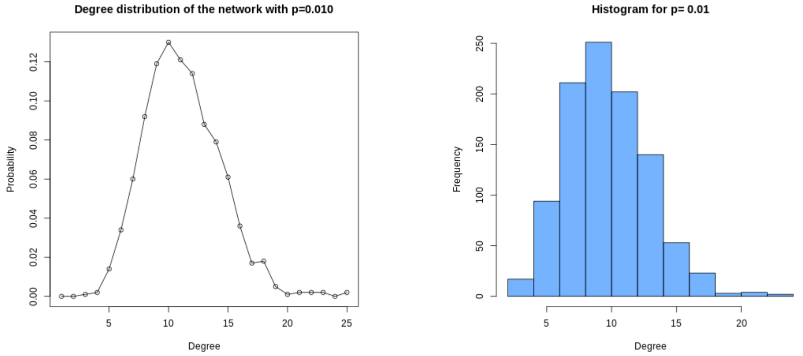
(b)

Figure 1: Degree distribution and histogram of distribution for p=0.003



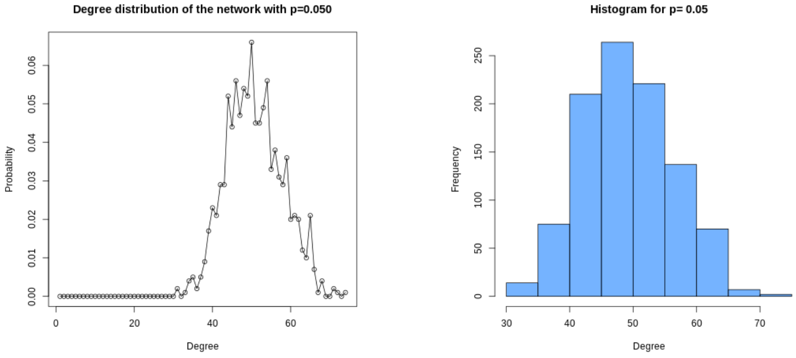
(a) (b)

Figure 2: Degree distribution and histogram of distribution for p=0.004



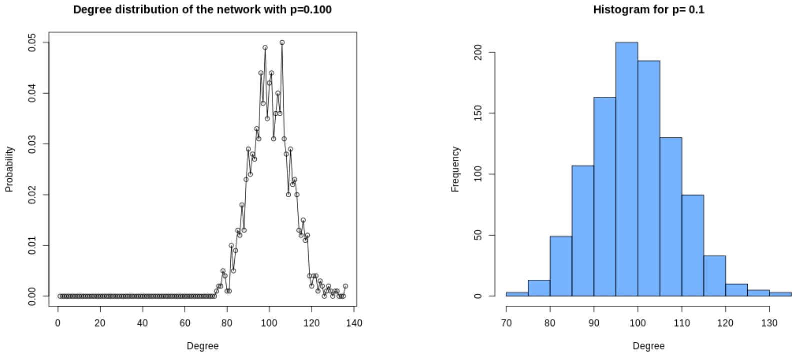
(a) (b)

Figure 3: Degree distribution and histogram of distribution for p=0.01



(a) (b)

Figure 4: Degree distribution and histogram of distribution for p=0.05



(a) (b)

Figure 5: Degree distribution and histogram of distribution for p=0.1

As mentioned before, the ER model is a random graph generation model in which each of the n nodes is independently connected (or not) with probability p (or $1-p$). The distribution of the degree of any particular vertex is therefore **binomial**. The Erdos–Renyi model has a degree distribution k given by the binomial distribution:

$$P(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k} \tag{2}$$

where n is the number of nodes and p is the probability for drawing an edge between two arbitrary vertices. In a particular case, when $n >> 1$ and $p << 1$, this binomial distribution approximates to a Poisson distribution given by:

$$P(k) = \frac{(np)^k e^{-(np)}}{k!} \tag{3}$$

p value	Actual Mean	Theoretical Mean	Actual Variance	Theoretical Variation
0.003	3.002	3	3.167	2.991
0.004	4.156	4	4.370	3.984
0.01	10.162	10	9.964	9.9
0.05	49.750	50	50.574	47.5
0.1	99.766	100	92.786	90

Table 1: Mean and Variance of the degree distribution for $n = 1000$ and varying p.

Table 1 shows the experimental mean and variance of the degree distributions and a comparison to the theoretical values. The mean of a binomial distribution is np and the variance is given by $np(1 - p)$. We can see that the theoretical values and experimental values are in agreement with each other. The mean and Variance are close to each other in these cases because the p values are small.

1.1(b) For each p and $n = 1000$, answer the following questions: Are all random realizations of the ER network connected? Numerically estimate the probability that a generated network is connected. For one instance of the networks with that p, find the giant connected component (GCC) if not connected. What is the diameter of the GCC?

No, not all random realizations of the ER network are connected. In Table 2, we show the numerical estimate of the probability that a generated network is connected. The diameter of the giant connected component is also shown.

p value	Probability of being connected	Connectedness	Diameter of the GCC	Number of nodes of the GCC	Number of Edges of the GCC
0.003	0	False	17	932	1496
0.004	0	False	11	982	2046
0.01	0.955	True	6	1000	4955
0.05	1	True	3	1000	24737
0.1	1	True	3	1000	50001

Table 2: Diameter of the GCC for each value of p.

From the table, we can see that as p value gets larger, the probability of the network being connected increase and the diameter of the GCC decreases. When the probability of connections increases the number of edges also increases and the same is observed in our experiments. Some observations are as follows - even at $p = 0.01$, the probability that the graph is connected is still not 100%. For some p value between 0.01 and 0.05, the probability of connected-ness becomes 100%. Therefore, the ER modelled network becomes connected once it has a minimum threshold p value.

1.1(c) It turns out that the normalized GCC size (i.e., the size of the GCC as a fraction of the total network size) is a highly nonlinear function of p, with interesting properties occurring for values where $p = \mathcal{O}(\frac{1}{n})$ and $p = \mathcal{O}(\frac{\ln n}{n})$. For $n = 1000$, sweep over values of p from 0 to a $p_{\max}$ that makes the network almost surely connected and create 100 random networks for each p. $p_{\max}$ should be roughly determined by yourself. Then scatter plot the normalized GCC sizes vs p. Plot a line of the average normalized GCC sizes for each p along with the scatter plot.

For $n = 1000$, we sweep over values of p from 0 to a $p_{max} = 0.01$ which makes the network almost surely connected. For each p, we create 100 random networks. The value of p_{max} was determined empirically by choosing a value greater than $p = \mathcal{O}(\frac{\ln n}{n})$. The scatter plot of the normalized GCC sizes vs p and the average normalized GCC sizes for each p are as shown in Figures 6 and 7 respectively.

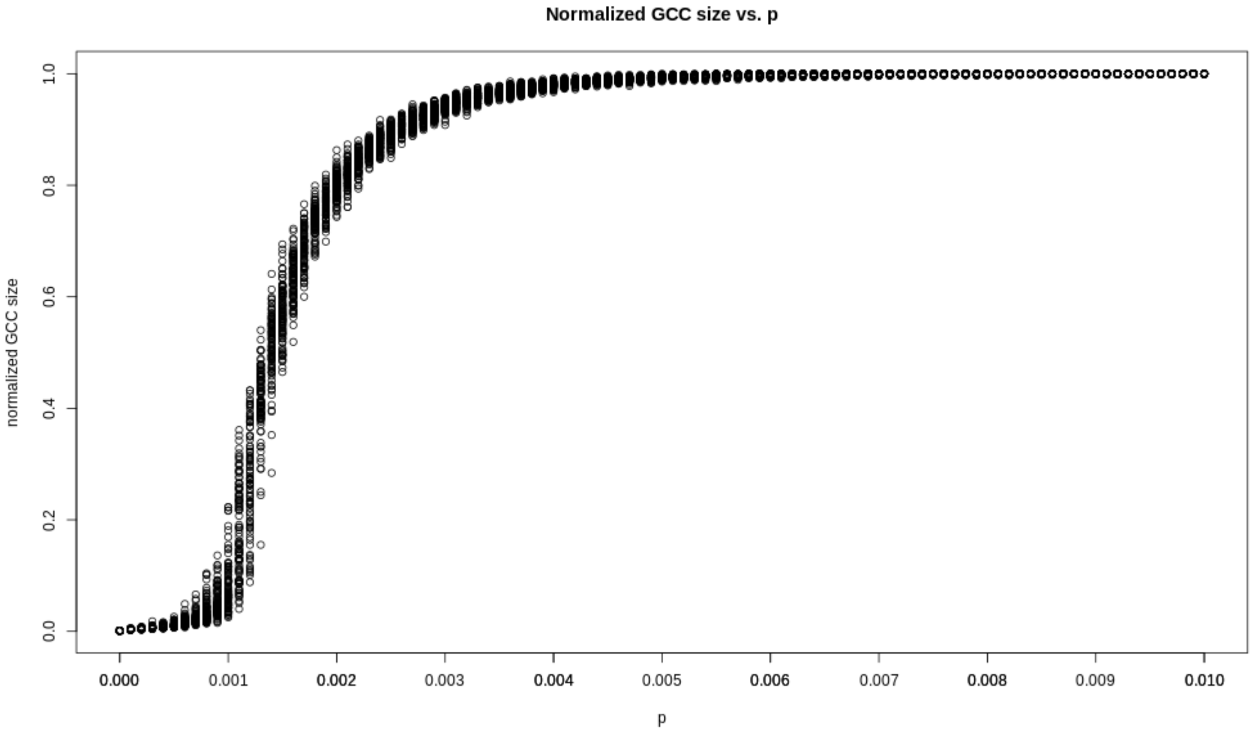


Figure 6: Normalised GCC size vs p

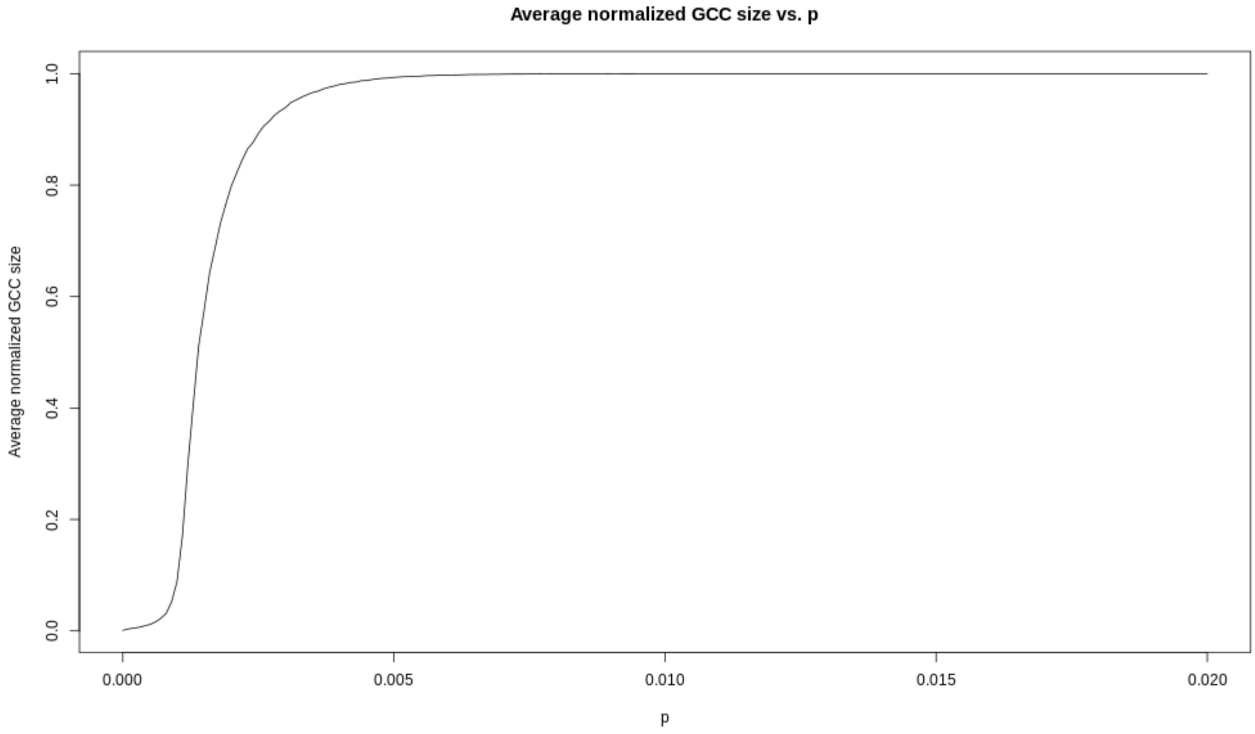


Figure 7: Average Normalised GCC size vs p

1.1(c).1 Empirically estimate the value of p where a giant connected component starts to emerge i.e., define your criterion of "emergence". Do they match with theoretical values mentioned or derived in lectures?

From the figure 6, we can see that GCC emerges at $p = 0.001$. The criterion for emergence of GCC is when size of normalized GCC starts to increase from zero. Theoretically this corresponds to $O(\frac{1}{n})$, where n=1000. Therefore, theoretical and empirical values match.

1.1(c).2 Empirically estimate the value of p where the giant connected component takes up over 99% of the nodes in almost every experiment.

At p=0.0069 in Figure 6, the giant connected component takes up over 99% of the nodes in almost every experiment. Theoretically this corresponds to $O(\frac{\log(n)}{n})$, where n=1000. Therefore, theoretical and empirical values match. After this point, the size the GCC has reached an asymptotic value.

1.1(d) Average degree of nodes np.

1.1(d).1 Define the average degree of nodes $c = np = 0.5$. Sweep over the number of nodes, n, ranging from 100 to 10000. Plot the expected size of the GCC of ER networks with n nodes and edge-formation probabilities $p = c/n$, as a function of n. What trend is observed?

We create networks with number of nodes n ranging from 100 to 10000. The average degree of nodes $c = 0.5$, the edge-formation probabilities p is equal to c/n. The expected size of the GCC for each of the networks created is shown in Figure 8 as function of p.

