EE 232E Graphs and Network Flows

HW2

Weikun Han 804774358

Duzhi Chen 004773782

2017/4/25

Introduction

In this assignment, there are some practices about using and learning the features of random walkers on graphs by using the igraph package and the netwr package developed by UCLA EE. Department. For the netwr package, at first, we could not import it into R with version 3.3.3 on Mac or Linux. Finally, we used the previous version of R – version 2.15.3 on Windows and imported the package.

Questions

1. Random walk on random networks

   We write a random walker function by using netwr package to simulate the random walk process on a graph with 100 walkers and 1000 iterations.

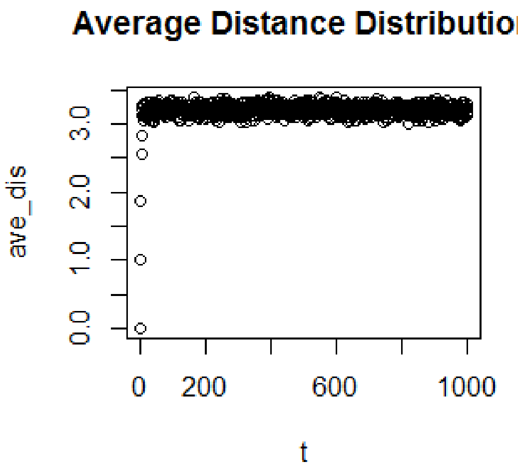
   (a) Create undirected random networks with 1000 nodes, and the probability p for drawing an edge between any pair of nodes equal to 0.01.

   We create a random graph follow the requirements above.

   (b) Let a random walker start from a randomly selected node (no damping). We use t to denote the number of steps that the walker has taken. Measure the average distance $<s(t)>$ of the walker from his starting point at step t. Also, measure the standard deviation $\sigma^2(t) = <(s(t) - <s(t)>)^2>$ of this distance. Plot $<s(t)>$ vs. t and $\sigma^2(t)$ vs. t. Here, the average $<*>$ is over all possible starting nodes and different runs of the random walk (or different walkers). You can measure the distance of two nodes by finding the shortest path between them.

   We us the random walker function to simulate the random walk on the graph with 1000 nodes.

   Figure 1 shows the average distance distribution and std of distance distribution of the random walk on the graph.



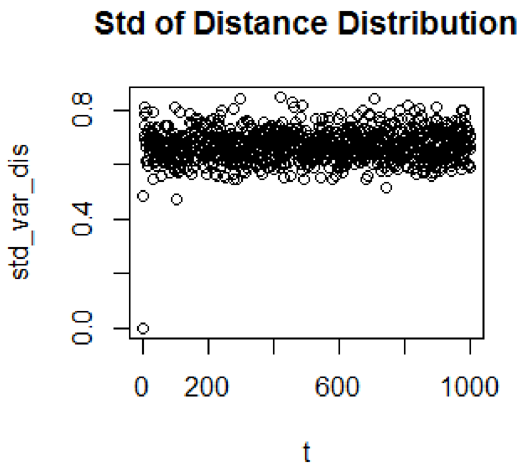


Figure 1, the average distance distribution and std of distance distribution for 1000 nodes

(c) We know that a random walker in d dimensional has average (signed) distance $<s(t)> = 0$ and $\mathrm{sqrt}<s(t)^2> = \sigma \propto \mathrm{sqrt}(t)$. Compare this with the result on a random network. Do they have similar relations? Qualitatively explain why.

They do not have similar relations.

A random walker in d dimensional can have average distance = 0 because the distance can be a negative value, which means the average distance can be 0 by the cancellation of positive and negative distances. But, the result from part (b) is not the same as the random walker in d dimensional space, because it does not have negative distance, the result from part (b) is a positive value around 3.

(d) Repeat (b) for undirected random networks with 100 and 10000 nodes. Compare the results and explain qualitatively. Does the diameter of the network play a role?

We create two random graphs with 100 and 10000 nodes respectively, and then repeat part (b) on them.

Figure 2 and 3 show the average distance distribution and std of distance distribution of the random walk on these two graphs.

