1. Objectives

In this homework, we are supposed to learn features of random walkers on a specific graph. The project is based on what has been done in homework 1 using igraph package. Moreover, a specialized R package named “netrw” developed by EE Dept. of UCLA will be utilized to simulate the random walkers on a given graph. Properties such as distribution of nodes reached at the end of random walk will be studied in details.

2. Problems

2.1. Problem 1 – Random walk on random networks

For random walk on the graph, we write a function RandomWalker by using netwr. It simulates the random walk process of 100 walkers, and the default number of steps(iterations) is 1000.

The function code:

```
RandomWalker =function(g,steps=1000,DF)
{
   ave_dis = std_var_dis = numeric(0)
   for(i in 1:steps){
      print(i)
      #100 is the default walker.num
      r=netrw(g, T=i,damping=DF, output.walk.path=TRUE)
```

```
        shortest_path=numeric(0)
        #j is the same with walker.number
        for(j in 1:100){
            temp = shortest.paths(g,from=r$walk.path[1,j],to=r$walk.path[i,j])
            shortest_path=c(shortest_path,temp)
        }
        ave = mean(shortest_path)
        std_var = sd(shortest_path)
        ave_dis=c(ave_dis, ave)
        std_var_dis=c(std_var_dis, std_var)
    }
}
```

(a)

Using random.graph.game to create a random graph, which is the same as HW1.

(b)

Using the function RandomWalker to simulate random walk on the graph with 1000 nodes.

Figure 1.1 shows the average distance and standard deviation of graph with 1000 nodes:

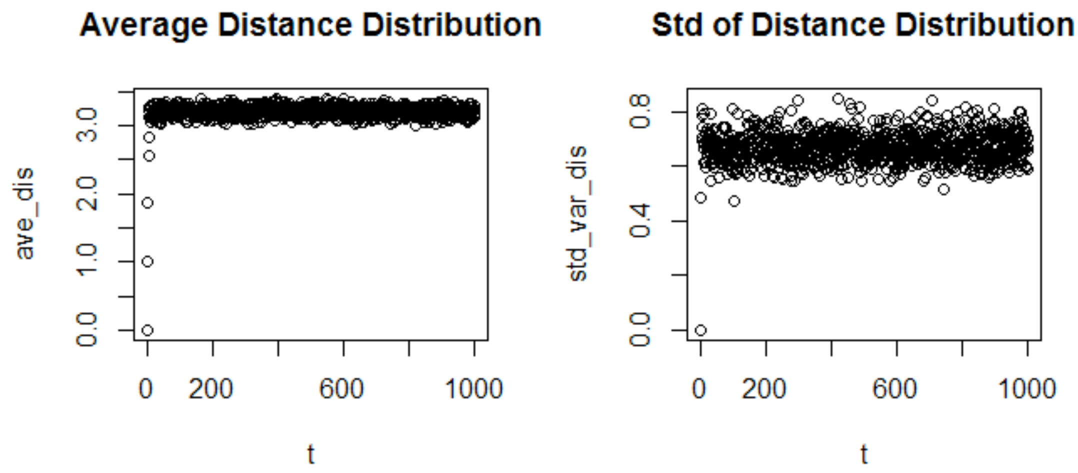

Figure 1.1 Ave_dis and Std_var_dis of random walk in graph with 1000 nodes

(c)

The result in (b) is not the same as the random work in the d-dimensional space. This is because, in the d-dimensional space, the distance could be negative, which means it could have cancellation between positive and negative distance that causes the average distance to be 0. However the random graph we generated has no negative distance, so the result is a positive value around 3. Thus, the result is different from a random walker in d dimensional.

(d)

Repeat the process for random networks with 100 and 10000 nodes.

For random networks with 100 nodes, we set the start node to be fixed.

