Large-Scale Data Mining: Models and Algorithms
ECE 232E Spring 2018

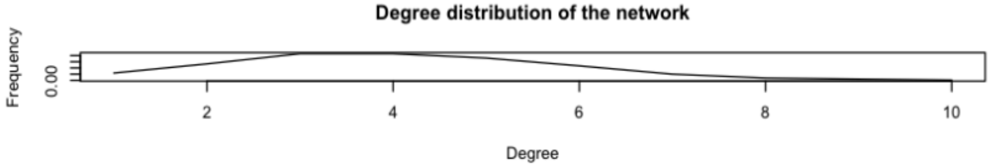
Project 1
Random Graphs and Random Walks

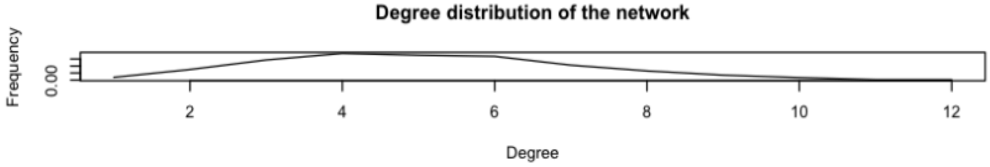
Qinyi Tang (204888348)
Shuo Bai (505032786)
Jinxi Zou (605036454)
Xuan Hu (505031796)

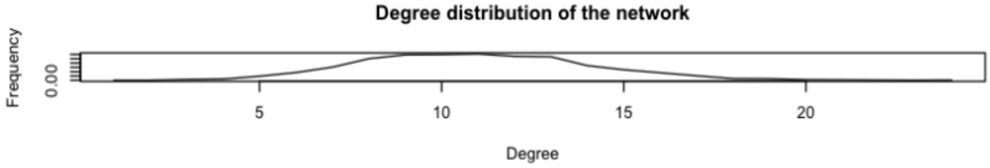
1 Generating Random Networks

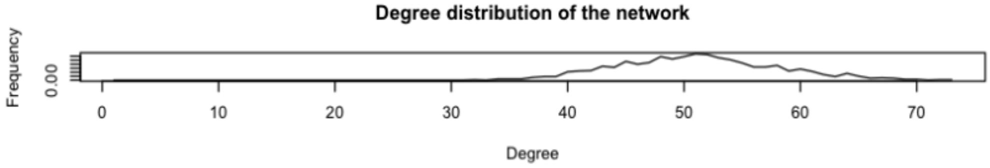
1.Create random networks with E-R model

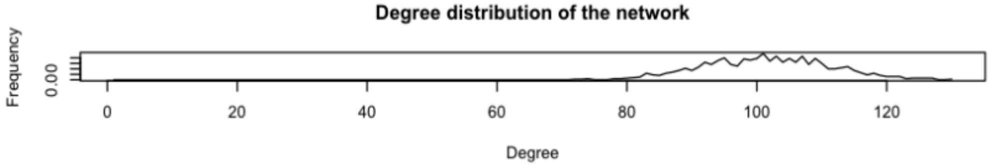
(a)From the following result, we can observe that as the probabilities increase. There will be more possible for random two nodes get connected and thus the edge number will get larger. This will cause a high degree distribution.





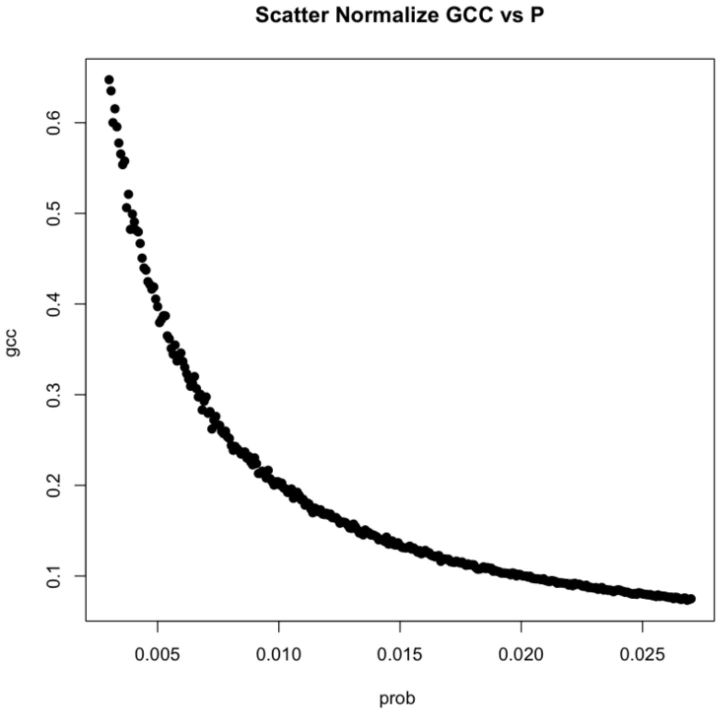






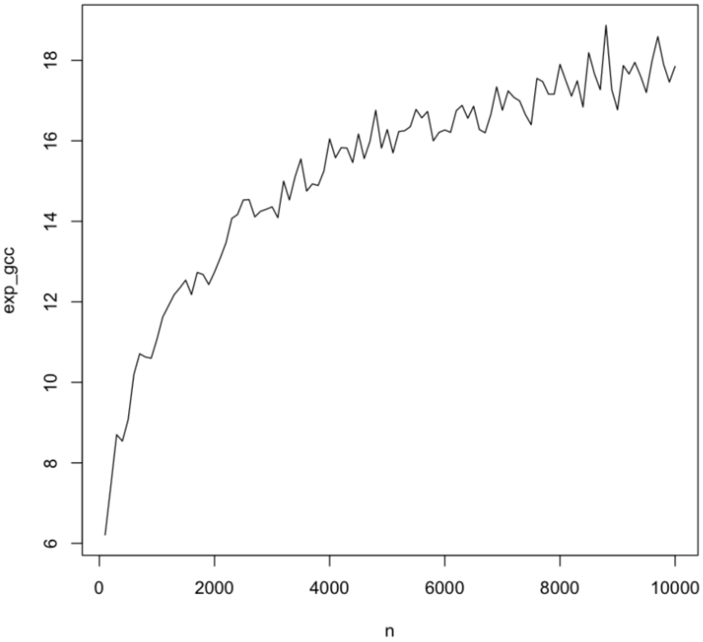
(b)The diameter for P = 0.003,0.004,0.01,0.05,0.1 is 15,11,5,3,3. It is easy to understand why. As the probabilities go higher. Two nodes have more high possibility to get connected, it will decrease the graph geodesic heavily.

(c)As derived in class, when the probability increase to a high value, the scatter will show a linear change, the phase change point is around 0.02.