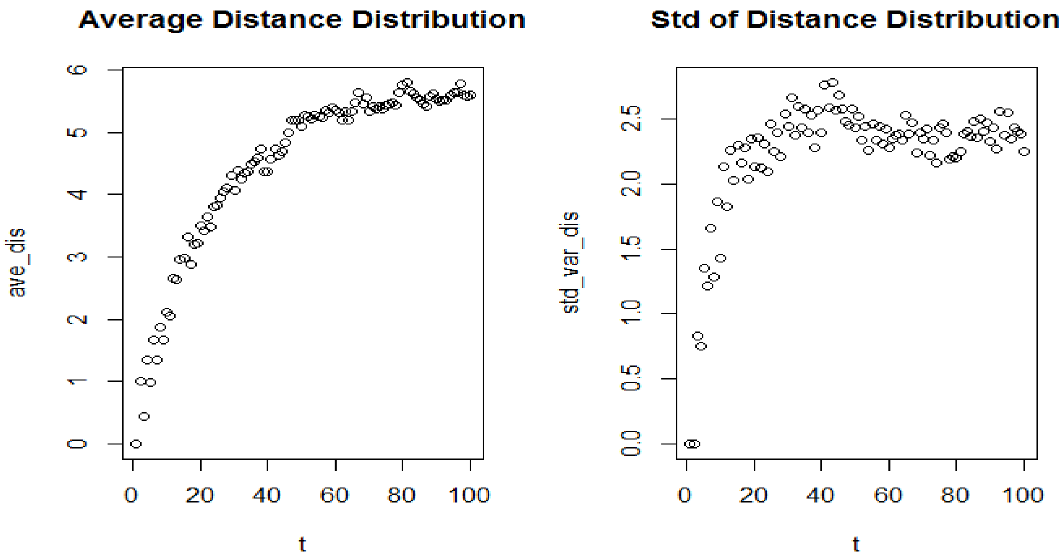


Figure 2, the average distance distribution and std of distance distribution for 100 nodes

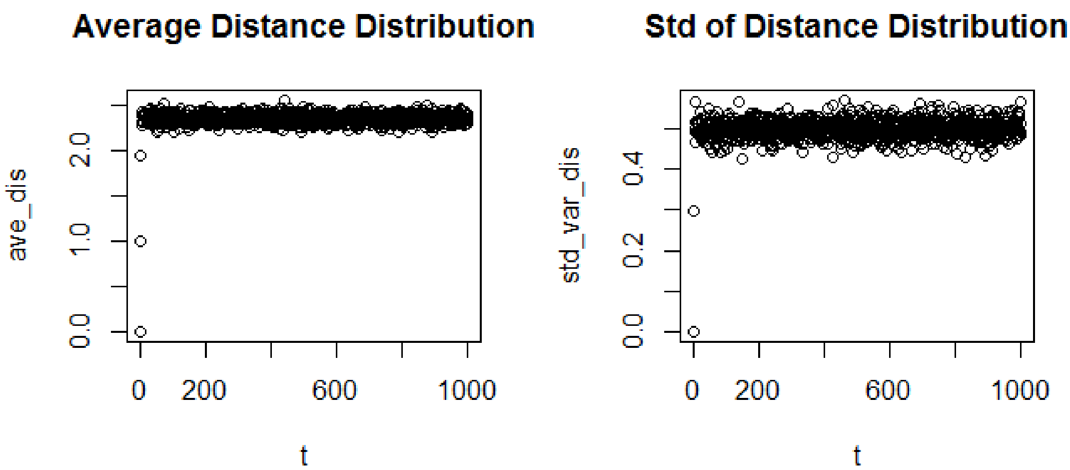


Figure 3, the average distance distribution and std of distance distribution for 10000 nodes

For 100 nodes, diameter = 10
For 1000 nodes, diameter = 5
For 10000 nodes, diameter = 3

As shown from the plots above, the smaller the diameter is, the average distance distribution and std of distance distribution converge more rapidly. We also can say that as the diameter decreases, the average distance distribution and std of distance distribution decrease.

(e) Measure the degree distribution of the nodes reached at the end of the random walk on the 1000-node random network. How does it compare with the degree distribution of graph?