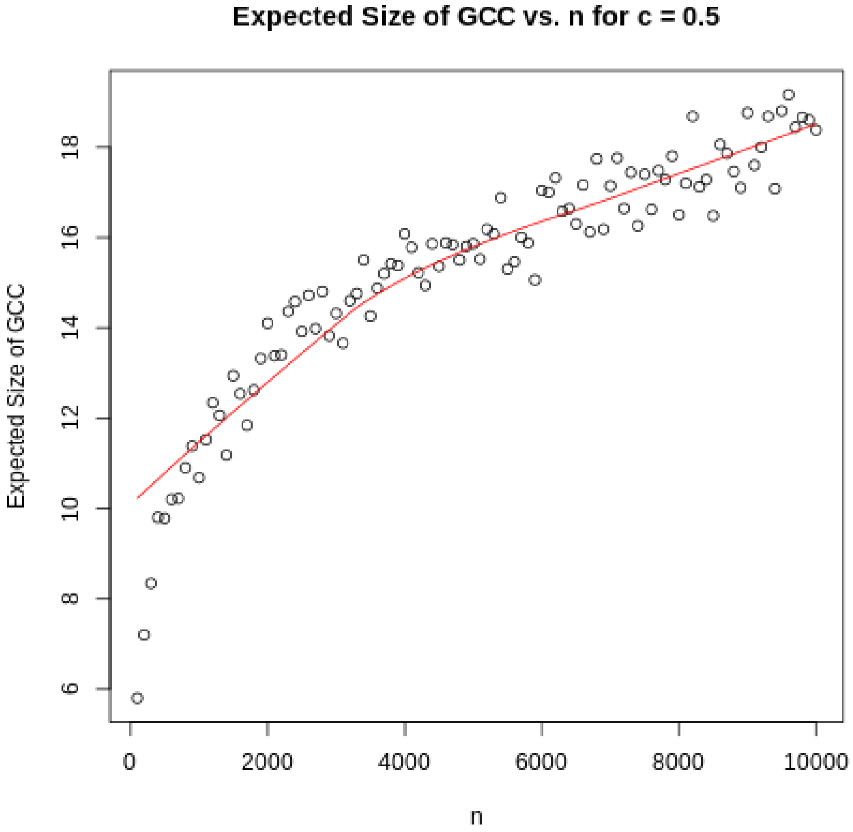


Figure 8: Expected Size of GCC as a function of number of nodes n for c=0.5. The red curve shows the trend.

The best fitting curve on the same plot shows the relationship between expected GCC size and the number of nodes(n) in each network for a given c. We observe a **linear** trend in the expected size of the GCC with respect to the size of the network. Beyond a certain threshold of n, the rate of increase in the expected size of GCC starts to reduce.

1.1(d).2 Repeat the same for $c = 1$.

The same experiments are repeated for $c = 1.0$. The plot of expected size of GCC vs number of nodes is as shown in Figure 9.

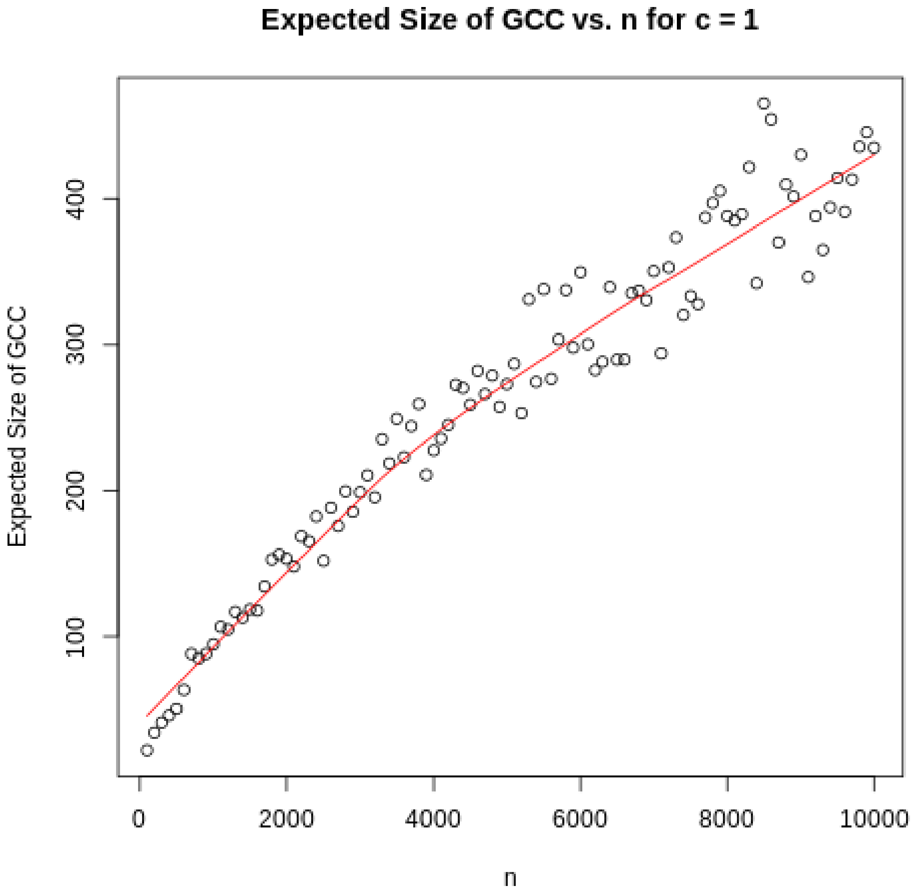


Figure 9: Expected Size of GCC as a function of number of nodes n for c=1

Again, the same trend is observed where the expected size of GCC has a linear increase with an increase in the number of nodes in the network. In this case, the slope of the curve (or the rate of increase in expected GCC size with respect to n) is smaller than the slope of the curve when $c = 0.5$. This is because the avg degree c was increased. So for a given n, p value will increase. Therefore, the probability of the edges getting connected increases and thus, the expected size of GCC is higher.

1.1(d).3 Repeat the same for values of $c = 1.1, 1.2, 1.3$, and show the results for these three values in a single plot.

The plot of Expected size of GCC as a function of number of nodes n for varying values of c is as shown in Figure 10.

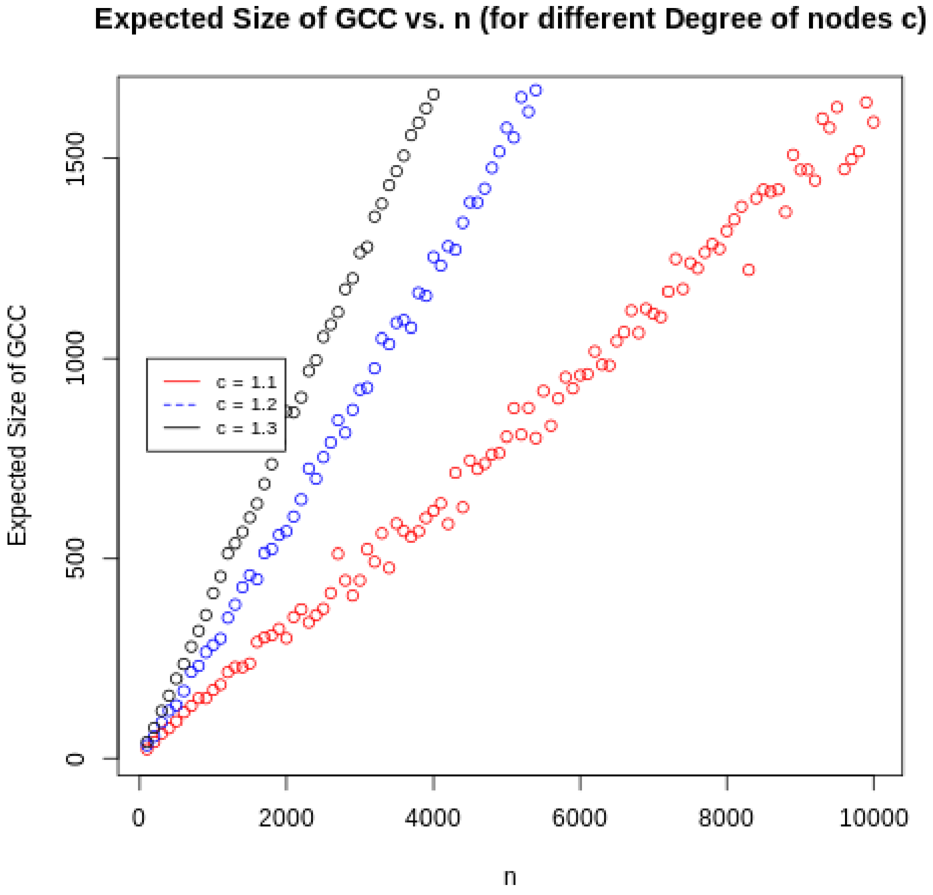


Figure 10: Expected Size of GCC as a function of number of nodes n for varying c.

1.1(d).4 What is the relation between the expected GCC size and n in each case?

From the plot in Figure 10 we can see that the relationship between the expected size of GCC and the number of nodes n is linear for a given average degree of nodes c. The slope of this linear relationship increases as the value of c increases. This is expected because for a given constant n, if c increases, then probability of edge formation p increases. This means more nodes are connected thereby increasing the size of the GCC.

1.2 Create networks using preferential attachment model

1.2(a) Create an undirected network with $n = 1000$ nodes, with preferential attachment model, where each new node attaches to $m = 1$ old nodes. Is such a network always connected?

In this section, we generate networks using the Barabasi–Albert (BA) model of network construction. This BA model is an algorithm for generating random scale-free networks using a **preferential attachment mechanism**. In this mechanism, the new nodes of the network are attached to older nodes of the network in a preferential manner in accordance with their degrees, so that those nodes which have higher degree receive more new connections than those who do not. In simpler terms, the new nodes have a ”preference” to connect themselves to the already heavily connected nodes. The degree distribution resulting from the BA model is scale free, in particular, it is a power law of the form

$$P(k) \sim ck^{-\gamma}, \tag{4}$$

where, γ is the power-low coefficient whose value is in the range $-2 \leq \gamma \leq -3$. We create an undirected network with n=1000 nodes, with preferential attachment model. Here, each new node attaches to m=1 old nodes. As a result, the network is always connected.

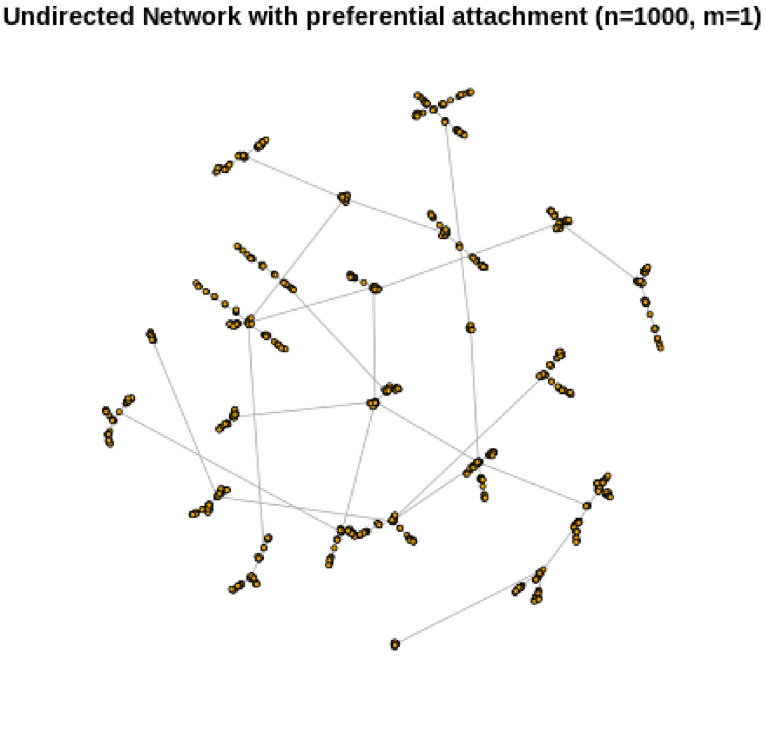

Undirected Network with preferential attachment (n=1000, m=1)

Figure 11: Undirected network with preferential attachment model (n=1000, m=1)

1.2(b) Use fast greedy method to find the community structure. Measure modularity.

Qualitatively, a community structure is defined as a subset of nodes within the graph such that connections between the nodes are denser than connections with the rest of the network. The fast greedy method finds the community structure within the network topology by greedily optimizing the modularity. It starts with considering each vertex as being the sole member of a community of one and then repeatedly joins together the two communities whose merging produces the largest increase in the modularity.

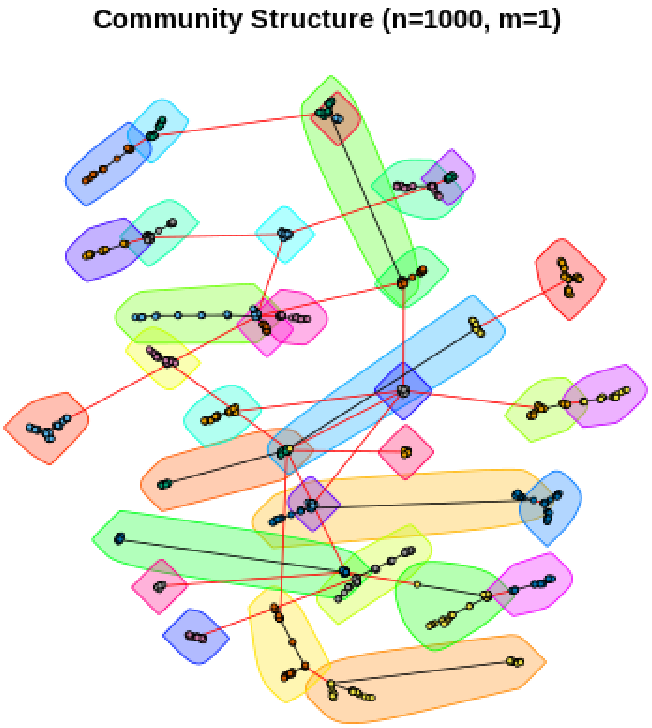
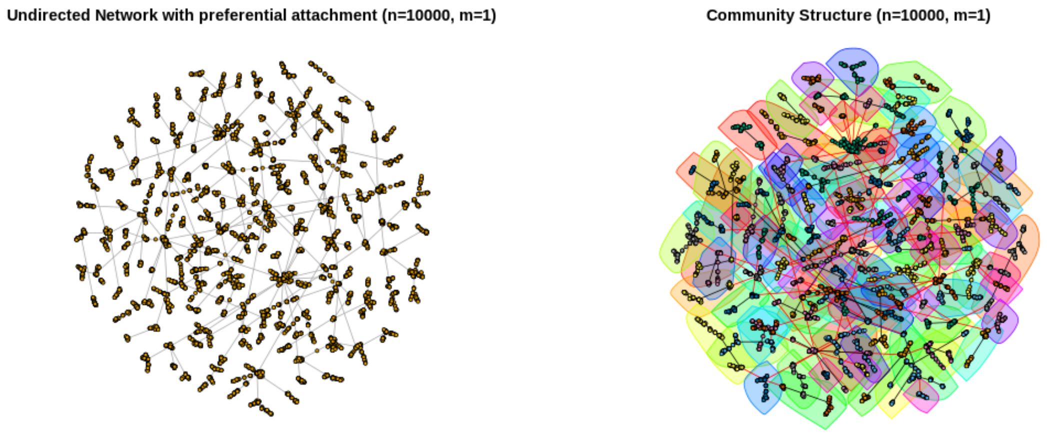


Figure 12: Community structure of the Undirected network with preferential attachment model (n=1000, m=1)

Modularity is a measure of the structure of networks or graphs. It helps us measure the strength of division of a network into modules or smaller communities by quantifying the goodness of the division of the network into communities. Networks with high modularity have dense connections between the nodes within communities but sparse connections between nodes in different communities. In out case, modularity of the generated network is 0.934286.