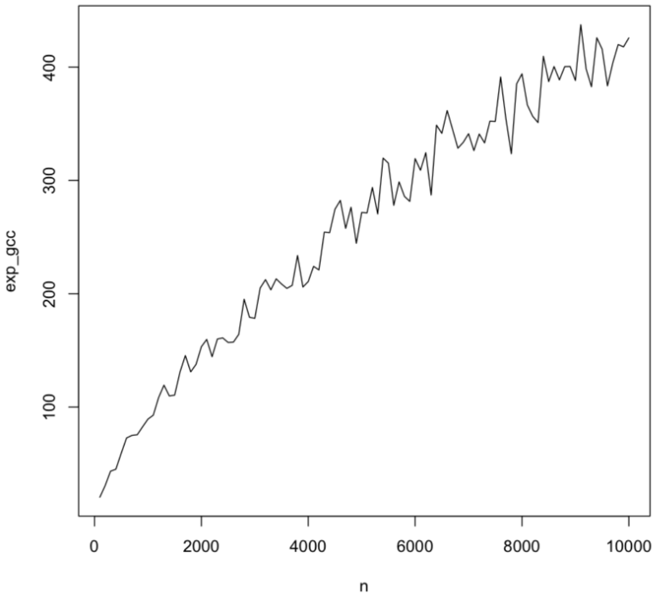


(d)

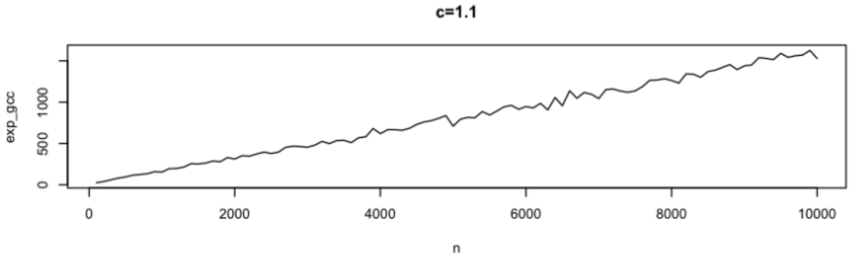
1) As the nodes number goes larger, the expected gcc size also increases, the trend is more close to a log function.

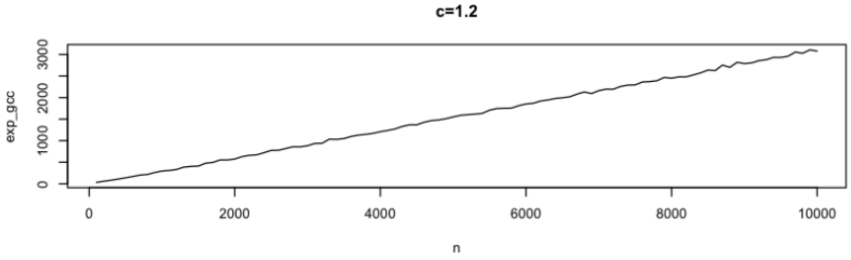


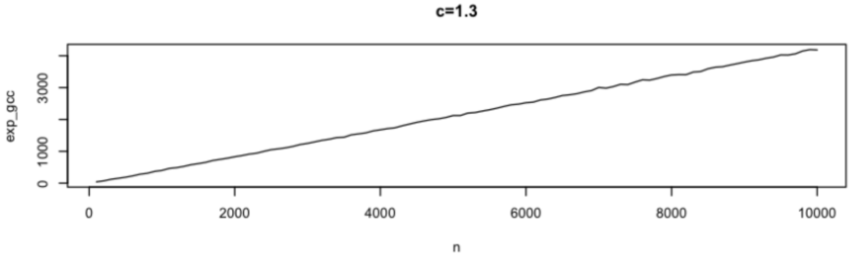
2) Comparing with the previous result, this trends is more close to a linear related function between n and gcc size



3) We draw the same plots for 1.1, 1.2, 1.3. The result is shown as following: as the nodes number increasing, the trend is more closer to a line.

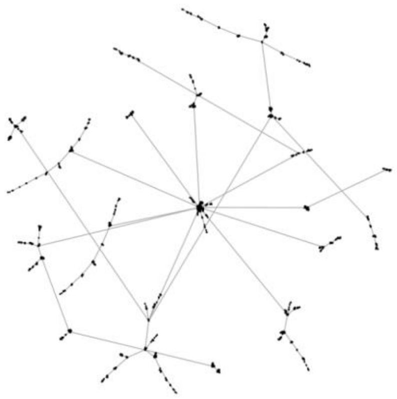






2.create networks using preferential attachment model

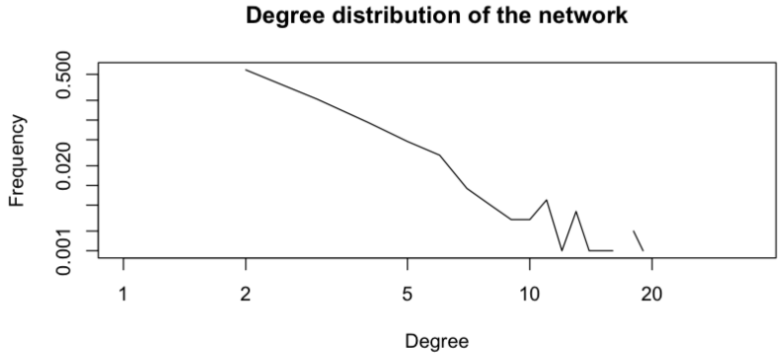
(a) Yes, one of them looks like the following figure:

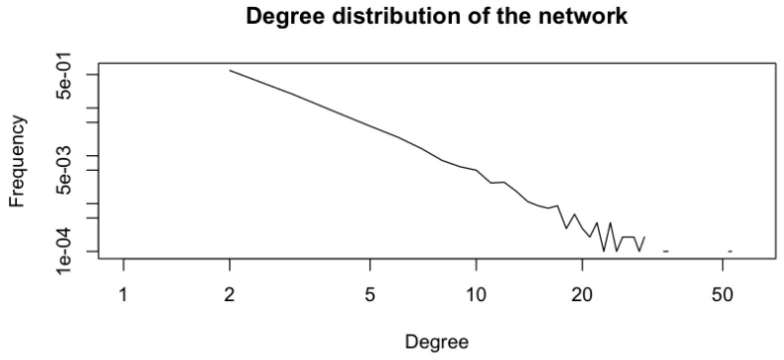


(b)The modularity for the model above is 0.928968007046085.

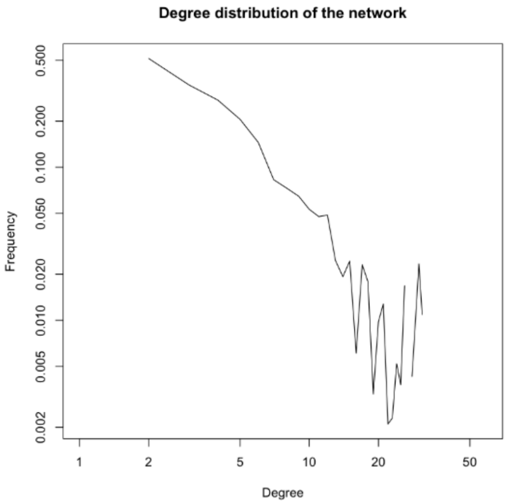
(c)When we create a large scale nodes of 10000 using this model, we get a higher modularity which is 0.977715728368517.