Figure 4 shows the degree distribution of the graph and the degree distribution at the end of random walk on the 1000-node fat-tailed network.

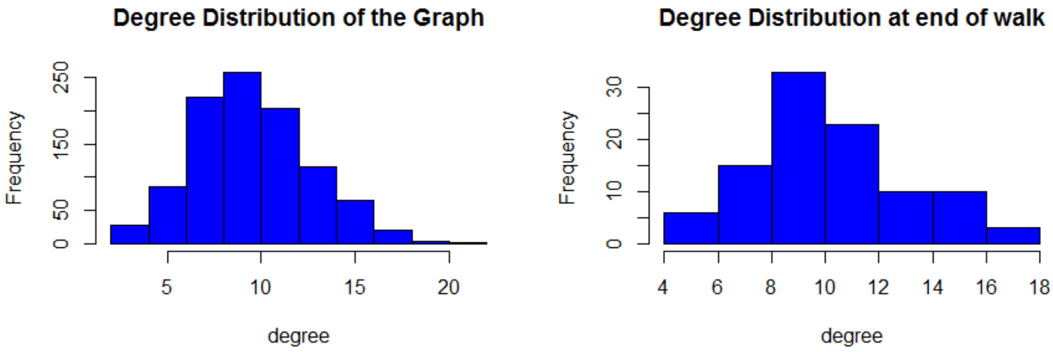


Figure 4, the degree distribution of the graph and the degree distribution at the end of walk

From the graph above we can see that these two degree distributions have similar features.

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

(a) Use barabasi.game to generate a network with 1000 nodes and degree distribution proportional to x^3.

We create a random graph follow the requirements above.

(b) Let a random walker start from a randomly selected node. Measure and plot.

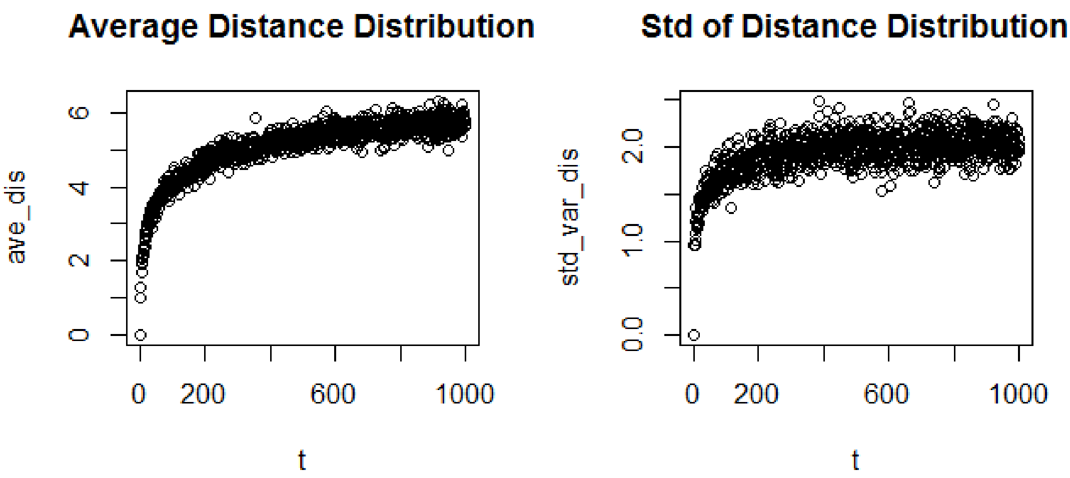


Figure 5, the average distance distribution and std of distance distribution for 1000 nodes

(c) Are these results like results of random walks in d dimensional space? Explain why.

They are not similar.

A random walker in d dimensional can have average distance = 0 because the distance can be a negative value, which means the average distance can be 0 by the cancellation of positive and negative distances. But, the result from part (b) is not the same as the random walker in d dimensional space, because it does not have negative distance.

(d) Repeat (b) for fat-tailed networks with 100 and 10000 nodes. Compare the results and explain qualitatively. Does the diameter of the network play a role?