1.2(c) Try to generate a larger network with 10000 nodes using the same model. Compute modularity. How is it compared to the smaller network's modularity?

Modularity for network with n=10000 increases to 0.978358. This is expected because as n increases, number of nodes in the smaller modules (communities) increases. This increases the intra-community connection and increases the sparsity of inter-community node connection thereby increasing the modularity of the network.


Undirected Network with preferential attachment (n=10000, m=1)
Community Structure (n=10000, m=1)

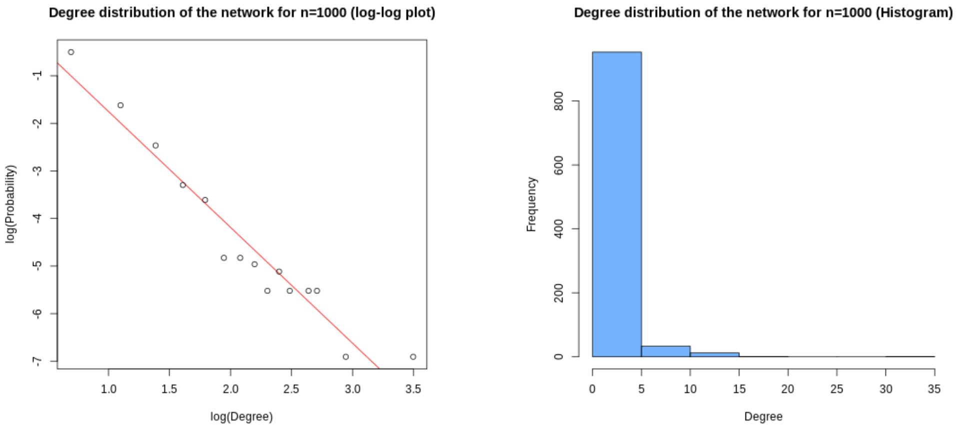
(a) Network

(b) Community Structure

Figure 13: Network with n=10000 and m=1

1.2(d) Plot the degree distribution in a log-log scale for both $n = 1000, 10000$, then estimate the slope of the plot using linear regression.

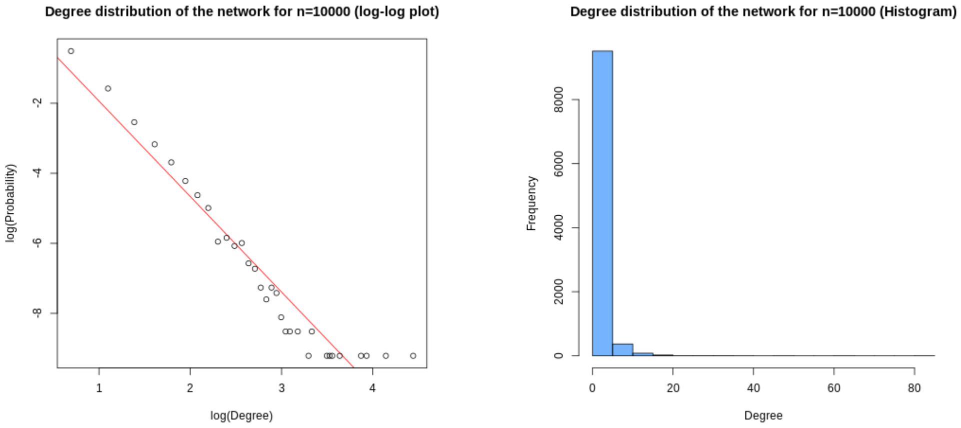
As shown in Equation 5, the degree distribution follows a power law distribution. The slope of the log-log degree distribution for a preferential attachment network can be useful to calculate the power law exponent of the distribution. Therefore, when we use a log-log plot to observe the distribution, theoretically, we must see a negative slope approximately equal to -3. Figures 14 and 15 below show the log-log distribution plot for $n = 1000$ and $n = 10000$. The slope of the linear regression line for the respective log-log plots is presented in Table 3.



(a) log-log plot

(b) histogram

Figure 14: Network with n=1000 and m=1



(a) log-log plot

(b) histogram

Figure 15: Network with n=1000 and m=1

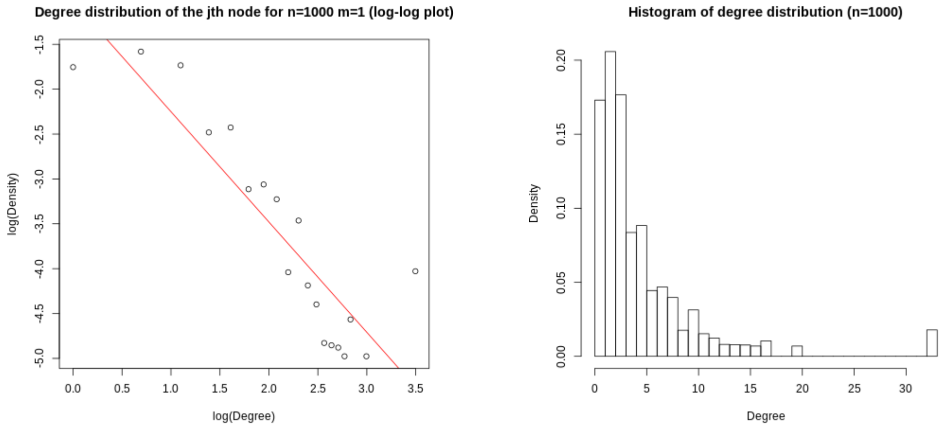
n	slope	intercept
1000	-2.4369	0.6869
10000	-2.7255	0.7856

Table 3: Slope of linear regression line for n=1000 and n=10000

From the figures, the power law exponent for n = 1000 is 2.4 and for n = 10000 is 2.7. This is a reasonable estimate given that the power law exponent of the Barabasi model is 3 (refer Equation 5).

1.2(e) In the two networks generated in 2(d), perform the following: Randomly pick a node i, and then randomly pick a neighbor j of that node. Plot the degree distribution of nodes j that are picked with this process, in the log-log scale. Is the distribution linear in the log-log scale? If so, what is the slope? How does this differ from the node degree distribution?

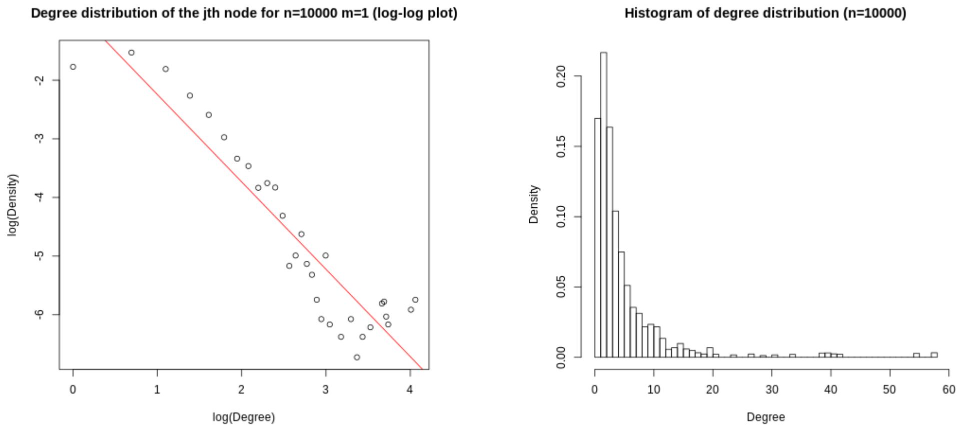
For the n=1000 and 10000, we randomly pick a node i and the randomly pick a neighbour j. of that node. We plot the degree of distribution of nodes j picked in this method on the log-log scale.



(a) log-log plot

(b) hist

Figure 16: degree distribution of jth node for n=1000.



(a) log-log plot

(b) hist

Figure 17: degree distribution of jth node for n=10000.

The trend is linear. Table 4 shows the slope of the line. It can be observed that although the slope is negative and decreases and n decreases, the degree distribution of the random node selection does not follow the power law distribution of the Barabasi Model having an exponent of 3.

n	slope	intercept
1000	-1.229	-1.022
10000	-1.4900	-0.7553

Table 4: Slope of linear regression line for jth node distribution for n=1000 and n=10000

1.2(f) Estimate the expected degree of a node that is added at time step i for $1 \leq i \leq 1000$. Show the relationship between the age of nodes and their expected degree through an appropriate plot.

The plots agree with theoretical intuition that older nodes have higher expected degree.

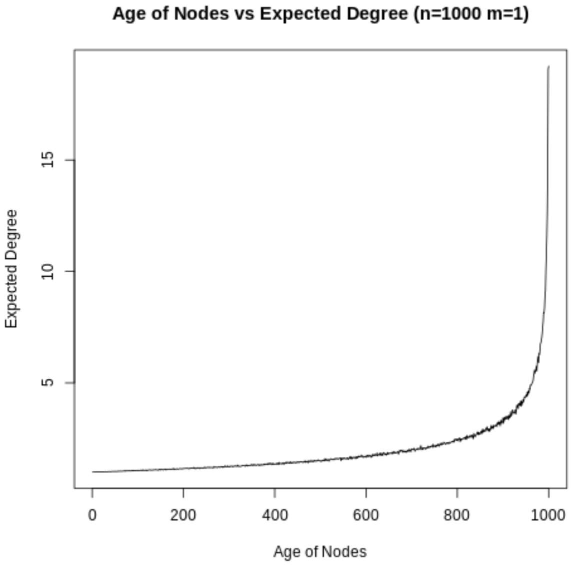


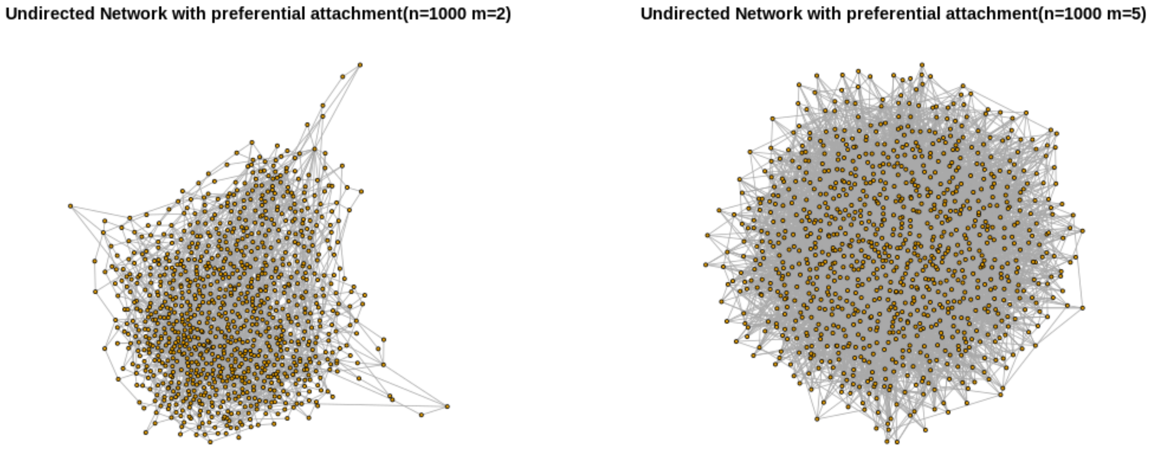
Figure 18: Expected degree as a function of Age of nodes

The above plot describes the relationship between age of the node and it's expected degree. It can be concluded that, as the age of the nodes increases, the expected degree of the node gets higher. This is in good agreement with the mathematical model of the Barabasi networks. As explained before, in the preferential attachment model a new node that gets added to the network has a higher probability of connecting or forming an edge with a well connected node (node with a higher degree). Hence, the nodes that are created in the beginning of the procedure benefits more compared to nodes that are created much recently.

1.2(g) Repeat the previous parts for $m = 2$, and $m = 5$. Compare the results of each part for different values of m.

In a Barabasi Network, the value m represents the number of edges to attach from a new node to the existing nodes. As m increases, the number of connections between new node and other nodes increases. This reduces the sparsity of inter-community connections which means a reduction in the modularity of the network.

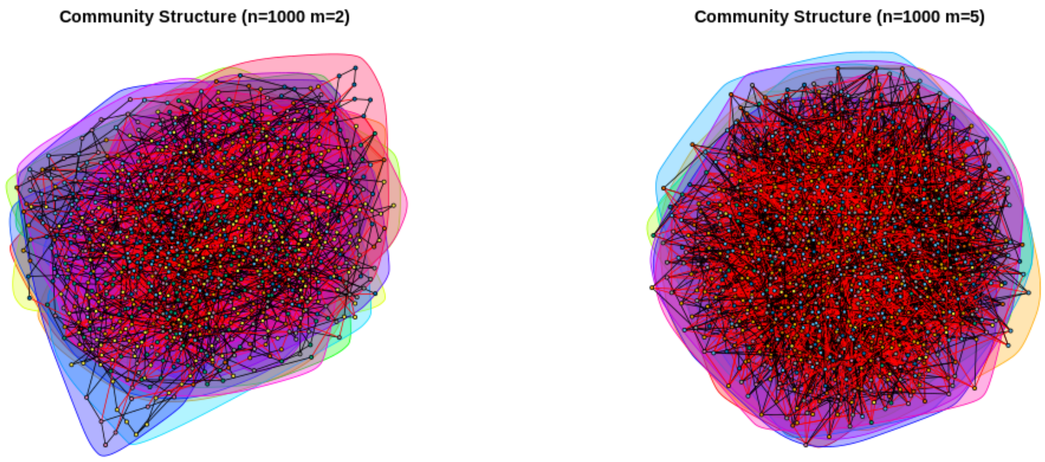
We repeat the above experiments 2a through 2f for $m = 2$ and $m = 5$. Plots and figures for each are shown below.