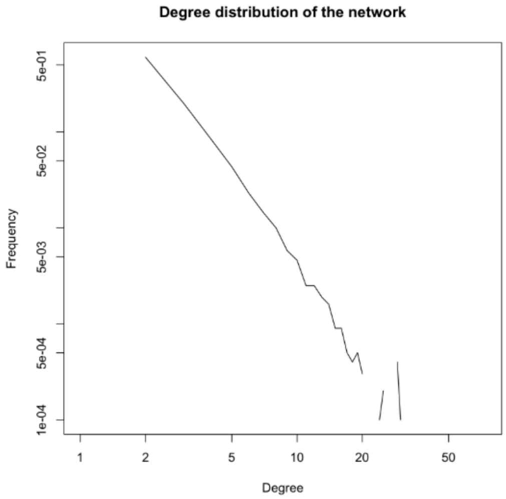
(d)The degree distribution from top to bottom is for 1000 and 10000 separately. For 1000 the slope is (0.5-0.2)/6 = 0.05. For 10000 the slope is (0.5-0.005)/8 = 0.062



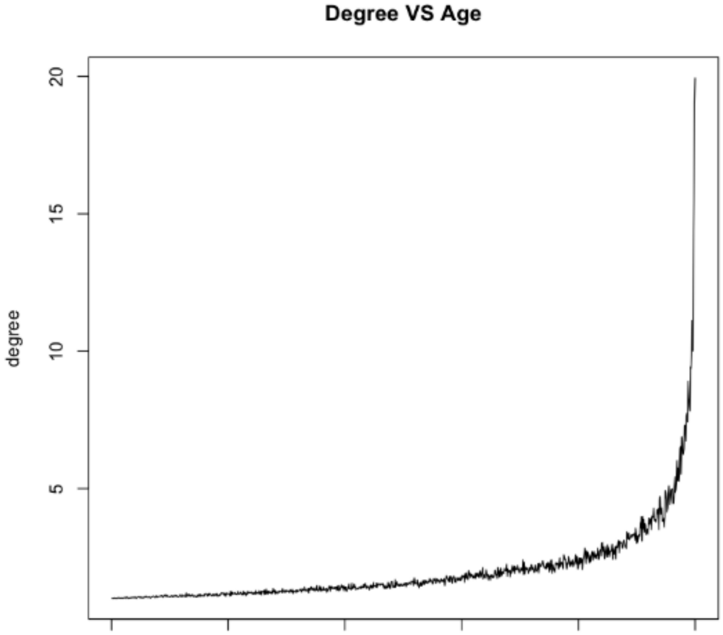


(e)Instead of using the whole nodes set, we randomly pick nodes to observe the degree distribution. According to the definition, the distribution is the fraction of number of certain degree nodes to all nodes. We pick n = 10000 as an example, the comparison is shown as following: The left is random-picking policy result. The curve is not an linear relation between degree and frequencies in log-scale.

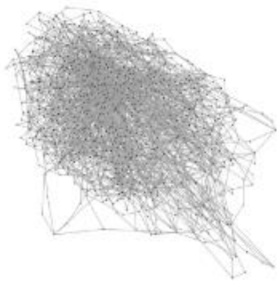


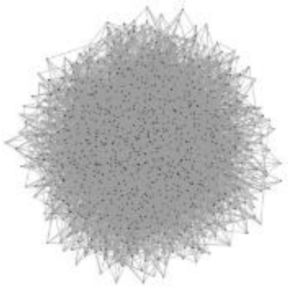


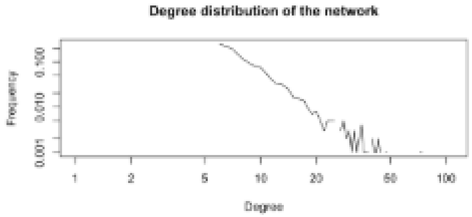
(f)In this part, we are asked to observe the simple relationship between age and degree. The age is simply total node number minus the index which means the less index the larger age. As we expected, if a node exists for a longer time, it will have higher degree.

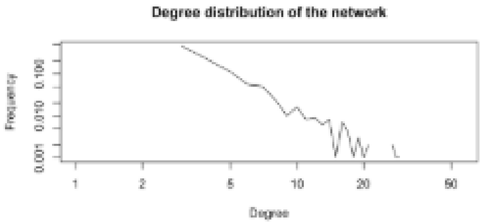


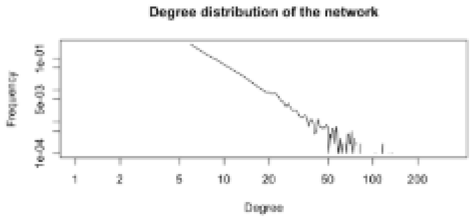
(g) We repeat the previous procedure for m = 2 and m = 5. When m = 2, the modularity is 0.517 and 0.531 for 1000 and 10000 separately. When m = 5, the modularity is 0.283 and 0.277 for 1000 and 10000 separately. When we draw the community plot for m = 1 , 2 and 5 separately.The reason for a high m = 1 modularity is clear. The modularity is to evaluate the goodness of separation between different communities. The following picture is hard to detect the community, while m = 1 figure shown before is clear for community partition.