We create two fat-tailed networks with 100 and 10000 nodes respectively, and then repeat part (b) on them.

Figure 6 and 7 show the average distance distribution and std of distance distribution of the random walk on these two graphs.

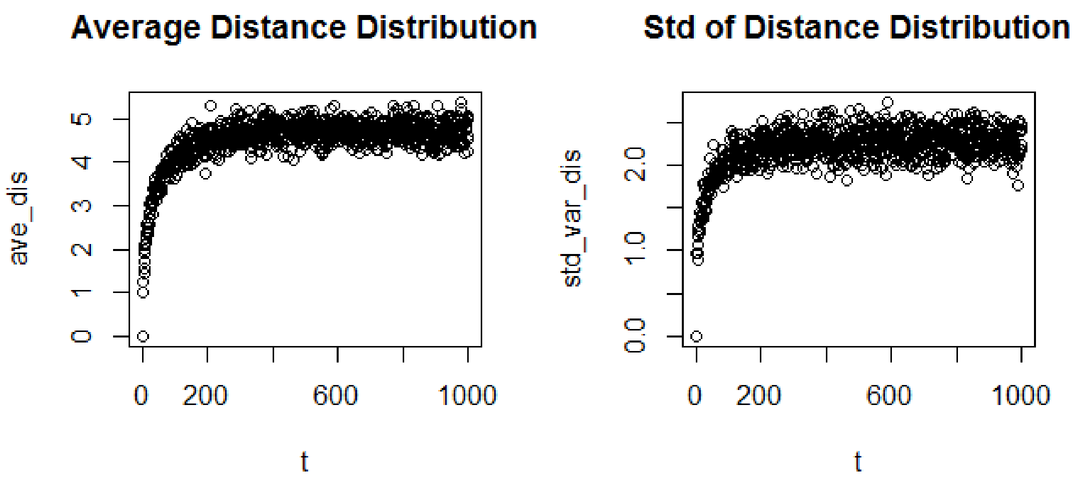


Figure 6, the average distance distribution and std of distance distribution for 100 nodes

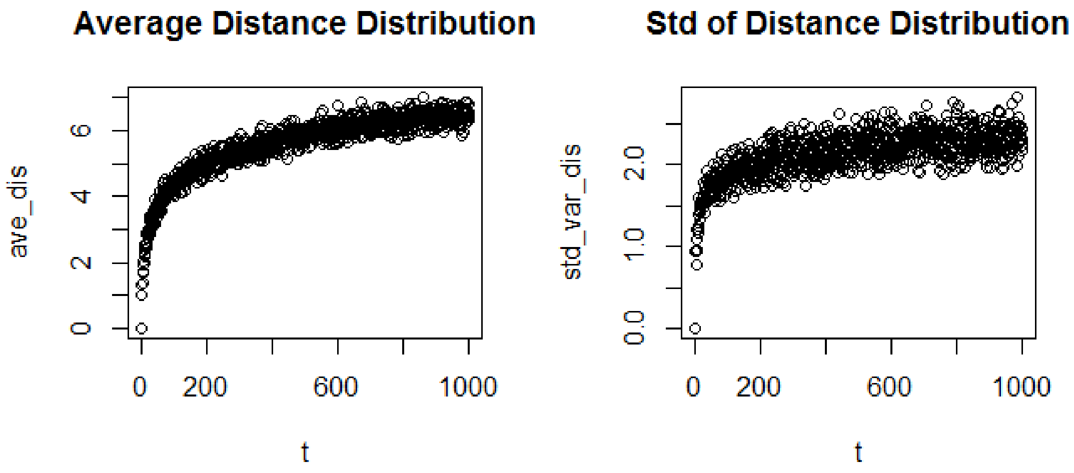


Figure 7, the average distance distribution and std of distance distribution for 10000 nodes

For 100 nodes, diameter = 14
For 1000 nodes, diameter = 16
For 10000 nodes, diameter = 20

As shown from the plots above, the smaller the diameter is, the average distance distribution and std of distance distribution converge more rapidly. We also can say that as the diameter decreases, the average distance distribution and std of distance distribution decrease.