Undirected Network with preferential attachment(n=1000 m=2)
Undirected Network with preferential attachment(n=1000 m=5)

(a)

(b)

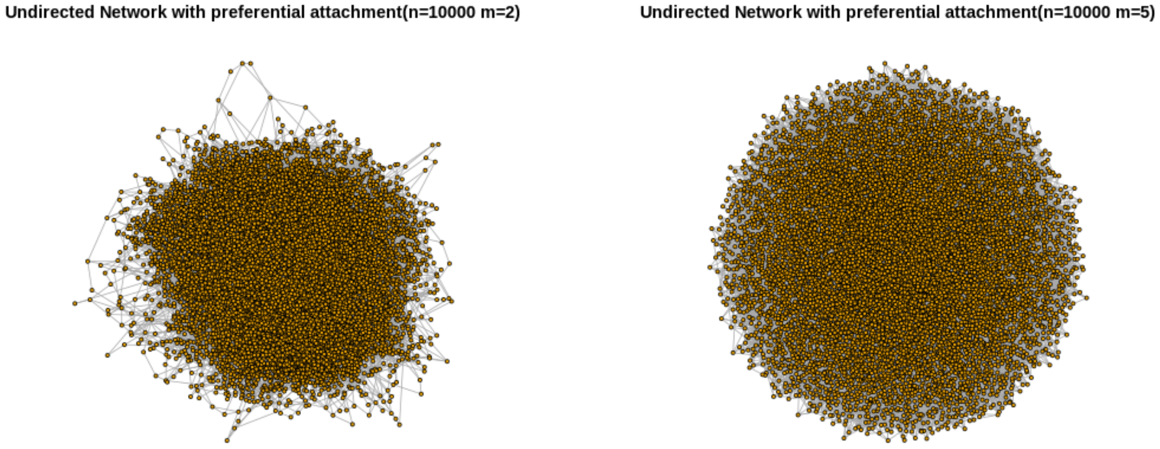
Figure 19: Undirected network with n=1000 and m = 2, m=5


Community Structure (n=1000 m=2)
Community Structure (n=1000 m=5)

(a) Modularity = 0.5278

(b) Modularity = 0.28145

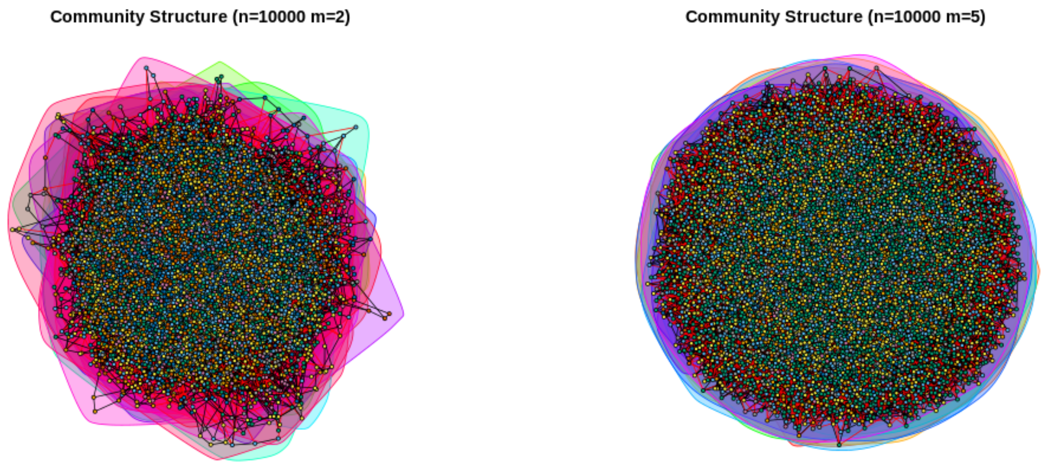
Figure 20: Community Structure with n=1000 and m = 2, m=5.


Undirected Network with preferential attachment(n=10000 m=2)
Undirected Network with preferential attachment(n=10000 m=5)

(a)

(b)

Figure 21: Undirected network with n=10000 and m = 2, m=5


Community Structure (n=10000 m=2)
Community Structure (n=10000 m=5)

(a) Modularity = 0.530985

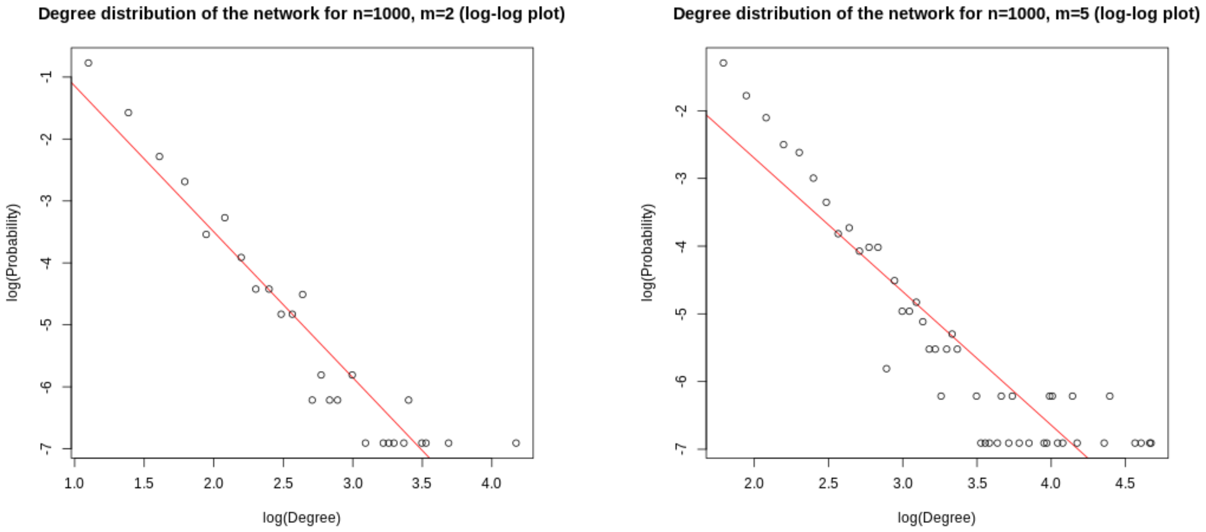
(b) Modularity = 0.278873

Figure 22: Community Structure with n=10000 and m = 2, m=5.

Figures 22-22 show the network and the community structure of the network for $n = 1000, 10000$ and $m = 2, 5$. Since $m > 1$, the networks are always connected in every case presented above. As the value of m increases, the number of edges becomes larger, thereby reducing the quality of the community structure.

Empirically, this is shown from the fact that the value of modularity decreases for larger m values. Modularity values are presented along with each plot of the community structure. Even in the case of larger m value, as number of nodes n increases, modularity further reduces. Therefore, the modularity is lowest for a network with $n = 10000$ and $m = 5$.

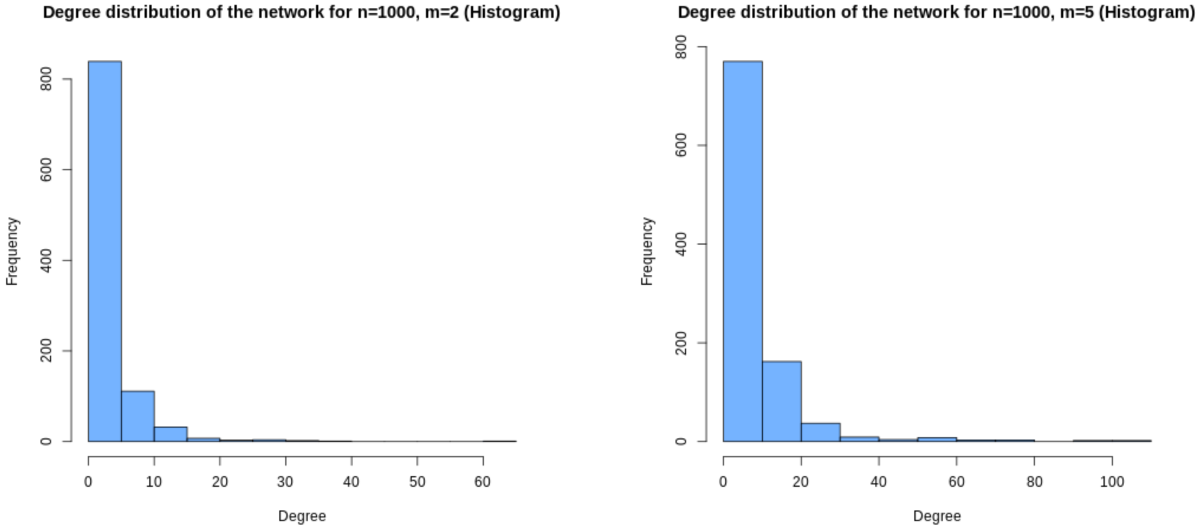
In the following figures 23 to 26, we plot the log-log degree distribution for varying values of n and m.



(a) Slope = -2.354

(b) Slope = -1.972

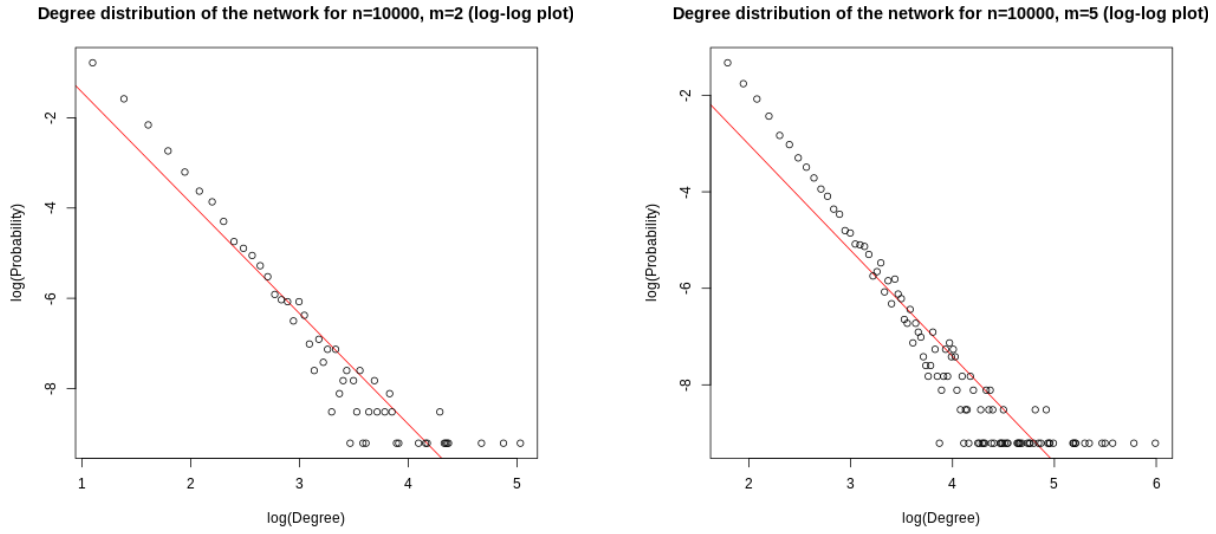
Figure 23: log-log distribution plot with n=1000 and m = 2, m=5



(a)

(b)

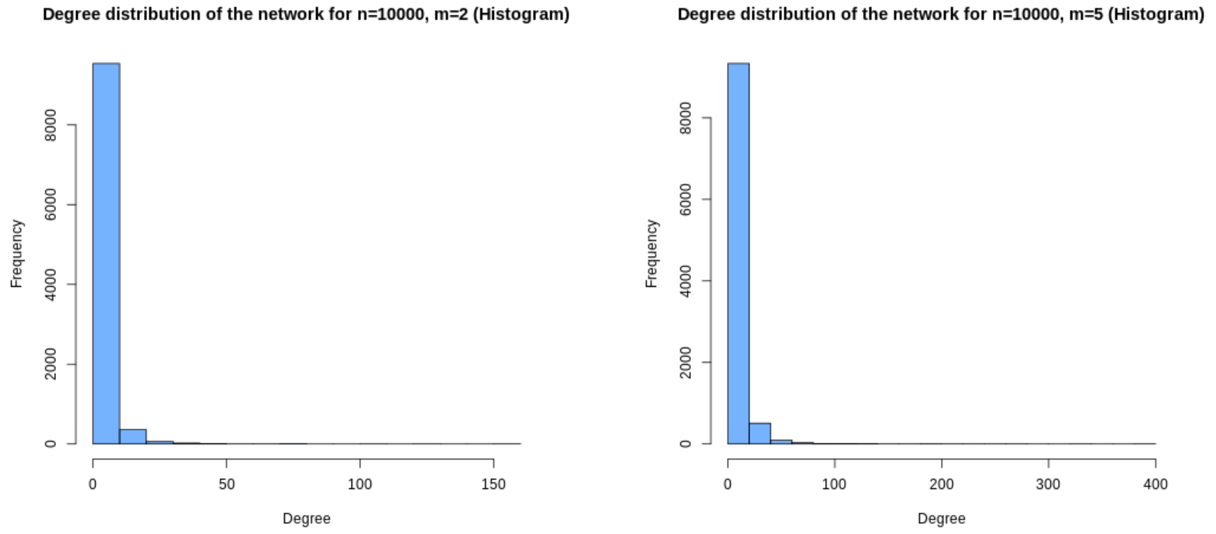
Figure 24: Histogram of degree distribution with n=1000 and m = 2, m=5



(a) Slope =-2.452

(b) Slope = -2.195

Figure 25: log-log distribution plot with n=10000 and m = 2, m=5



(a)

(b)

Figure 26: Histogram of degree distribution with n=10000 and m = 2, m=5

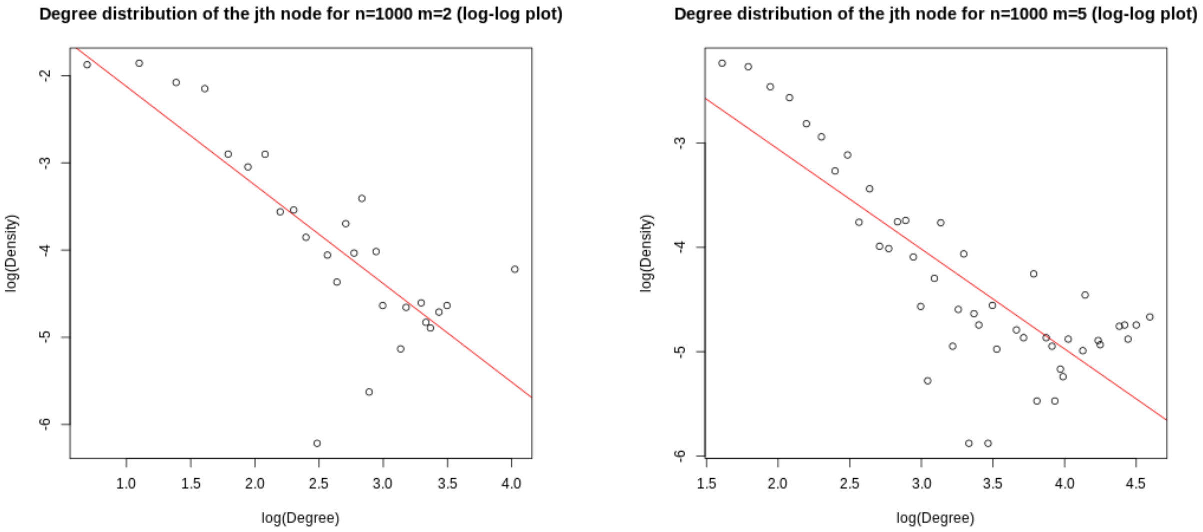
We try to estimate the slop of the log-log distribution plot which helps us estimate the value of the exponential component of the power law distribution. In each case, we observe a **linear relationship** with negative slope. The slopes for the above experiments are as shown below:

n	**m**	**slope**	**intercept**
1000	2	-2.354	1.209
1000	5	-1.972	1.243
10000	2	-2.452	1.019
10000	5	-2.195	1.371

Table 5: Slope of linear regression line for log-log plot for $m = 2$ and $m = 5$

We observe that for each value of n, as we increase the value of m from 1 to 5 and then to 5, the power law exponent deviates from the value of -3 for the Barabasi network and increases gradually. As γ, probablilty of a certain degree also increases and this increases is larger of larger value of k.