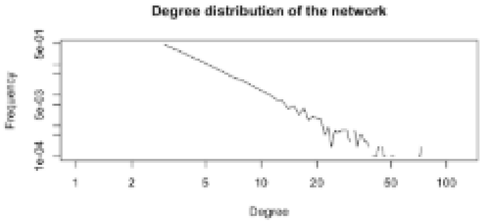


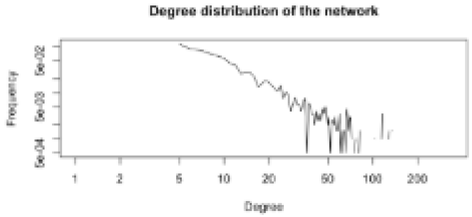


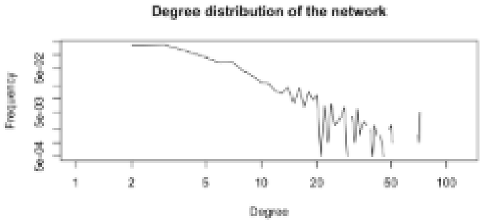


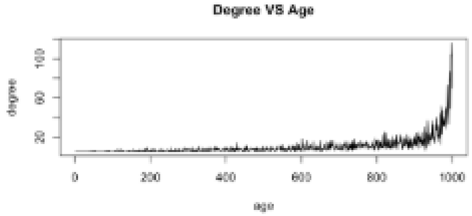


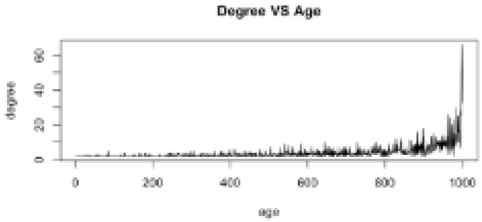




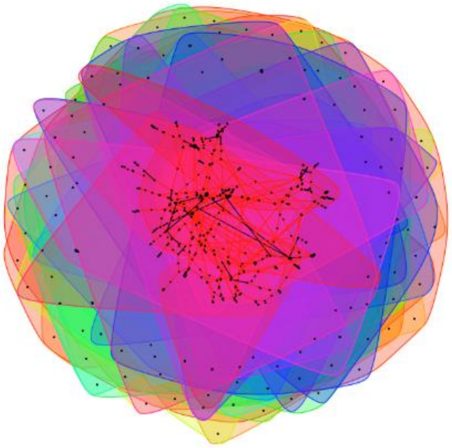


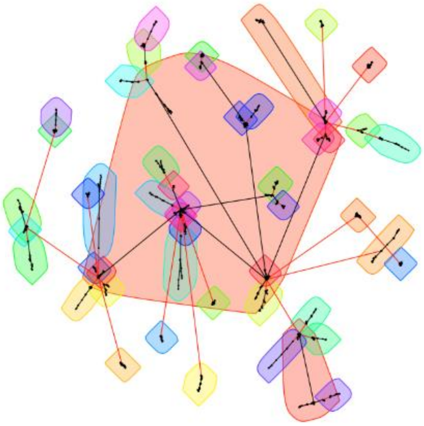






(h)After reconstruct a random figure with stub matching method with given degree sequence, we get the following result:

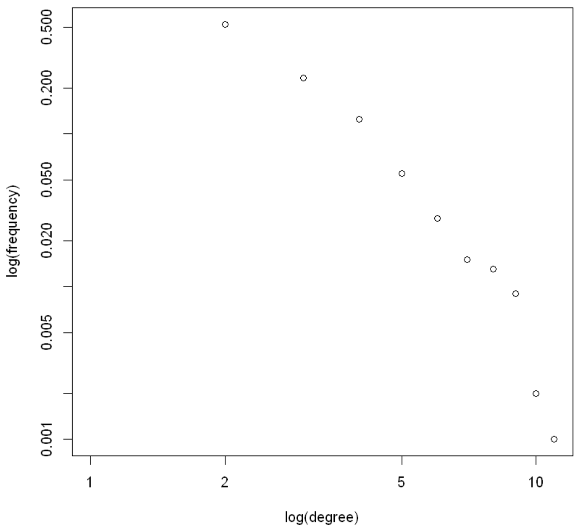




Using stub matching, there are a lot of communities and the graph is unconnected. While with preferential attachment procedure, the graph is connected because there will be an edge add to the existing nodes every time.

3. Modified preferential attachment model

(a)In this part, we created modified preferential attachment model which can penalizes the age of a node. The degree exponent of probability that a newly added vertex was set to 1 and age exponent of that was set to -1. Then degree distribution was calculated and shown in following figure. After transform x and y to log, the results of degree distribution are in a rough linear relationship, and the power law exponent was about -0.15.

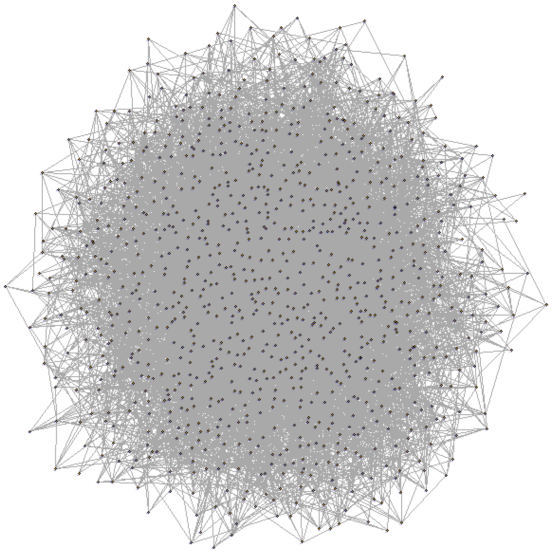


(b)After applying fast greedy method to find the community structure, the modularity was calculated. The result was 0.93.

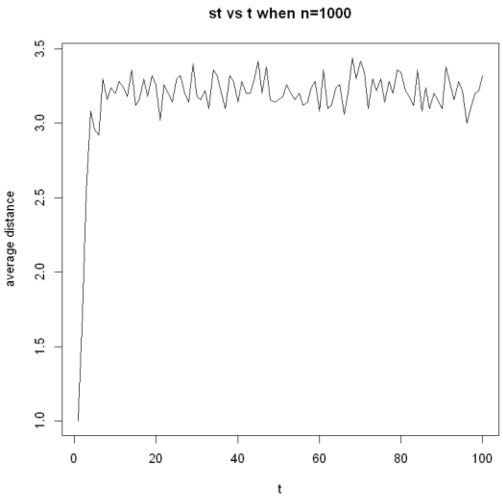
2 Random Walk on Networks

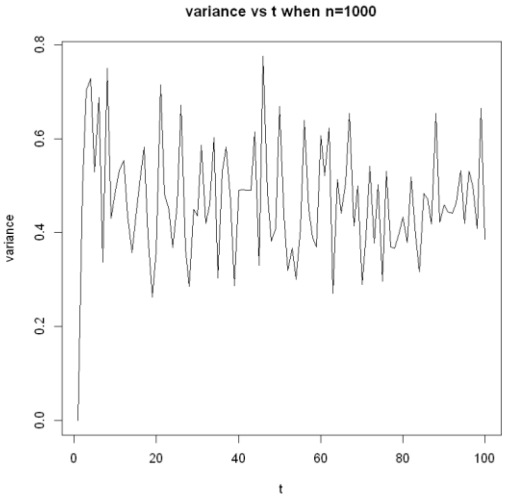
1. Random walk on Erdös-Rényi networks

(a) In this part, we created an undirected random network with 1000 nodes, and the probability for drawing an edge between any pairs of nodes equal to 0.01.

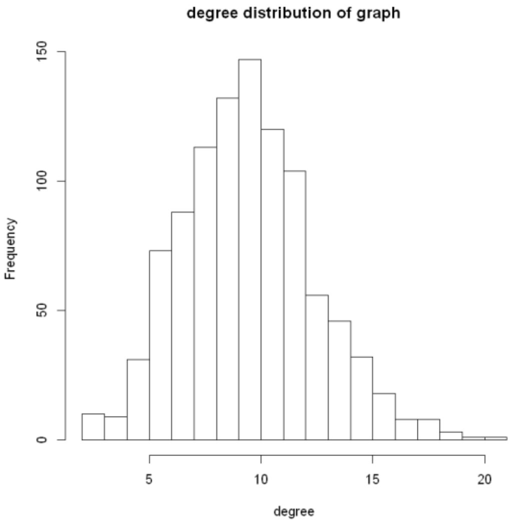


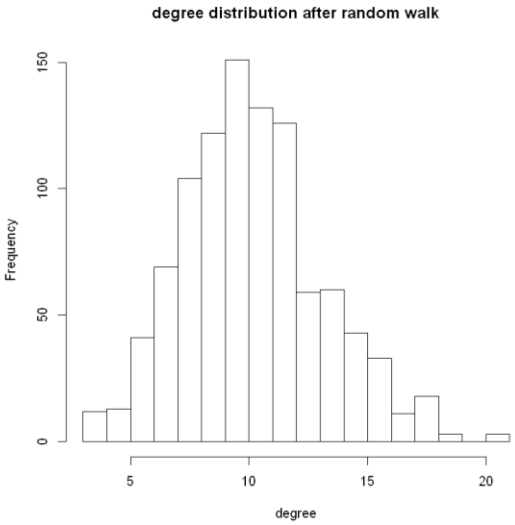
(b) Here we swept graph with step t from 1 to 100 and calculated their average distance and variance. As the pictures showing below, average distance of graph continuously rises at beginning. Then, after certain threshold, the value fluctuates around a stable status. The variance of distance shows same tendency except more violent fluctuations.