(f) Measure the degree distribution of the nodes reached at the end of the random walk on the 1000-node fat-tailed network. How does it compare with the degree distribution of the graph?

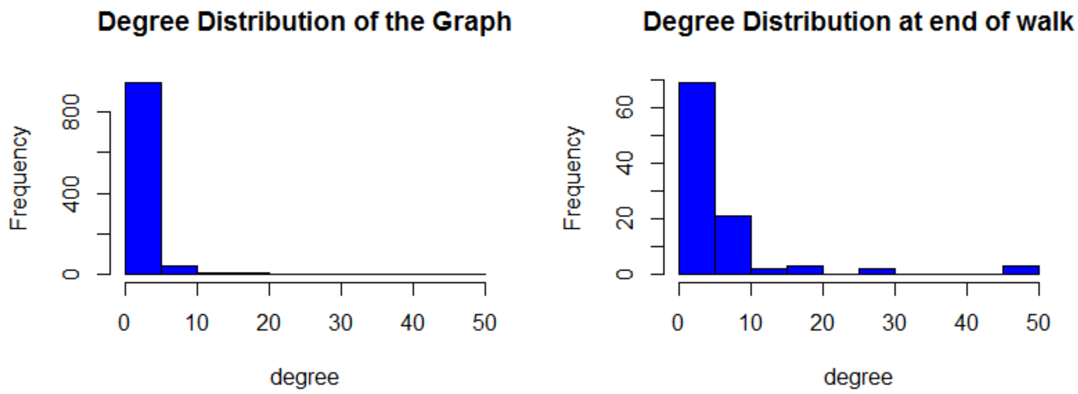


Figure 8, the degree distribution of the graph and the degree distribution at the end of walk

From the graph above we can see that these two degree distributions have similar features.

3. PageRank

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

(a) For random walks on the network created in 1(a), measure the probability that the walker visits each node. Is this probability related to the degree of the nodes?

Figure 9 shows the relationship between the probability and the degree.

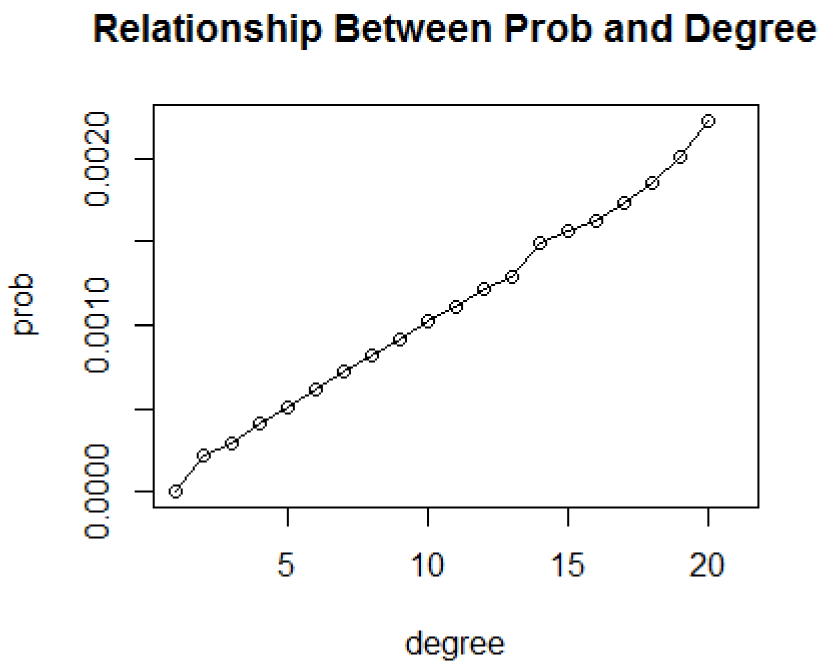


Figure 9, the relationship between probability and degree

From the plot above we can see that the probability is linearly increasing with the degree.

(b) Create a directed random network with 1000 nodes, where the probability p for drawing an edge between any pair of nodes is 0.01. Measure the probability that the walker visits each node. Is this probability related to the degree of the nodes?