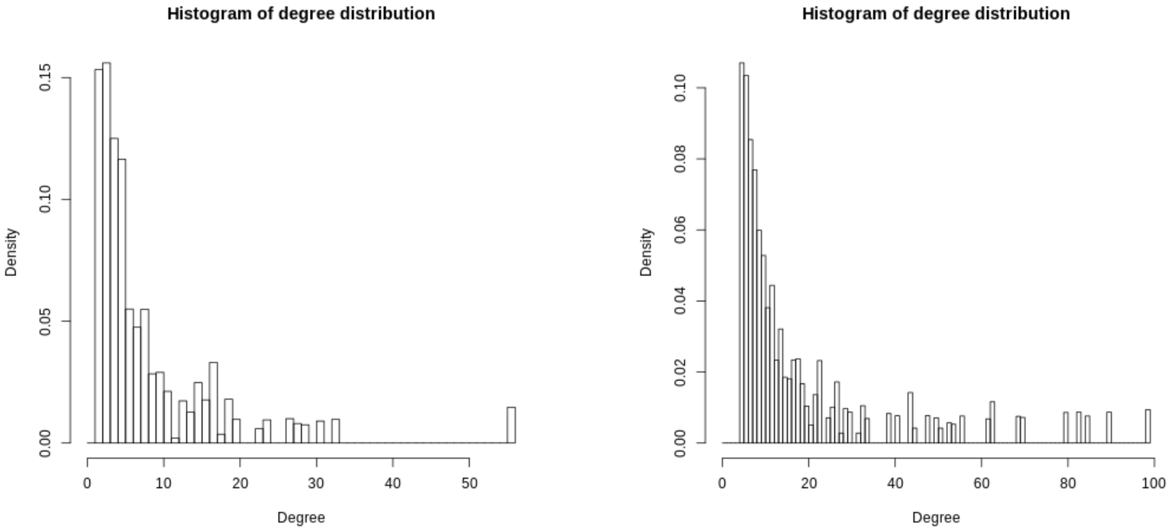
In the figures 27-30 to follow, we pick a random neighbour j of a random node i and plot the degree distribution on a log-log plot of that node. The slope for each case is mentioned in the table 6 following the plots.



(a) Slope = -1.1291

(b) Slope = -0.9575

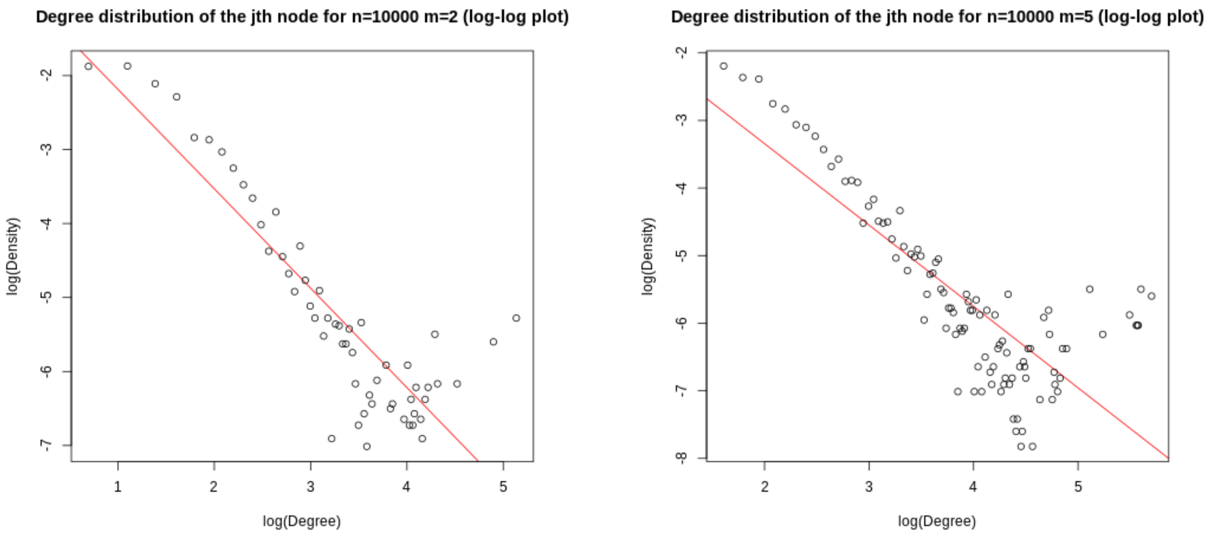
Figure 27: log-log plot for jth node distribution with n=1000 and m=2, m=5



(a)

(b)

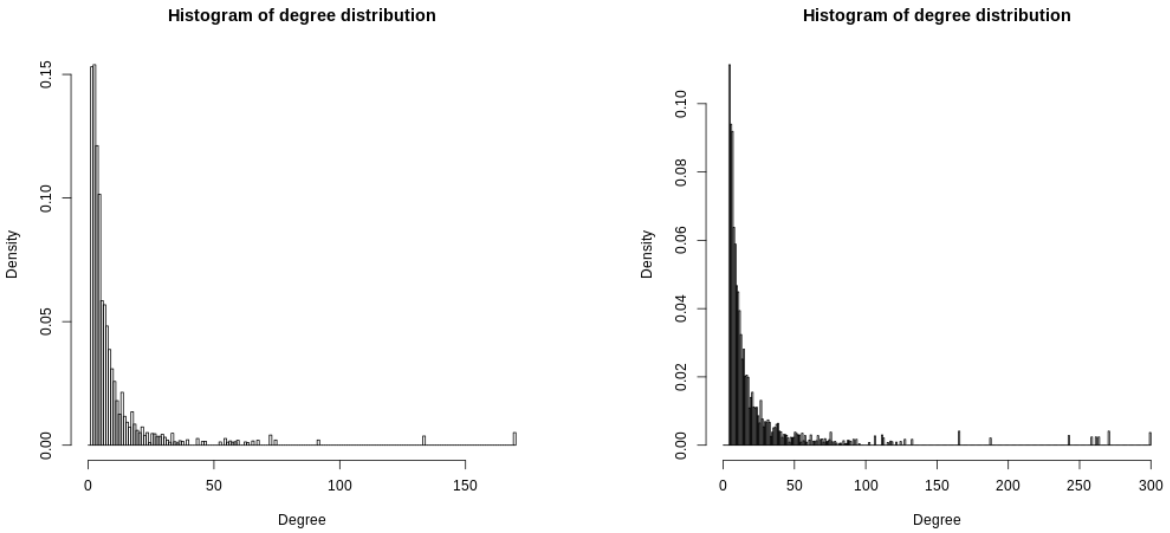
Figure 28: Histogram for jth node distribution with n=1000 and m = 2, m=5



(a) Slope = -1.3433

(b) Slope = -1.205

Figure 29: log-log plot for jth node distribution with n=10000 and m=2, m=5



(a)

(b)

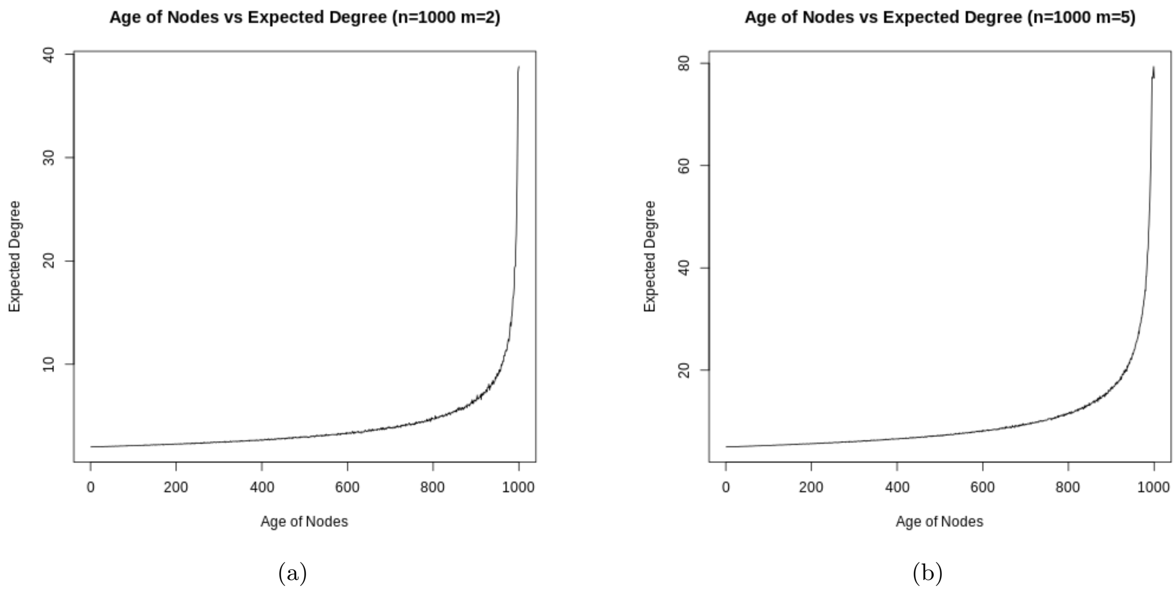
Figure 30: Histogram for jth node distribution with n=10000 and m = 2, m=5

n	m	slope	intercept
1000	2	-1.1291	-0.9971
1000	5	-0.9575	-1.1439
10000	2	-1.3433	-0.8447
10000	5	-1.205	-0.935

Table 6: Slope of linear regression line for log-log distribution plot of jth node for $m = 2$ and $m = 5$

The slope of the degree distribution for the jth node deviates significantly from the power law exponent of the Barabasi networks which is closer to -3. The trend however, is still linear with a negative slope. As the value of m increases, for a given number of nodes in the network n, the power law exponent for the jth node increases. This is the same as the power law exponent of the graph in general.

In figure 31, we show the Age of the node vs expected degree for networks with $n = 1000$ and $m = 2, 5$. We can infer from this plot that as age of the node increases, the expected degree increases. This is true for both values of m and is in good agreement with the preferential attachment model of the Barabasi network.



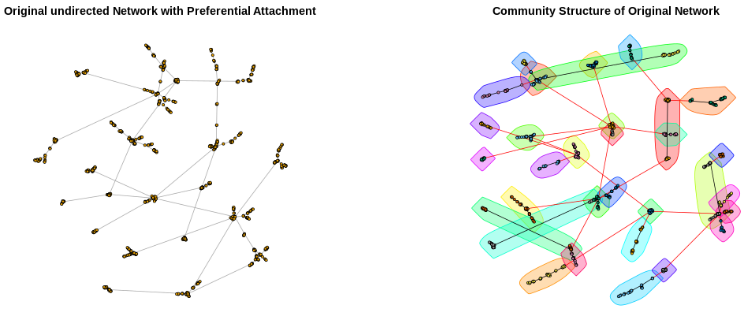
(a)

(b)

Figure 31: Expected degree as a function of Age of nodes for network with n=1000 and m=2,m=5

1.2(h) Again, generate a preferential attachment network with $n = 1000$, $m = 1$. Take its degree sequence and create a new network with the same degree sequence, through stub-matching procedure. Plot both networks, mark communities on their plots, and measure their modularity. Compare the two procedures for creating random power-law networks.

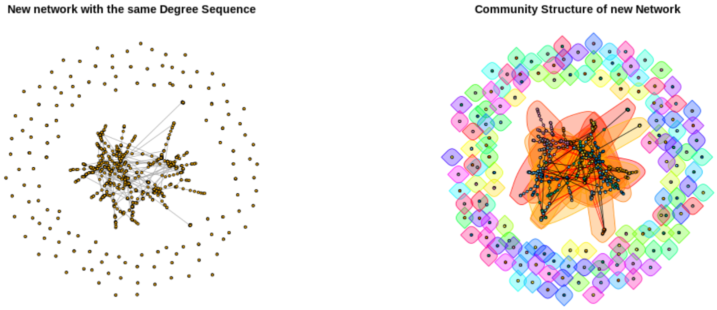
In the stub matching process, the degree of each vertex is pre-defined, rather than having a probability distribution from which the given degree is chosen. When compared to the ER model, the degree sequence of the configuration model is not restricted to have a Binomial or Poisson-distribution and the model gives the user the freedom to give the network any desired degree distribution. The main idea behind the stub-matching algorithm for creating random networks is that, the network with a given degree sequence is broken down into a collection of "stubs" (nodes with dangling edges) and then the random network is generated by randomly pairing up the stubs and connecting them. For both the networks, we use the fast greedy algorithm to detect the Community structure.


Original undirected Network with Preferential Attachment
Community Structure of Original Network

(a)

(b) Modularity = 0.933809

Figure 32: Original Network with Community structure using fast greedy algorithm


New network with the same Degree Sequence
Community Structure of new Network

(a)

(b) Modularity = 0.832903

Figure 33: Network after stub matching with same degree sequence with Community structure using fast greedy algorithm

The modularity of the two networks is shown along with the community structure. It can be seen that for a network obtained after the stub matching a lot of the dangling nodes are left unpaired which reduces the number of connections in the communities. The major differences between the two networks are as follows:

- As the stub-matching procedure is random, it may not generate a fully connected. However, the preferential attachment model begins with an initial connected network and every new node connects to a node already connected in the network which ensure that the network is always connected as it grows.
- In comparison with the preferential attachment model, the modularity of the the network generated from the stub-matching procedure is observed to be lower. This can be a direct consequence of the slight reduction in the sparseness of inter-module connections resulting from the stub-matching procedure thereby causing the relatively lower modularity.

1.3 Create a modified preferential attachment model that penalizes the age of a node

1.3(a) Each time a new vertex is added, it creates m links to old vertices and the probability that an old vertex is cited depends on its degree, preferential attachment, and age. Produce such an undirected network with 1000 nodes and parameters $m = 1$, $\alpha = 1$, $\beta = 1$, and $a = c = d = 1$, $b = 0$. Plot the degree distribution. What is the power law exponent?

In this part, we penalize the age of a node. The distribution for a such a network is given as follows:

$$P(i) \sim (ck_i^{-\alpha} + a)(dl_i^{-\beta} + b), \tag{5}$$

where, k_i is the degree and l_i is the age of the vertex i. α is the preferential attachment exponent. β is the exponent of the aging. a is the degree-dependent part of the 'attractiveness' of the vertices with no adjacent edges and b is the age-dependent part of the 'attractiveness' of the vertices with age zero. c and d are coefficients of degree and age. For the given parameters $\beta = -1$ and $\alpha = 1$, probability of a link to an old node is directly proportional to its degree and inversely proportional to its age. The network obtained when the age of the node as is penalized in this way is as shown in Figure 36a. The degree distribution, log-log plot and the histogram are as show in Figures 34 and 35.