st vs t when n=1000

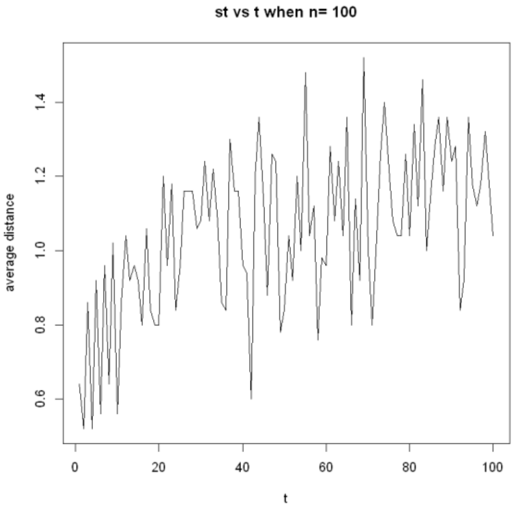

variance vs t when n=1000

(c) Here we plot degree distribution of graph and degree distribution after random walk separately. The step t we use here is 15. It seems there is no big difference between two pictures. Random walk doesn't influence degree distribution of graph obviously.

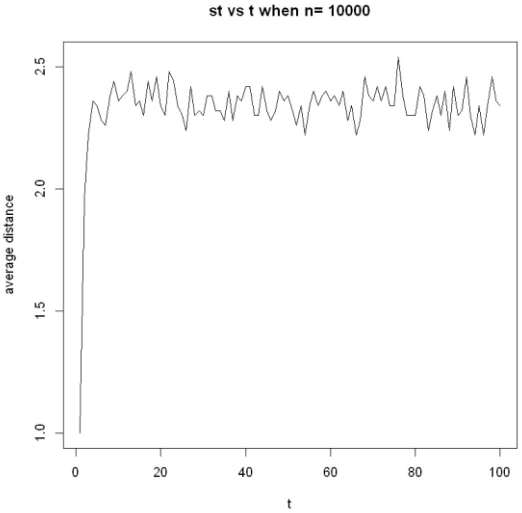

degree distribution of graph


degree distribution after random walk

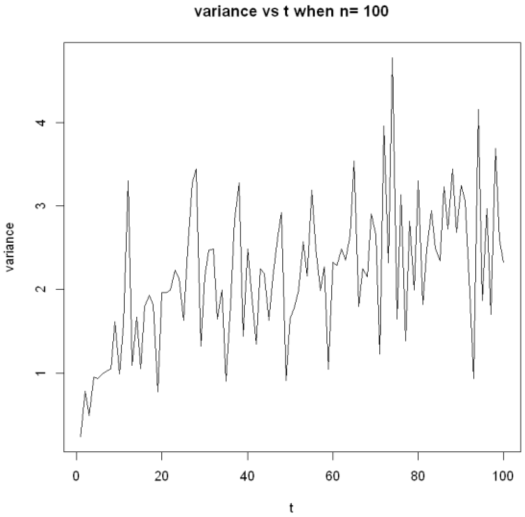
(d) Repeat part(b) with different node number. Here we set node number as 100 and 10000 separately. We still sweep step t from 1 to 100. The results are shown below. When n=100, both average distance and variance show a similar rising tendency with big fluctuation. For n=10000, the tendency of average distance and variance are similar to, but in contrast with n=100, they reach a stable state quickly after t>5, although with some small fluctuation. Combined situation n=1000 above, we can conclude that the diameter of graph play an important role in these procedure. With the increase of diameter, we tend to get a stable status of average distance and variance more easily. In other word, big diameter can be stable with less random walk steps.



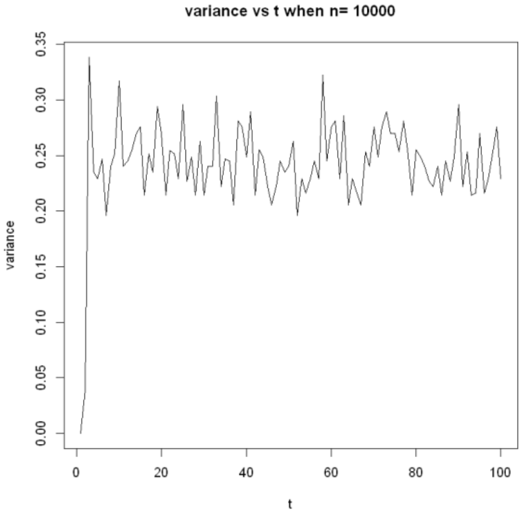
st vs t when n= 100



st vs t when n= 10000



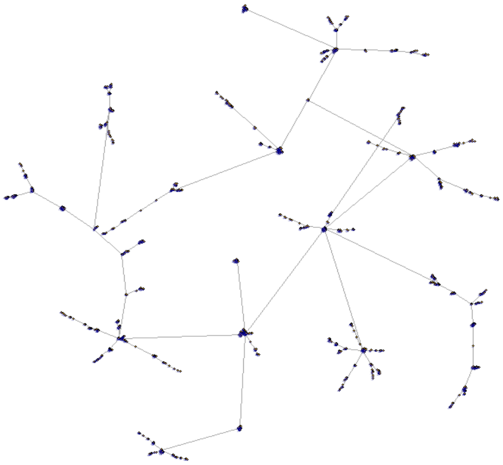
variance vs t when n= 100



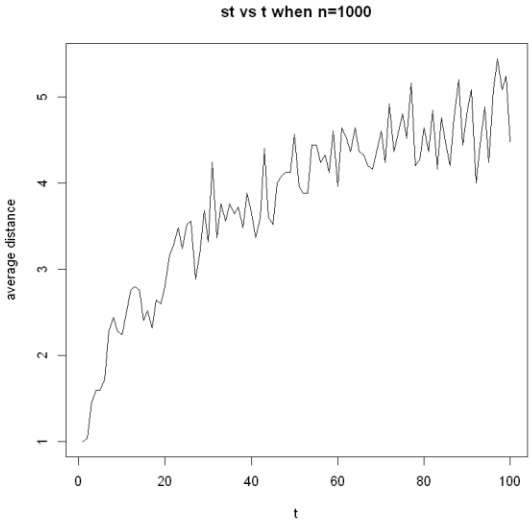
variance vs t when n= 10000

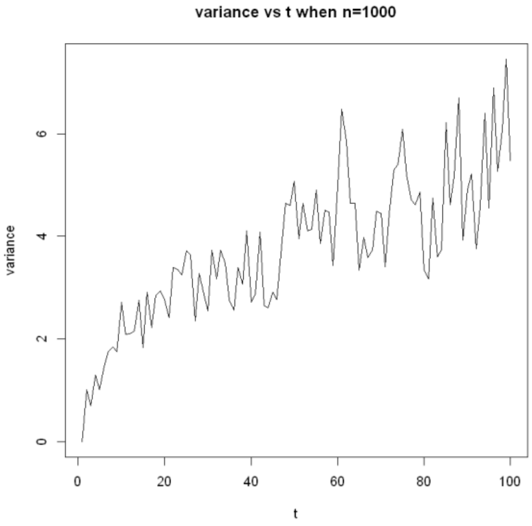
2. Random walk on networks with fat-tailed degree distribution