We create a directed random graph. And for the degree of a directed graph, we should measure both the in and out degrees. Then Figure 10 and 11 show the relationship between the probability and the in and out degrees.

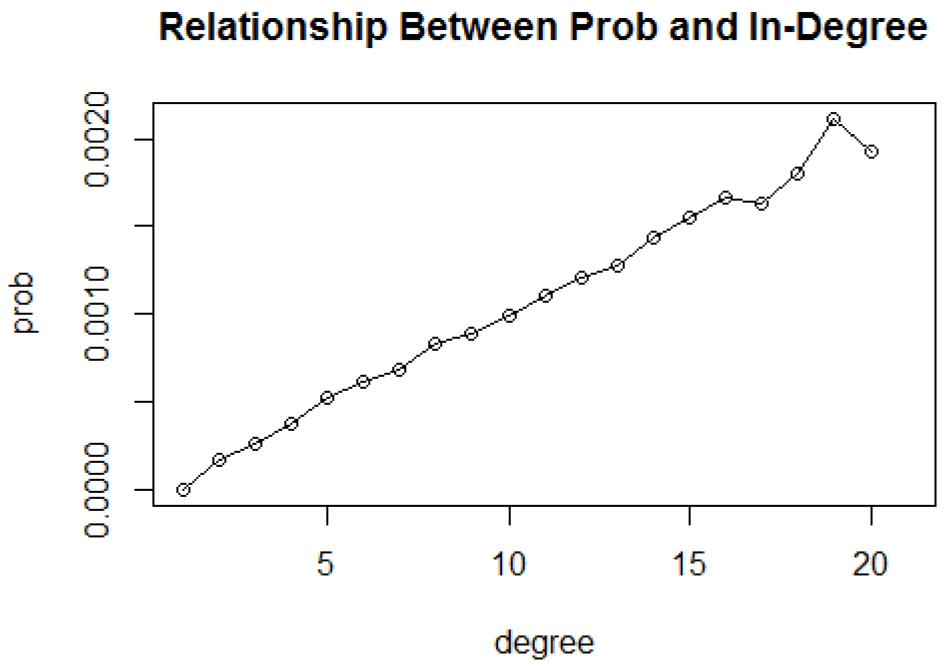


Figure 10, the relationship between probability and in-degree

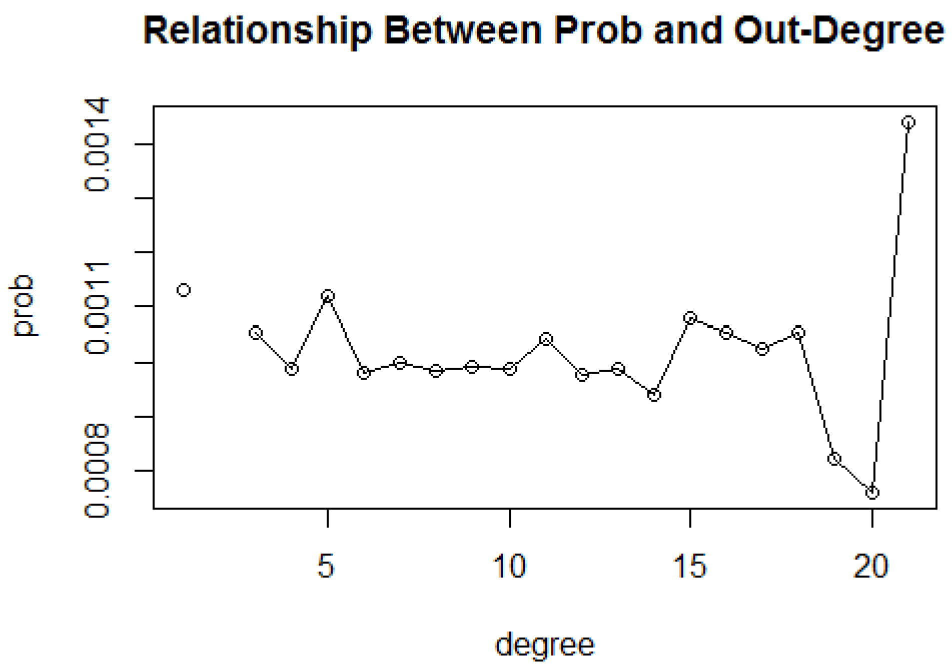


Figure 11, the relationship between probability and out-degree

From the plots above we can see that the probability is linearly increasing with the in-degree. But the probability is not related to the out-degree.