Changed code:

r=netrw(g,walker.num=vcount(g),start.node=1,T=i,damping=DF,output.walk.path=TR

UE)

Figure 1.2 and 1.3 show the average distance and standard deviation of graph with 100 and 10000 nodes, respectively:

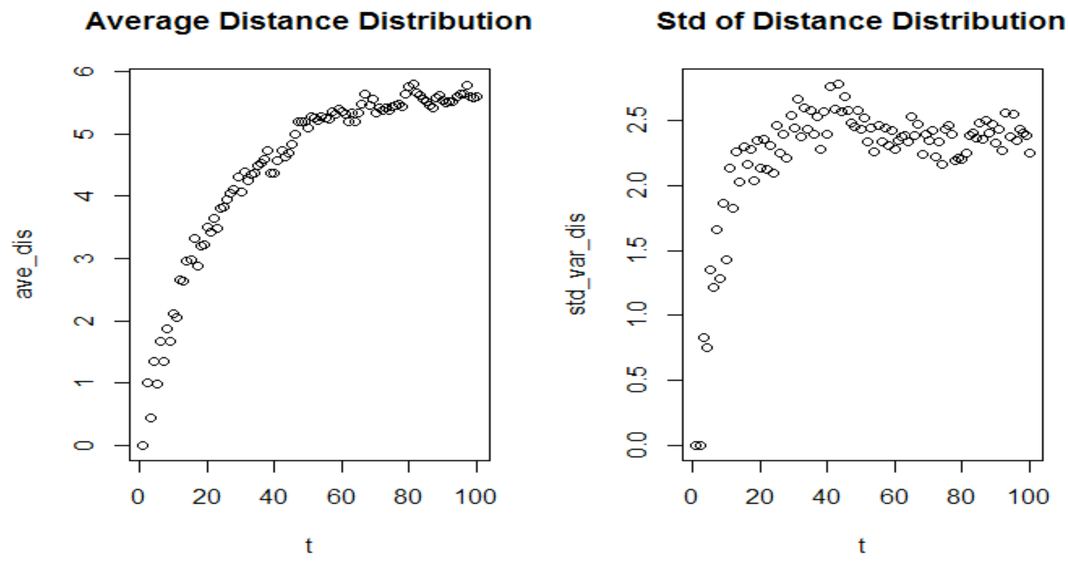

Figure 1.2 Ave_dis and Std_var_dis of random walk in graph with 100 nodes

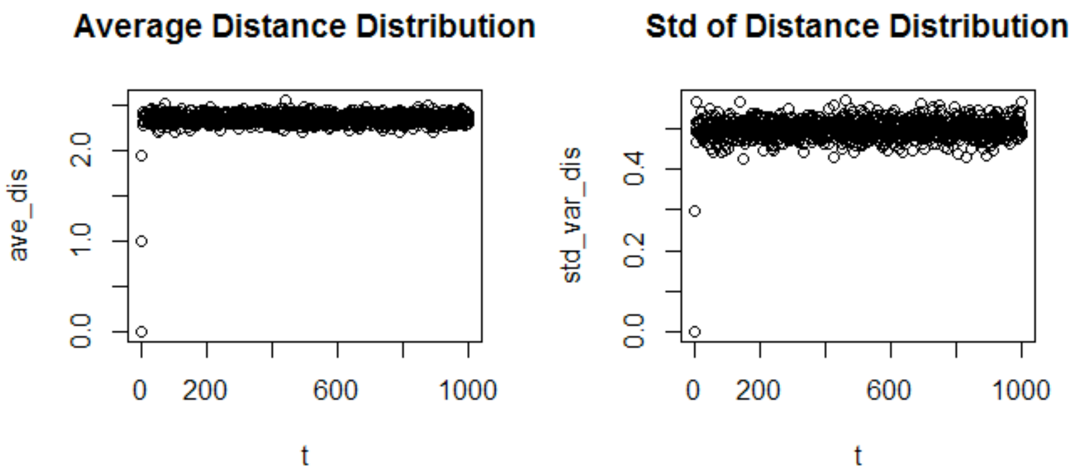

Figure 1.3 Ave_dis and Std_var_dis of random walk in graph with 10000 nodes

Diameter: The diameters of graphs with 100, 1000 and 10000 nodes are 10,5,3, respectively. We could see from the figures, for the graph with 100 nodes the distribution converge slowly. The smaller the diameter is, the Ave_dis and Std_var_dis converge more quickly. In addition, the distribution of each step for smaller diameter seems to be closer, which means the variance of the Std_var_dis becomes smaller.