(a) We generated a preferential attachment network with 1000 nodes, where each new node attaches to m=1 old nodes.

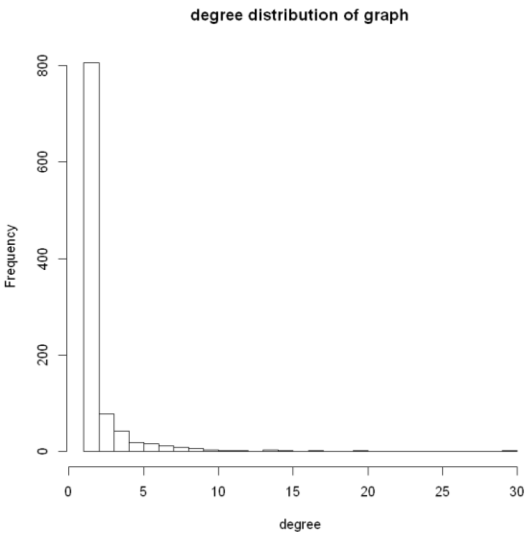


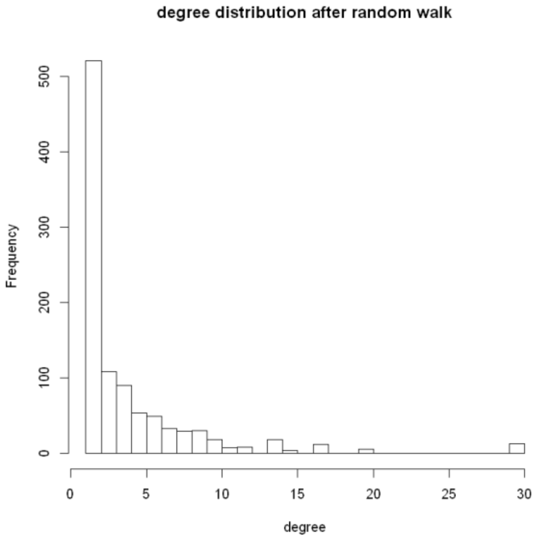
(b) Here we swept graph with step t from 1 to 100 and calculated their average distance and variance. As the pictures showing below, quite different from Erdos-Renyi graph, fat-tailed distribution doesn’t reach a stable state with the increase of step length t. Both average distance and variance rise with the increase of step length t.



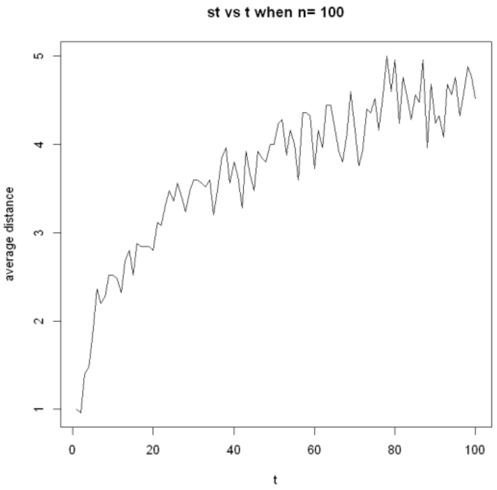


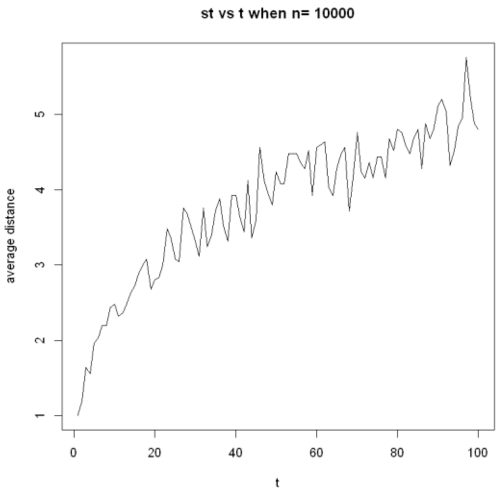
(c)The pictures below are degree distribution of graph and degree distribution after random walk. Similar as average distance and variance, these distribution are totally different compared with distribution of Erdos-Renyi graph. It seems degree from 1 to 5 occupy majority nodes and only a small part of nodes have high degree. Besides, it seems that there is no big difference between degree distribution of graph and degree distribution after random. In this point, they are similar to Erdos-Renyi graphs.

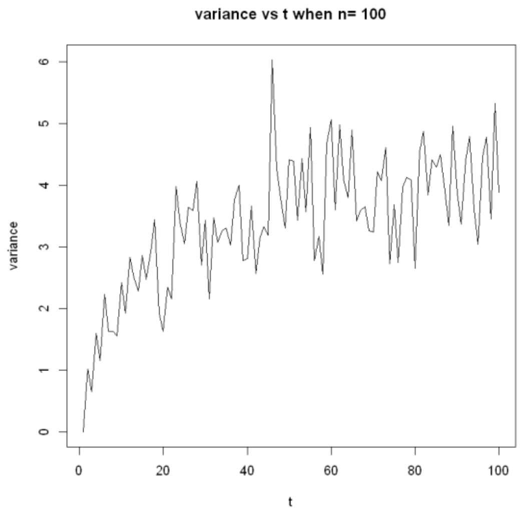




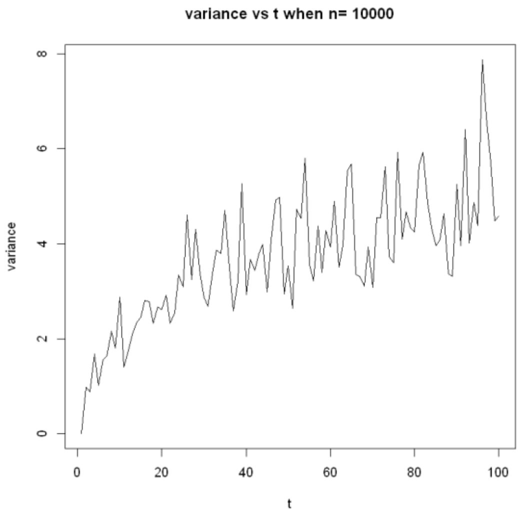
(d) Compared with Erdos-Renyi graph, diameter isn't a big influencing factor. There are no obvious difference between average distance or variance when n equals 100 or 10000. The average distance increase with the rise of step length t. Also, the variance increases with the rise of step length t, too.