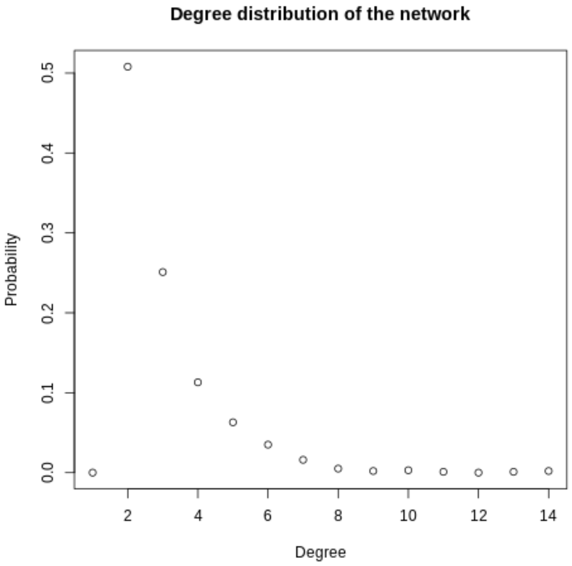
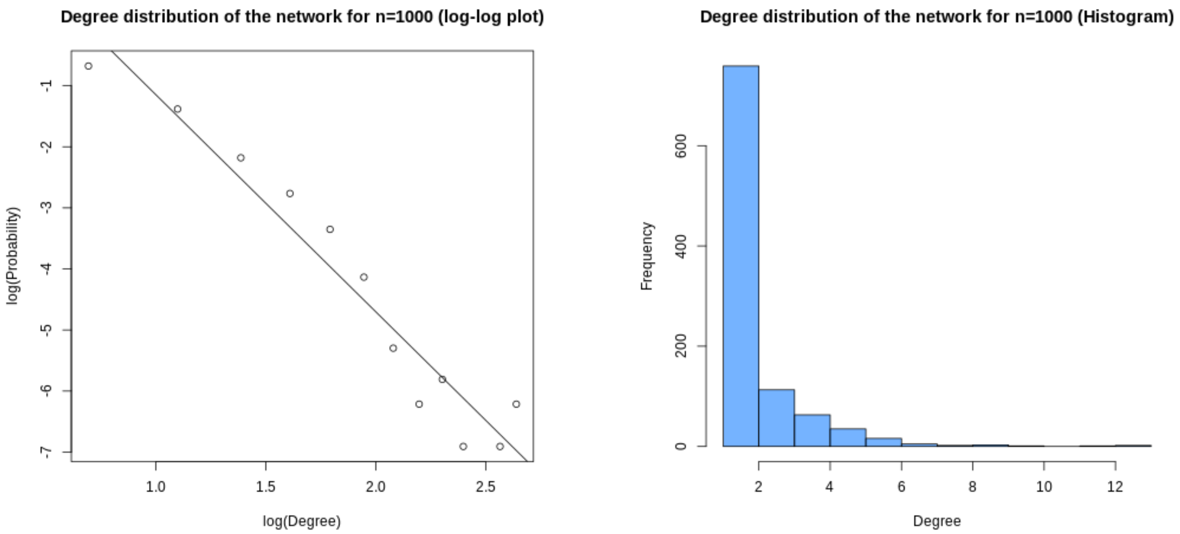


Figure 34: Degree of distribution of network



(a) Slope=-3.555, Intercept=2.406

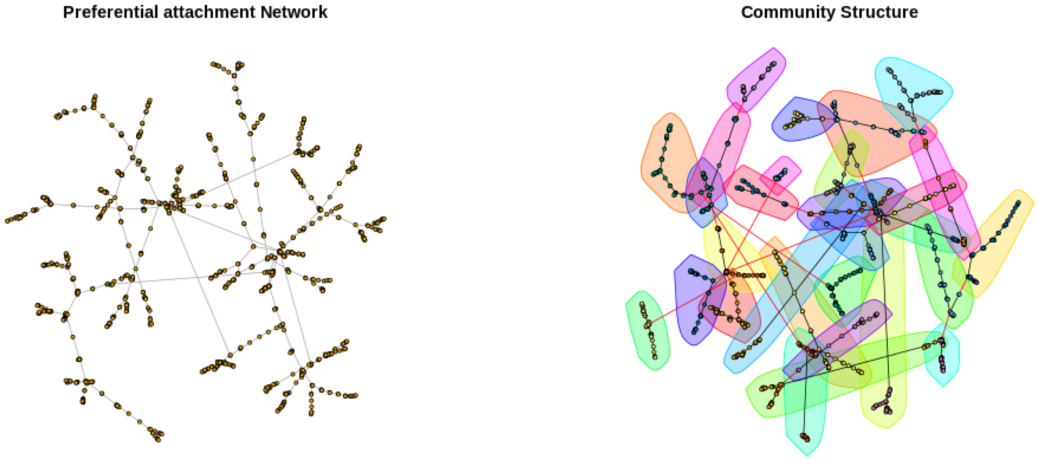
(b)

Figure 35: Degree of distribution of network.

The slope or the power-law component obtained from the log-log plot is -3.555. This is in good agreement with the theoretical value.

1.3(b) Use fast greedy method to find the community structure. What is the modularity?

The network obtained after penalizing age is as shown in figures below. The modularity of the community structure is 0.935354. Since the modularity is a measure of the community structure of the network, we can conclude that penalizing the age of a node during network creation constructs a networks with strong modules or communities such that the connections within the community are dense and the connections between nodes of different communities are sparse.



(a)

(b) Modularity = 0.935354

Figure 36: Network and Community Structure when age of node is penalized.

2 Random Walk on Networks

2.1 Random walk on Erdos-Renyi networks

2.1(a) Create an undirected random network with 1000 nodes, and the probability p for drawing an edge between any pair of nodes equal to 0.01.

Using `sample_gnp`, we created an undirected Erdos-Renyi graph with $n = 1000$ and $p = 0.01$. Fig. 37 gives a visualization of the generated graph.

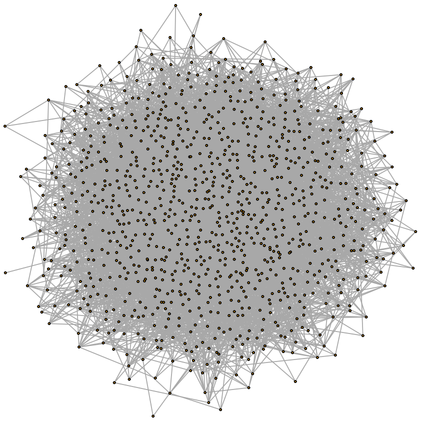
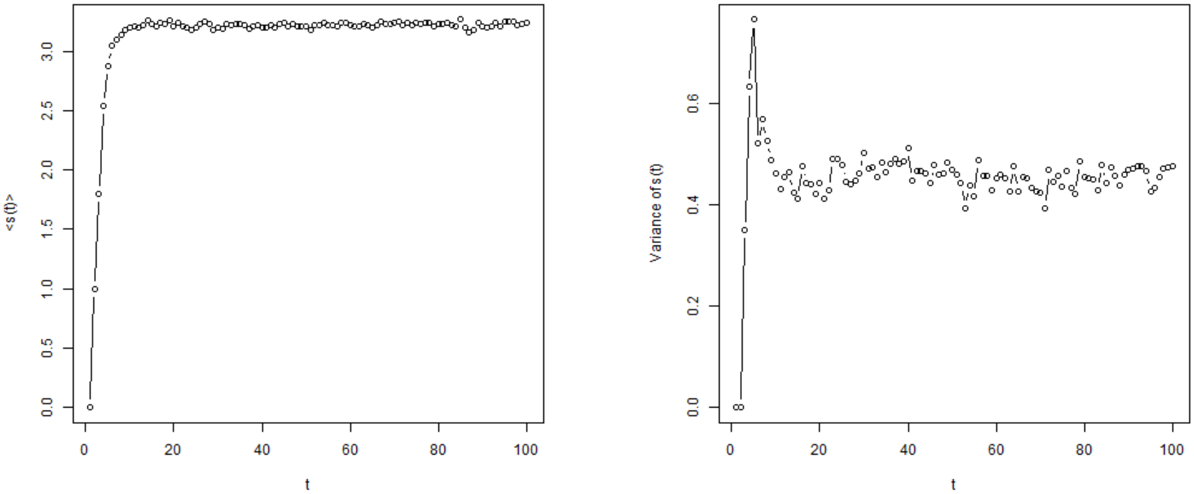


Figure 37: Plot of the undirected E-R network with $n = 1000$ and $p = 0.01$

2.1(b) Let a random walker start from a randomly selected node (no teleportation). We use t to denote the number of steps that the walker has taken. Measure the average distance (defined as the shortest path length) $\langle s(t) \rangle$ of the walker from his starting point at step t. Also, measure the variance $\sigma^2(t) = \langle (s(t) - \langle s(t) \rangle)^2 \rangle$ of this distance. Plot $\langle s(t) \rangle$ v.s. t and $\sigma^2(t)$ v.s. t. Here, the average $\langle \cdot \rangle$ is over random choices of the starting nodes.

We started a random walk that started at a randomly selected node in the Giant Connected Component (GCC) and let it run for $T = 100$ steps. For each step $t \leq T$, we measured the shortest distance $s(t)$ from the starting node. To get $\langle s(t) \rangle$ and $\sigma^2(t)$, we averaged and calculated the variance of $s(t)$ over 1000 such random walks. These results are summarized in 38.



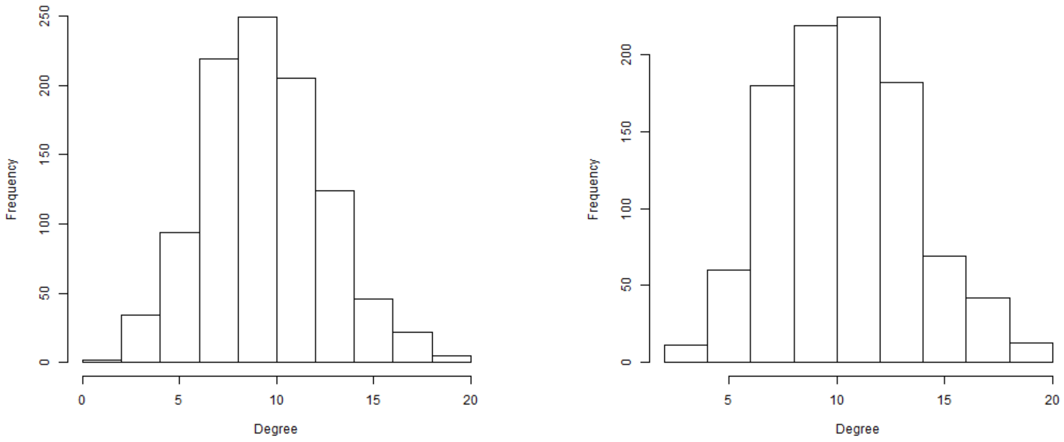
(a) $\langle s(t)\rangle$ v.s. t

(b) $\sigma^2(t)$ v.s. t

Figure 38: Variation of average and variance of the shortest distance to the walker from starting node with the number of steps ($n = 1000$, $p = 0.01$)

Observe that $\langle s(t)\rangle$ and $\sigma^2(t)$ reach a steady state around $t = t_{\mathrm{ss}} = 10$ where $\langle s(t_{\mathrm{ss}})\rangle \approx 3.2$ and $\sigma^2(t_{\mathrm{ss}}) \approx 0.42$.

2.1(c) Measure the degree distribution of the nodes reached at the end of the random walk. How does it compare to the degree distribution of graph?



(a) Degree distribution of the original network

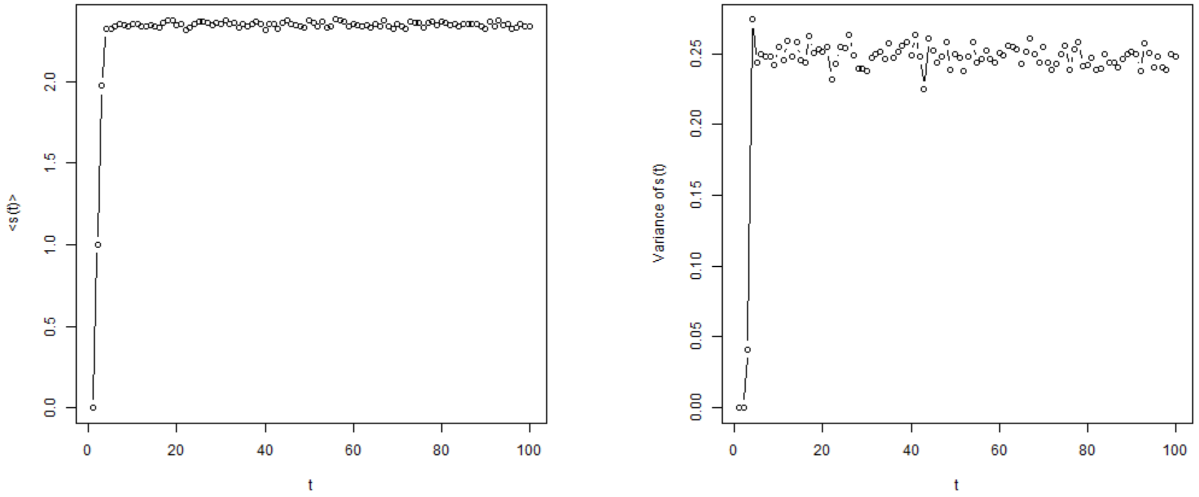
(b) Degree distribution of the nodes at the end of the random walk

Figure 39: Degree of distribution of the original network and the terminal nodes of the random walk ($n = 1000$, $p = 0.01$)

For each of the 1000 random walks run in 2.1(b), we calculated the degree of the node at the end of the walk (terminal node). The degree distribution of these terminal nodes is given in Fig. 39 (b) while the degree distribution of the original graph is given in Fig. 39 (a). We note that, approximately, both follow a similar binomial distribution, as expected for E-R networks.

2.1(d) Repeat 1(b) for undirected random networks with 10000 nodes. Compare the results and explain qualitatively. Does the diameter of the network play a role?

In this part, we ran a similar experiment to 2.1(b), but for an E-R network with $n = 10000$ and $p = 0.01$. The obtained $\langle s(t) \rangle$ and $\sigma^2(t)$ are summarized in Fig. 40.