(c) In all previous questions, we used a damping parameter equal to d = 1, which means no teleportation (because teleportation probability is equal to 1 - d = 0). Now, we use a damping parameter d = 0.85. For random walks on the network created in 1(a), measure the probability that the walker visits each node. Is this probability related to the degree of the node?

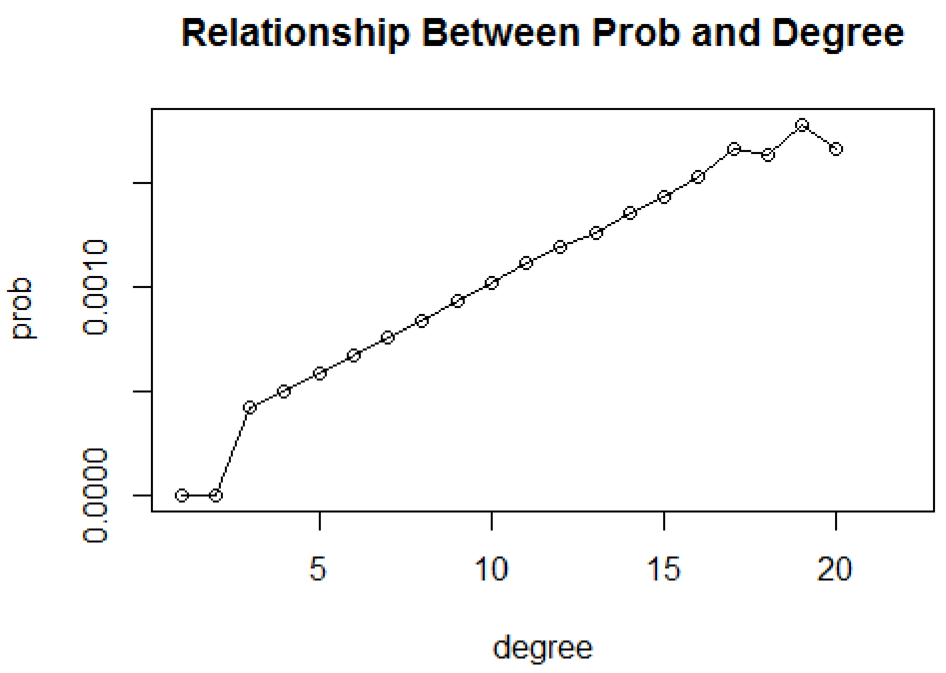


Figure 12, the relationship between probability and degree with d = 0.85

From the Figure 12 we can see that the probability is linearly increasing with the degree.

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.

(a) Create a directed random network with 1000 nodes, where the probability p for drawing an edge between any pair of nodes is 0.01. Use the random walk with damping parameter 0.85 to simulate the PageRank of the nodes