variance vs t when n= 100

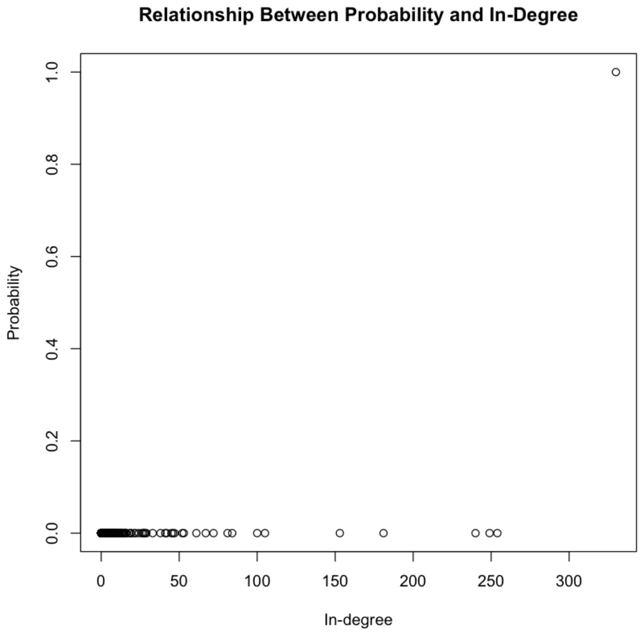


variance vs t when n= 10000

3. PageRank

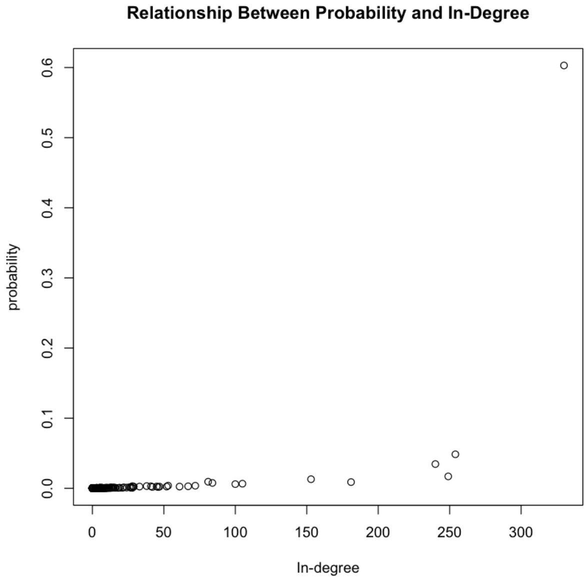
(a) We generated the directed preferential attachment graph using “barabasi.game” function provided by igraph package. Then we used the random walk to simulate PageRank. With a large enough step in each iteration, we can find the steady solution for each visit. We used the step as the diameter*10. We chose the number of iteration as 10000. The probability that the walker visits each node is derived by the number of the visits divided by the number of iterations. Because of the properties of the preferential attachment model, the newly generated node has the preference to connect to the node with the high in-degree. Thus, the first node has the most in-degree and it has the probability of being visited as 100%, which means that whatever node is chosen as the start node in the random walk, the last node visited will be the first node. The other nodes have the probability of being visited as zeros. This probability is different from the pagerank vector derived from the in-built page_rank function partly because the niter and eps set in the page_rank function.

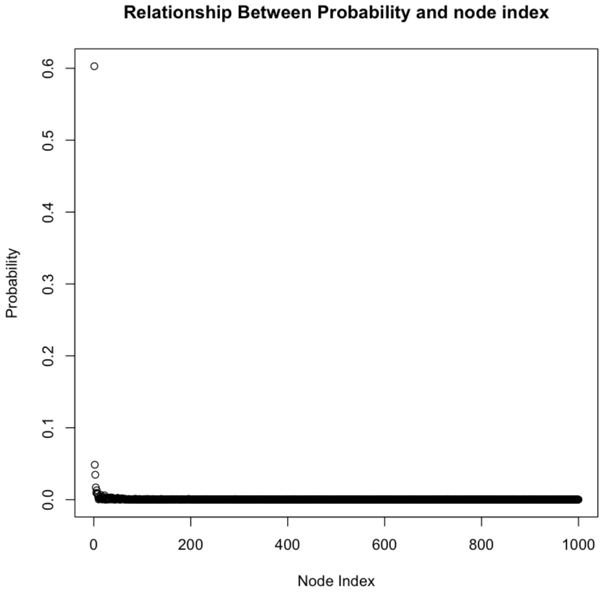
As mentioned, the first node with the largest in-degree has the largest probability, but there is no obvious relation between the probability and the degree regarding to the other nodes using random walk because their probabilities are zeros. We plotted the relationship between probability and In-Degree. Also, we calculated the correlation coefficient between the degree vector and the probability vector, which is 0.506634.



(b) We used the graph generated in 3(a). We modified the random_walk function with teleportation probability to allow teleportation during the random walk. Instead of modifying the transition matrix, we modified directly the random_walk function. We first sample to decide whether the current step teleports or not with the probability (0.15, 0.85). If not teleporting, the next step is chosen by the original transition matrix. And if teleporting, the next step is chosen by the probability of 1/N for each node. By doing in this way, it can help to decrease the running time.

The final probability vector we get is printed in the Problem2_34.ipynb as visit_prob_b. The first node has a rather bigger probability compared to other node. The first several nodes have the major probability of being visited compared to the others. It seems that the probability that walker visits each node(pagerank) and the in-degree follow a similar power laws. The following left plot shows the relationship between probability and in-degree, where we can see that the node with the higher in-degree is more likely to be visited. Although the first node has an overwhelming probability of being visited, similar in 3(a), but there are more nodes has non-zero probability of being visited. The correlation coefficient between the degree vector and the probability vector is 0.5865174, compared to 0.506634 in 3(a). Also, we plotted the probability vs the node index (the time step the node is added to the graph).