(a) $\langle s(t)\rangle$ v.s. t

(b) $\sigma^2(t)$ v.s. t

Figure 40: Variation of average and variance of the shortest distance to the walker from starting node with the number of steps ($n = 10000$, $p = 0.01$)

Observe that for this network, $\langle s(t)\rangle$ and $\sigma^2(t)$ reach a steady-state around $t = t_{\text{ss}} = 5$ where $\langle s(t_{\text{ss}})\rangle \approx 2.35$ and $\sigma^2(t_{\text{ss}}) \approx 0.25$. In Table 7, we compare these values to the corresponding values of the E-R network created in 2.1(a).

n	Diameter	t_{ss}	$\langle s(t_{\text{ss}})\rangle$	$\sigma^2(t_{\text{ss}})$
1000	5	10	3.2	0.42
10000	3	5	2.35	0.25

Table 7: The number of steps to reach the steady state t_{ss}, $\langle s(t_{ss})\rangle$ and $\sigma^2(t_{ss})$, for networks with $n = 1000$ and $n = 10000$

Interestingly, when n is increased, the number of steps to reach the steady state t_{ss} decreases and so do the values of $\langle s(t_{\text{ss}})\rangle$ and $\sigma^2(t_{\text{ss}})$. Qualitatively, this means that the larger network converges to the steady state sooner, and that, at convergence, there is less spread in the shortest distance to the starting node. Although this might seem counter-intuitive, we note that although n increased from 1000 to 10000, the diameter of the network decreased from 5 to 3. This means that, on average, the network is more concentrated around the starting node, resulting in a lower number of steps to convergence.

2.2 Random walk on networks with fat-tailed degree distribution

2.2(a) Generate an undirected preferential attachment network with 1000 nodes, where each new node attaches to $m = 1$ old nodes.

Using `sample_pa`, we created an undirected preferential attachment network with $n = 1000$ and $m = 1$. Fig. 41 gives a visualization of the generated graph.

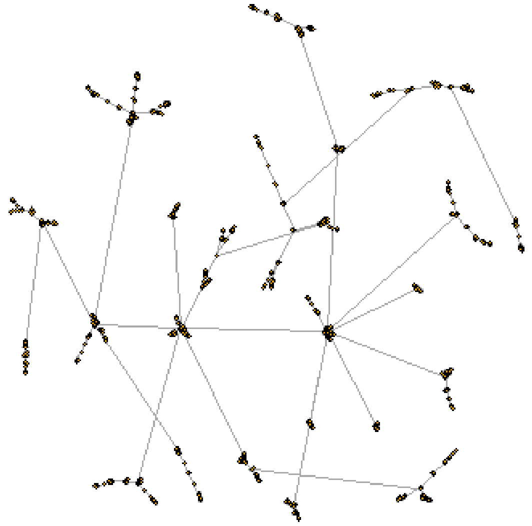
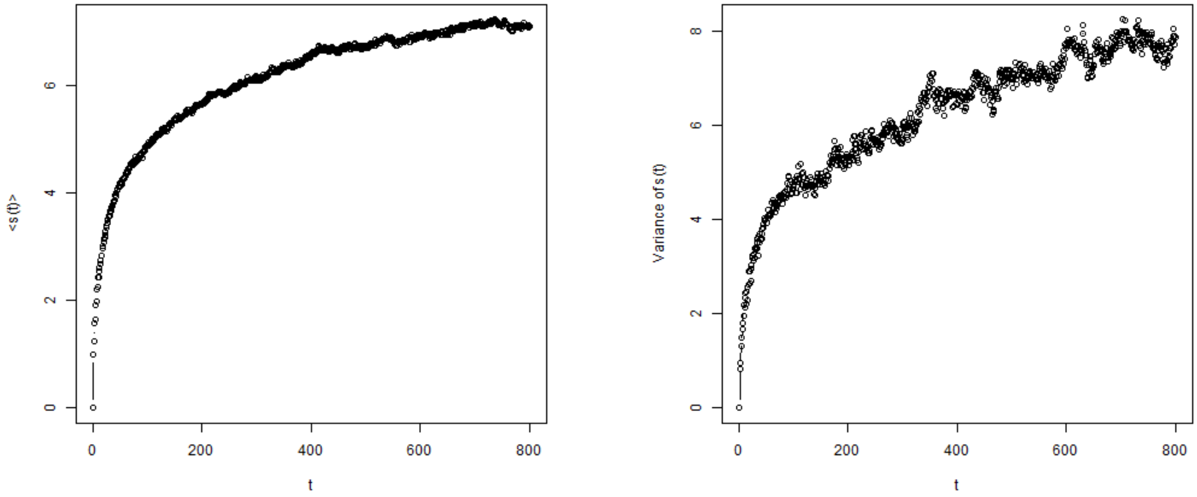


Figure 41: Plot of the undirected preferential attachment network with $n = 1000$ and $m = 1$

2.2(b) Let a random walker start from a randomly selected node. Measure and plot $\langle s(t) \rangle$ v.s. t and $\sigma^2(t)$ v.s. t.

We started a random walk that started at a randomly selected node in the network and let it run for $T = 800$ steps. For each step $t \leq T$, we measured the shortest distance $s(t)$ from the starting node. To get $\langle s(t) \rangle$ and $\sigma^2(t)$, we averaged and calculated the variance of $s(t)$ over 1000 such random walks. These results are summarized in 42.



(a) $\langle s(t)\rangle$ v.s. t

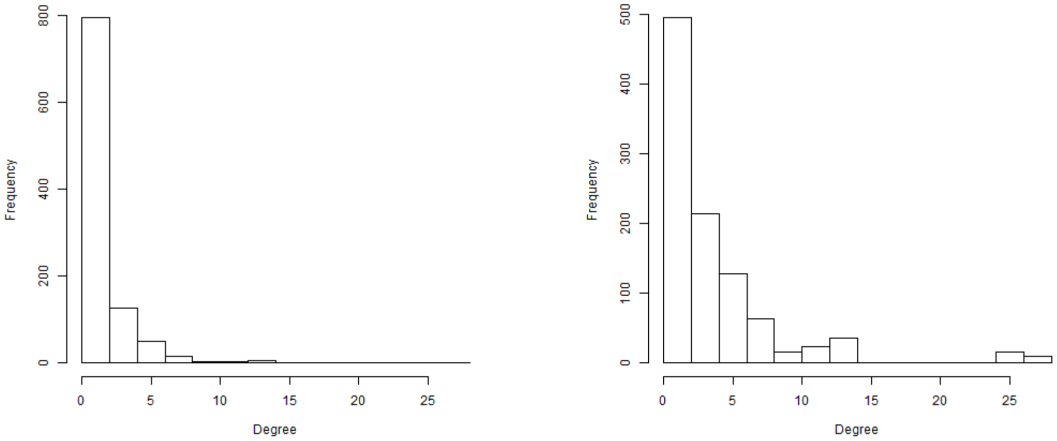
(b) $\sigma^2(t)$ v.s. t

Figure 42: Variation of average and variance of the shortest distance to the walker from starting node with the number of steps ($n = 1000$, $m = 1$)

Observe that $\langle s(t)\rangle$ and $\sigma^2(t)$ reach a steady state around $t = t_{\text{ss}} = 700$ where $\langle s(t_{\text{ss}})\rangle \approx 3.8$ and $\sigma^2(t_{\text{ss}}) \approx 2.9$.

2.2(c) Measure the degree distribution of the nodes reached at the end of the random walk on this network. How does it compare with the degree distribution of the graph?

For each of the 1000 random walks run in 2.2(b), we calculated the degree of the node at the end of the walk (terminal node). The degree distribution of these terminal nodes is given in Fig. 43 (b) while the degree distribution of the original graph is given in Fig. 43 (a). We note that, approximately, both follow a similar power law distribution, as expected for PA networks.



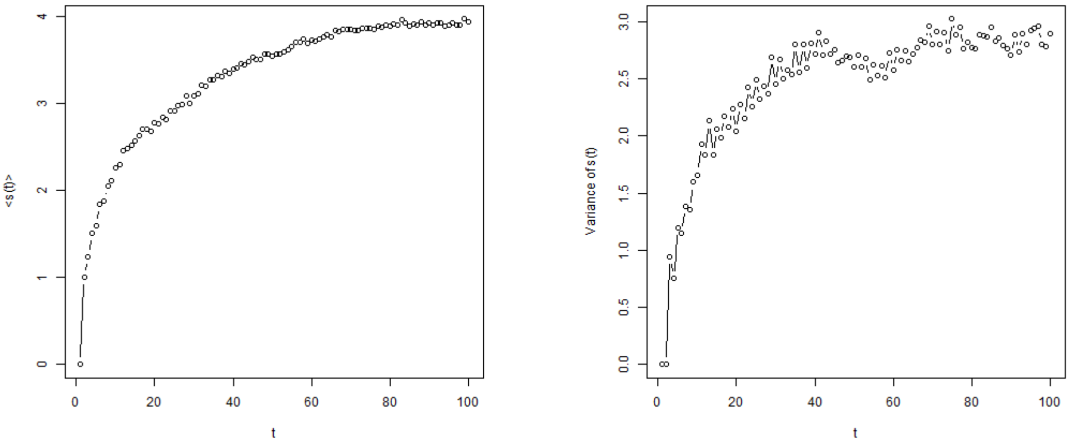
(a) Degree distribution of the original network

(b) Degree distribution of the nodes at the end of the random walk

Figure 43: Variation of average and variance of the shortest distance to the walker from starting node with the number of steps ($n = 100$, $m = 1$)

2.2(d) Repeat 2(b) for preferential attachment networks with 100 and 10000 nodes, and $m = 1$. Compare the results and explain qualitatively. Does the diameter of the network play a role?

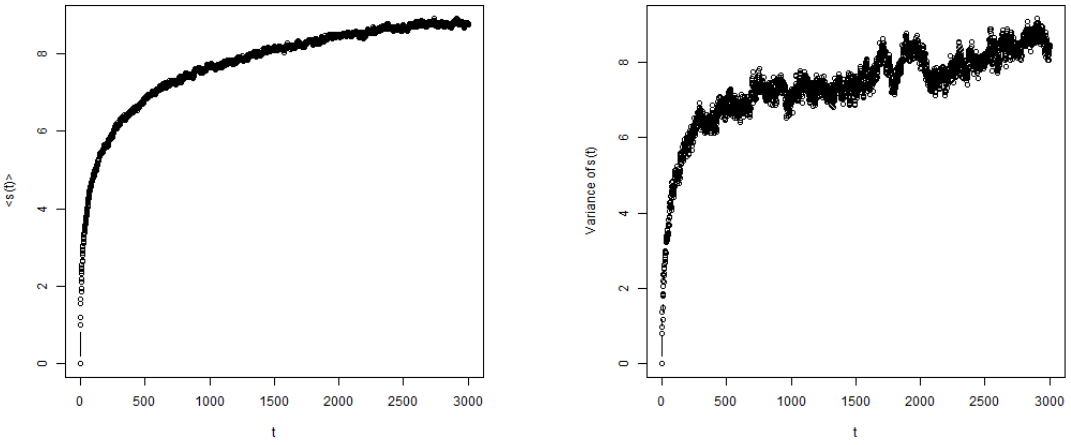
In this part, we ran a similar experiment to 2.2(b), for $n = 100$ and $n = 10000$. For $n = 100$, T was set to 100 and for $n = 10000$, T was set to 3000. The obtained $\langle s(t) \rangle$ and $\sigma^2(t)$ are summarized in Fig. 44 for $n = 100$ and in Fig. 45 for $n = 10000$.



(a) $\langle s(t)\rangle$ v.s. t

(b) $\sigma^2(t)$ v.s. t

Figure 44: Variation of average and variance of the shortest distance to the walker from starting node with the number of steps ($n = 100$, $m = 1$)



(a) $\langle s(t)\rangle$ v.s. t

(b) $\sigma^2(t)$ v.s. t

Figure 45: Variation of average and variance of the shortest distance to the walker from starting node with the number of steps ($n = 10000$, $m = 1$)

Observe that for $n = 100$, $\langle s(t)\rangle$ and $\sigma^2(t)$ reach a steady state around $t = t_{\text{ss}} = 80$ where $\langle s(t_{\text{ss}})\rangle \approx 3.8$ and $\sigma^2(t_{\text{ss}}) \approx 2.9$. For $n = 10000$, $\langle s(t_{\text{ss}})\rangle$ and $\sigma^2(t_{\text{ss}})$ reach a steady state around $t = t_{\text{ss}} = 5$ where $\langle s(t)\rangle \approx 8.7$ and $\sigma^2(t) \approx 8.5$. In Table 8, we compare these values to the corresponding values of the PA network created in 2.2(a).

n	Diameter	t_{ss}	$\langle s(t_{\text{ss}})\rangle$	$\sigma^2(t_{\text{ss}})$
100	10	80	3.8	2.9
1000	23	700	7	7.5
10000	29	2700	8.7	8.5

Table 8: The number of steps to reach the steady state t_{ss}, $\langle s(t_{ss})\rangle$ and $\sigma^2(t_{ss})$, for networks with $n = 100, 1000, 10000$

When n is increased, the number of steps to reach the steady state t_{ss} increases and so do the values of $\langle s(t_{\text{ss}})\rangle$ and $\sigma^2(t_{\text{ss}})$. Qualitatively, this means that the larger network converges to the steady state slower, and that, at convergence, there is more spread in the shortest distance to the starting node. Looking at the corresponding diameters, we see that as n increased, the diameter of the network also. So we see that the pattern we observed in 2.2(d) is followed here as well.

2.3 PageRank

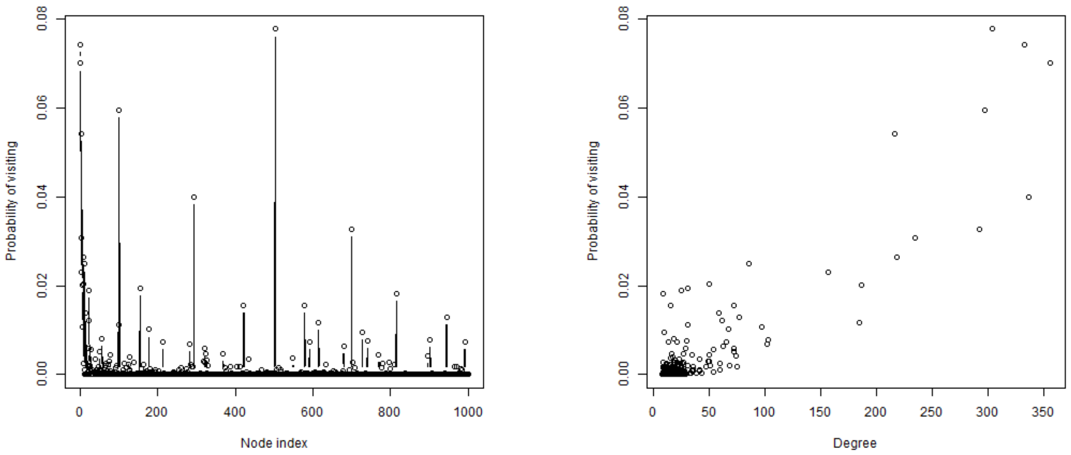
The PageRank algorithm, as used by the Google search engine, exploits the linkage struc- ture of the web to compute global "importance" scores that can be used to influence the ranking of search results. Here, we use random walk to simulate PageRank.