We simulate the PageRank of nodes

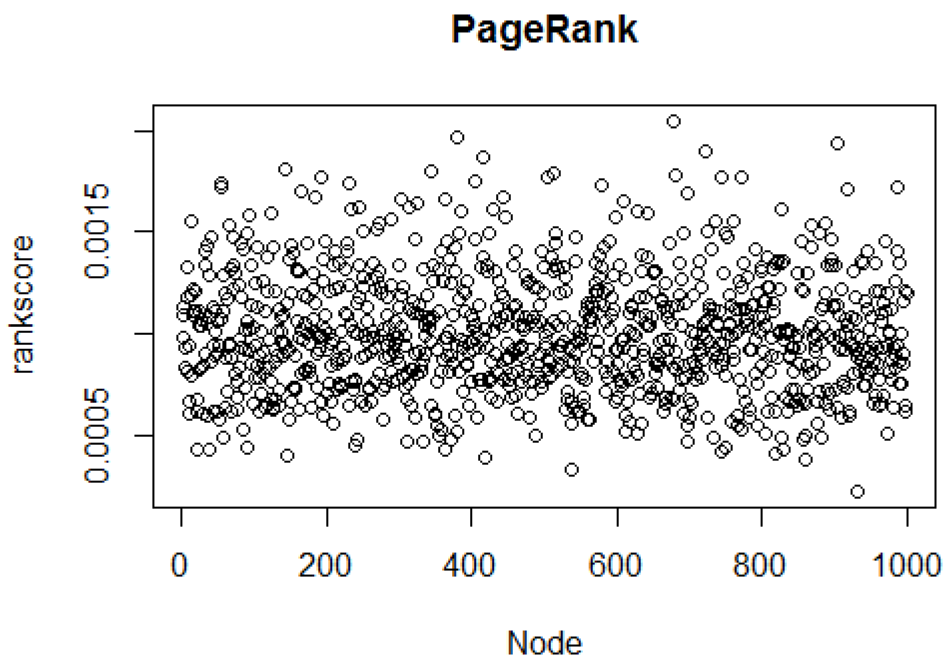


Figure 13, the PageRank of all nodes

(b) Suppose you have your own notion of importance. Your interest to 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 the random walk to simulate this personalized PageRank. Here the teleportation probability to each node is proportional to its PageRank (As opposed to the regular PageRank, where teleportation probability to all nodes are the same and equal to 1/N). The damping parameter is equal to d = 0.85. Compare the results with (a).

As shown in the Figure 14, the PageRank and personal PageRank are similar. The points on the plots have similar distributions.

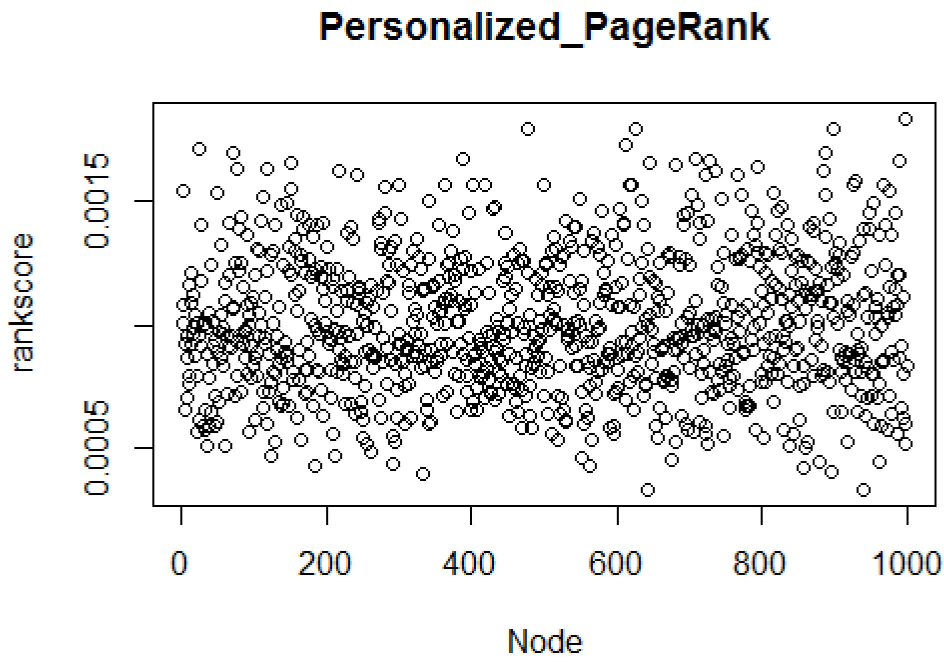


Figure 14, the personalized PageRank of all nodes

(c) Part (b) is what happens in the real world. However, this is against the original assumption of normal PageRank, where we assume that people's interest in all nodes are the same. Can you consider the effect of this self-enforcement and adjust the PageRank equation?

The equation of normal PageRank is

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

As the assumption in part (b), if people have their own notion of importance and have different interests in each node, we can just change the teleportation probability to each node to be proportional to its PageRank.

$$PR(p_i) = \frac{PR(p_i) - d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$