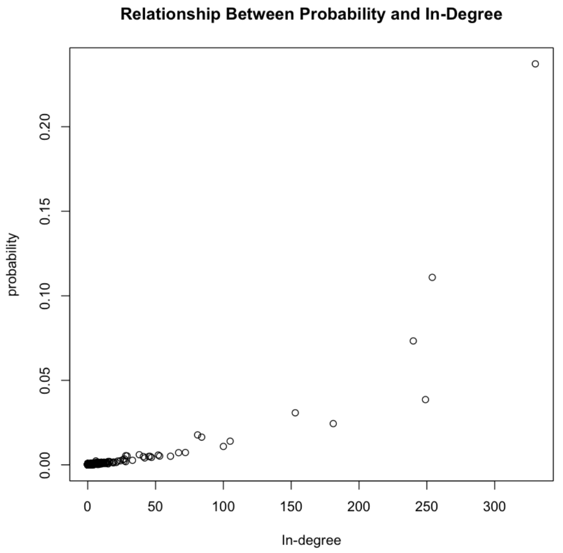


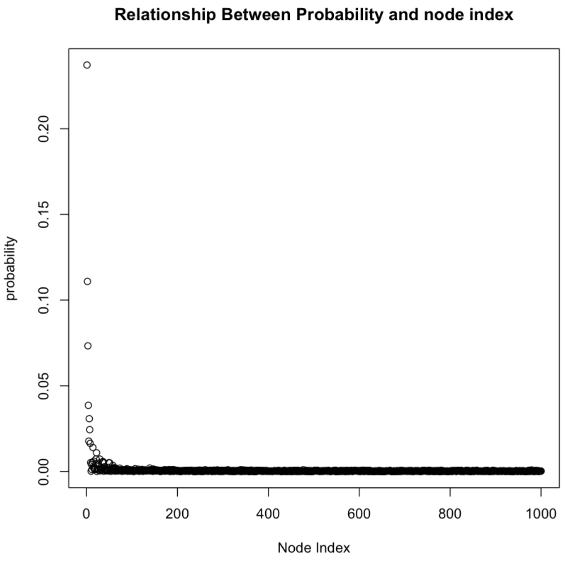


4. Personalized PageRank

(a) Use random walk on network generated in part 3 to simulate this personalized PageRank. Here the teleportation probability to each node is proportional to its PageRank. Compare the results with 3(b).

In this problem, we used the pagerank vector we get from 3(a) as the initial pagerank for the teleportation probability. Instead of modifying the random walk function, we modified the transition probability matrix with the pagerank vector. Similarly in the 3, we use the 10000 as the number of the iterations and diameter*10 as the number of steps. The final probability vector we get is printed in the Problem2_34.ipynb as visit_prob_4a.
Compared to the vector we get from problem 3(b), the personalized pagerank vector has a higher correlation coefficient with the degree of the nodes – 0.8322834. The following two plots are "visit probability vs. In-degree" and "visit probability vs. node index". The nodes with higher in-degree still tend to have a higher visit probability. Compared to problem 3(b), the similar-power-laws between the probability that walker visits each node(pagerank) and the in-degree is more significant. This is because with the teleportation of the pagerank probability, the property that the node s with higher in-degree tend to have a higher visit probability is enhanced.

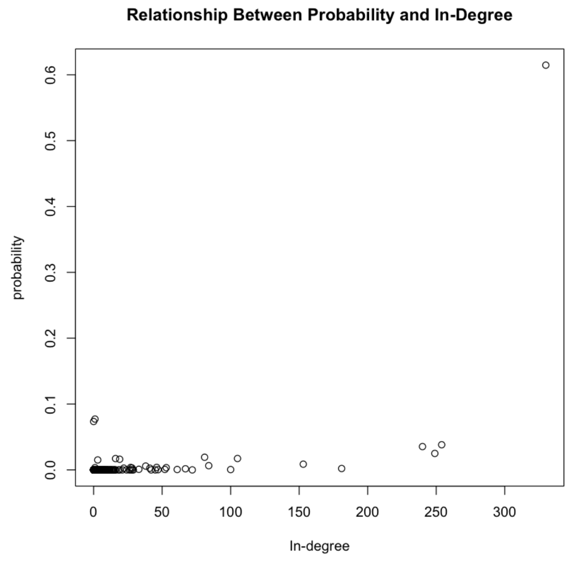


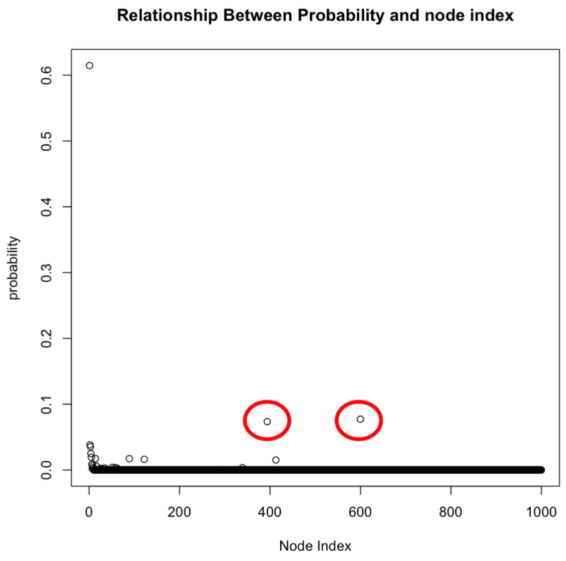


(b)We randomly choose two nodes for as the median PageRanks node because the most node has the pagerank of zero as derived in 3(a). During our test run, The node 600 and 394 are selected. Then, the teleportation probability vector becomes (0,0,0,0,...,0.5,0,0,...,0,0,5,0,0,0…..), where the values are all zeros except the 394th and the 600th. Their values are 0.5.

The final probability vector we get is printed in the Problem2_34.ipynb as visit_prob_4b. Although the general tendency is still that the nodes with higher in-degree have the higher visit probability, there are two exceptions – the pagerank of the two selected nodes have been affected by the teleportation. The pagerank of these two selected nodes has increased to around 0.075, although they do not have significant in-degree.

In the following "Probability vs. node index" plot, the visit probabilities of the selected node are circled in the red.





(c)

The original PageRank equation is:

$$\mathbf{PR} = \begin{bmatrix} (1-d)/N \\ (1-d)/N \\ \vdots \\ (1-d)/N \end{bmatrix} + d \begin{bmatrix} \ell(p_1,p_1) & \ell(p_1,p_2) & \cdots & \ell(p_1,p_N) \\ \ell(p_2,p_1) & \ddots & & \vdots \\ \vdots & & \ell(p_i,p_j) & \\ \ell(p_N,p_1) & \cdots & & \ell(p_N,p_N) \end{bmatrix} \mathbf{PR}$$

When the node is only teleported to the set of trusted node, we can make the following modification:

Suppose S denotes the set of trusted web pages which the selected nodes in our graph, $N(S)$ denotes the number of the elements in the set S, and V denotes the set of all the nodes in the graph. $\mathbf{1}_S(V)$ denotes the indicator function matrix, V_i denotes the ith node in the graph:

$$\mathbf{1}_S(V)_i = \begin{cases} 1 & , V_i \in S \\ 0 & , V_i \notin S \end{cases}$$

Then the PageRank equation can be modified as:

$$\mathbf{PR} = \begin{bmatrix} (1-d)/N(S) \\ (1-d)/N(S) \\ \vdots \\ (1-d)/N(S) \end{bmatrix} \mathbf{1}_S(V) + d \begin{bmatrix} \ell(p_1,p_1) & \ell(p_1,p_2) & \cdots & \ell(p_1,p_N) \\ \ell(p_2,p_1) & \ddots & & \vdots \\ \vdots & & \ell(p_i,p_j) & \\ \ell(p_N,p_1) & \cdots & & \ell(p_N,p_N) \end{bmatrix} \mathbf{PR}$$