2.3(a) We are going to create a directed random network with 1000 nodes, using the preferential attachment model. Note that in a directed preferential attachment network, the out-degree of every node is m, while the in-degrees follow a power law distribution. One problem of performing random walk in such a network is that, the very first node will have no outbounding edges, and be a "black hole" which a random walker can never "escape" from. To address that, let's generate another 1000-node random network with preferential attachment model, and merge the two networks by adding the edges of the second graph to the first graph with a shuffling of the indices of the nodes. Create such a network using $m = 4$. Measure the probability that the walker visits each node. Is this probability related to the degree of the nodes?

To achieve this, first we create two PA networks with $n = 1000$ and $m = 4$ using `sample_pa`. Then we permute the nodes of the second network using `permute`, get the edge-list with the permuted node-indices using `as_edgelist` and add these edges to the first network using `add_edges`, to create a modified PA network with $n = 1000$ nodes and $|E| = 7980$ edges.

To calculate the node-visitation probabilities, we run 100 random walks each with 1000 steps, and each starting at an initial node selected uniformly at random. Then we count the number of times each node is visited, averaged over the 100 random walks. Note that we attempt to count visits only after each random walk has reached its steady-state—we can approximate this by only recording visits after the first $\ln n$ steps, where $\ln n$ is the number of steps required for exponential convergence.

The probability that the random walker visits each node is given in Fig. 46 (a). Fig. 46 (b) goes a step further and gives us an idea about the probability that the random walker visits nodes with a given degree. The Pearson correlation coefficient of this relationship is 0.9045546 and a linear fit gives a slope of 0.0001713408. The relatively high Pearson coefficient reveals an approximately proportional relationship. This makes sense because a node with a high in-degree should have a higher chance of being visited.



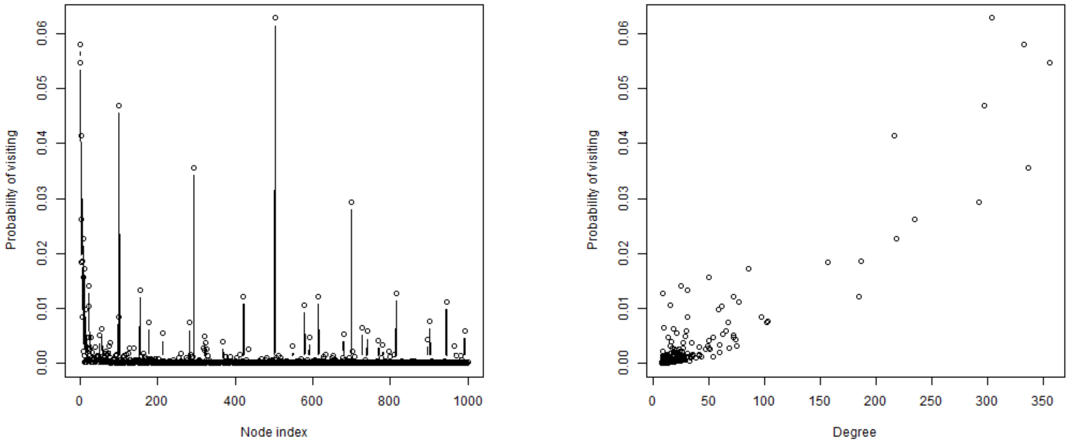
(a) Probability of random walker visiting each node

(b) Probability of random walker visiting nodes with given degree

Figure 46: Variation of the probability of random walker visitation with the node index and node degree (Modified PA network with $n = 1000$, $m = 4$. No teleportation.)

2.3(b) In all previous questions, we didn't have any teleportation. Now, we use a teleportation probability of $\alpha = 0.15$. By performing random walks on the network created in 3(a), measure the probability that the walker visits each node. Is this probability related to the degree of the node?

In this case, the experiment is run similar to 2.3(a), except that there is the ability of the random walk to teleport. This means that, at each step, the random walker chooses to visit one of its neighbors with probability $(1 - \alpha)$ and teleports with probability α, choosing the next node with equal probability. The probability that the random walker visits each node in this case is given in Fig. 47 (a). Fig. 47 (b) gives the probability that the random walker visits nodes with a given degree. The Pearson correlation coefficient of this relationship is 0.9288324 and a linear fit gives a slope of 0.0001397091. The relatively lower slope compared to 2.3(a) is consistent with the observation that the probability of visitation for nodes with higher degrees has decreased comparatively in 47 (b). This is the compromise of lower-degree-nodes now having a higher probability of being visited due to teleportation.



(a) Probability of random walker visiting each node

(b) Probability of random walker visiting nodes with given degree

Figure 47: Variation of the probability of random walker visitation with the node index and node degree (Modified PA network with $n = 1000$, $m = 4$. With teleportation ($\alpha = 0.15$).)

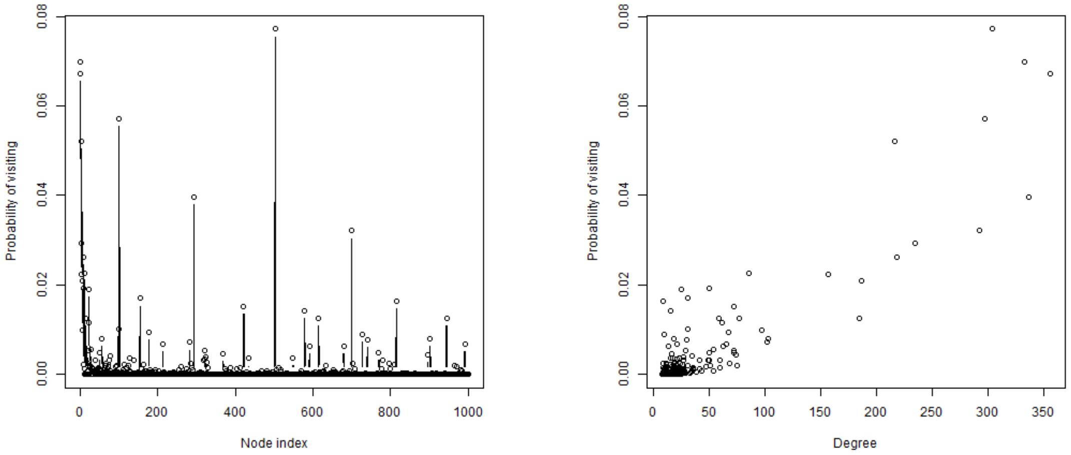
2.4 Personalized PageRank

While the use of PageRank has proven very effective, the web's rapid growth in size and diversity drives an increasing demand for greater flexibility in ranking. Ideally, each user should be able to define their own notion of importance for each individual query.

2.4(a) Suppose you have your own notion of importance. Your interest in a node is proportional to the node's PageRank, because you totally rely upon Google to decide which website to visit (assume that these nodes represent websites). Again, use random walk on network generated in question 3 to simulate this personalized PageRank. Here the teleportation probability to each node is proportional to its PageRank (as opposed to the regular PageRank, where at teleportation, the chance of visiting all nodes are the same and equal to $\frac{1}{N}$). Again, let the teleportation probability be equal to $\alpha = 0.15$. Compare the results with 3(a).

This experiment is run similar to 2.3(b), except that when the random walker teleports, the probability of any node being choses as the next is proportional to its PageRank. To achieve this, we can calculate the PageRanks using the `page_rank` function.

The probability that the random walker visits each node in this case is given in Fig. 48 (a). Fig. 48 (b) gives the probability that the random walker visits nodes with a given degree. The Pearson correlation coefficient is 0.9107512 and the slope of the linear fit is 0.0001660616. Note that in 48 (b) 1. the slope has increased from that of 47 (b) to almost that of 46 (b) (where there was no teleportation) and 2. the probability of visitation for nodes with higher degrees has also increased. This is because teleportations now have a higher probability of landing on nodes with higher PageRanks, which in turn have higher degrees. This is similar to the situation in 2.3(a), so we expect a relationship similar to 46 (b).



(a) Probability of random walker visiting each node

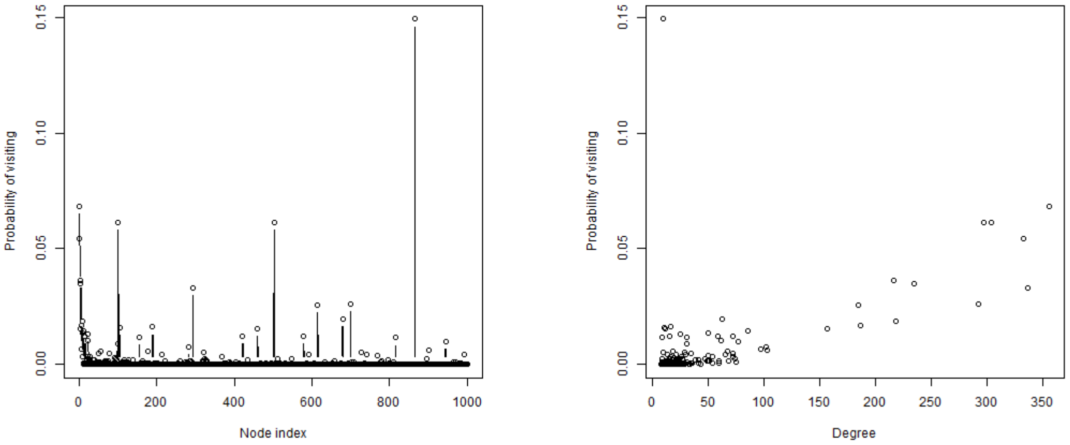
(b) Probability of random walker visiting nodes with given degree

Figure 48: Variation of the probability of random walker visitation with the node index and node degree (Modified PA network with $n = 1000$, $m = 4$. With teleportation ($\alpha = 0.15$, Teleport mode=Proportional Pagerank).)

2.4(b) Find two nodes in the network with median PageRanks. Repeat part 4(a) if teleportations land only on those two nodes (with probabilities 1/2, 1/2). How are the PageRank values affected?

The only difference here relative to 2.4(a) is that when the random walker teleports, it only lands at one of the two nodes with median PageRanks, choosing one over the other with equal probability.

The probability that the random walker visits each node in this case is given in Fig. 49 (a). Fig. 49 (b) gives the probability that the random walker visits nodes with a given degree. The Pearson correlation coefficient is 0.6448008 and the slope of the linear fit is 0.0001461807. A sharp decrease in the Pearson coefficient is immediately noteworthy. This is due to the outliers with much higher visitation probability, visible at the North-West corner of Fig. 49 (b), representing the two nodes with median PageRanks. The rest of the data-points are expected to follow a relationship similar to 2.3(a), because teleportation has no effect on them.



(a) Probability of random walker visiting each node

(b) Probability of random walker visiting nodes with given degree

Figure 49: Variation of the probability of random walker visitation with the node index and node degree (Modified PA network with $n = 1000$, $m = 4$. With teleportation ($\alpha = 0.15$, Teleport mode=Median Pagerank).)

To investigate the effect of the random walk on the PageRank, we plot Fig. 50.

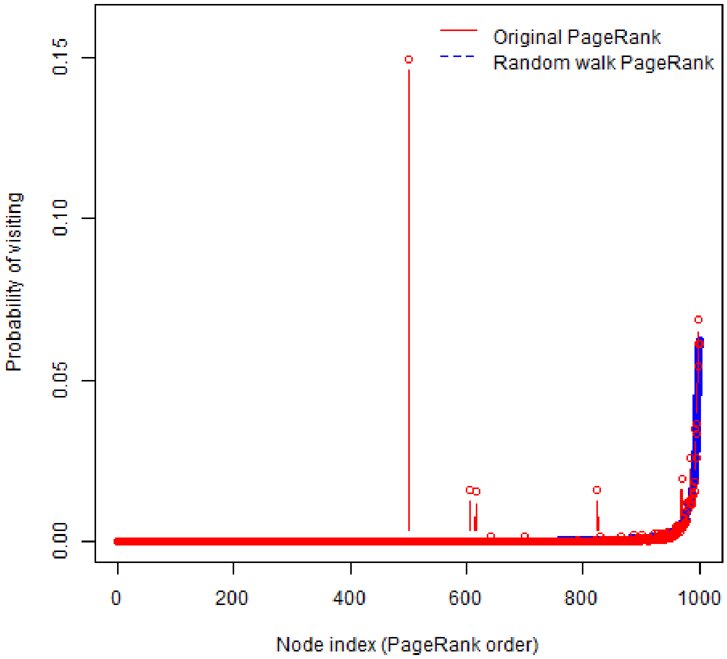


Figure 50: Probability of random walker visiting each node, when Node Index is assigned in ascending order of initial PageRank

Here, we have assumed the steady-state node visitation probabilities of the random walker to be the changed PageRank values. The Node Index is assigned based on the ascending order of the PageRanks of the original graph—so the huge spike at the middle corresponds to the two nodes with median PageRanks. An interesting observation is that there is a noticeable increase for a few other nodes as well. These could be nodes in the immediate neighborhood of the median nodes, most probably ones which have an in-link from one of them. When the median nodes get visited, the nodes in this neighborhood gets visited next, so we can expect them to get a boost in PageRank as well.

2.4(c) More or less, 4(b) is what happens in the real world, in that a user browsing the web only teleports to a set of trusted web pages. However, this is against the assumption of normal PageRank, where we assume that people's interest in all nodes are the same. Can you take into account the effect of this self-reinforcement and adjust the PageRank equation?

Denote by A the Node-Node incidence matrix of a directed graph, by P the Node transition matrix for a random walker on the graph (with teleportation probability α) and by $k_{\text{out}}(i)$ the out-degree of node i. Also let the P_{ij} element of P represent the probability of landing on node j after the random walker has landed on node i. Then for a standard teleporting random walker,

$$P_{ij} = (1-\alpha)\frac{1}{k_{\text{out}}(i)}A_{ij} + \alpha\frac{1}{n} \tag{6}$$

where $n =$ Number of nodes in the graph.
Now consider the case where we only teleport to some set T of trusted nodes in the graph with equal probability, and do not teleport to other nodes at all. Then the expression for P_{ij} changes to

$$P_{ij} = \begin{cases} (1-\alpha)\dfrac{1}{k_{\text{out}}(i)}A_{ij} + \alpha\dfrac{1}{|T|} & i \in T \\ (1-\alpha)\dfrac{1}{k_{\text{out}}(i)}A_{ij} & i \notin T \end{cases} \tag{7}$$

At equilibrium, for every node i,

$$\pi(i) = \sum_{j=1}^{n} P_{ji}\pi(j) \tag{8}$$

So the PageRank equation becomes (for every node i),

$$\pi(i) = \begin{cases} (1-\alpha)\displaystyle\sum_{j=1}^{n}\frac{1}{k_{\text{out}}(j)}A_{ji}\pi(j) + \alpha\frac{1}{|T|} & i \in T \\ (1-\alpha)\displaystyle\sum_{j=1}^{n}\frac{1}{k_{\text{out}}(j)}A_{ji}\pi(j) & i \notin T \end{cases} \tag{9}$$

In particular, for 2.4(b), $T = \{i|\text{Node } i \text{ is one of the two median nodes with respect to PageRank}\}$ where $|T| = 2$.