(e)

We set 500 step as the end of random walk.

Figure 1.4 shows the comparison between the degree distribution of the graph and the degree distribution of the nodes reached at the end of the random walk on the graph.

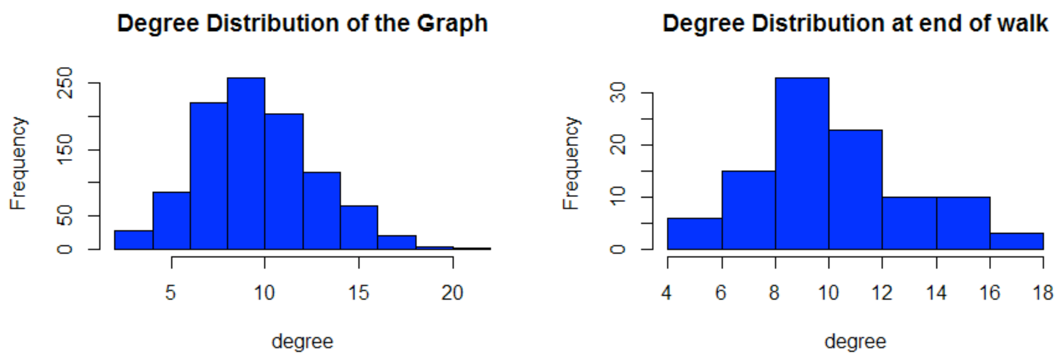

Figure 1.4 Comparison between two degree distribution

After running several times, we could see from the figure that after certain large number of steps of random walk, the two degree distributions are similar.

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

2.2 Problem 2 – RW on network with fat-tailed degree distribution

(a)

Using barabasi.game to create a random graph, whch is the same as HW1.

(b)

Using the function RandomWalker to simulate random walk on the graph with 1000 nodes.

Figure 2.1 shows the average distance and standard deviation of graph with 1000 nodes:

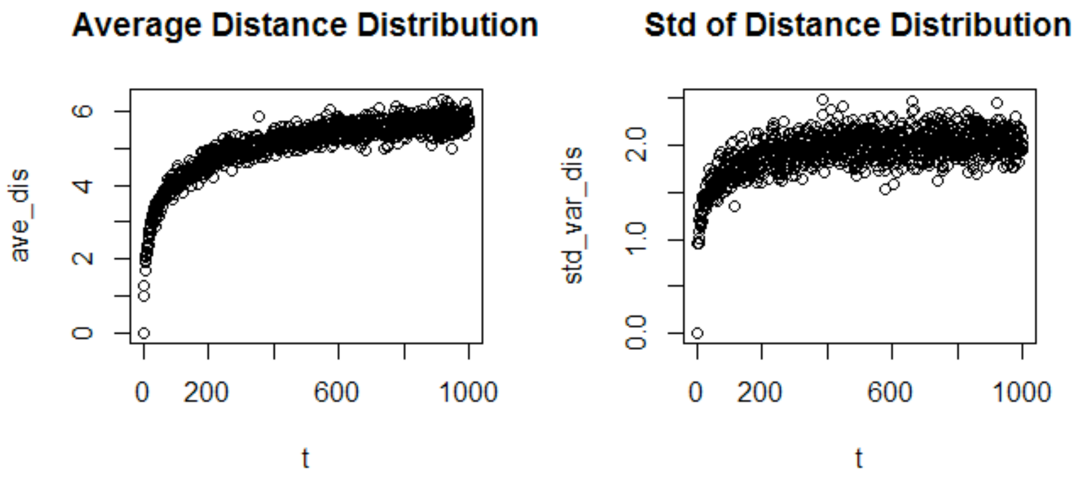

Figure 2.1 Ave_dis and Std_var_dis of random walk in graph with 1000 nodes

(c)

The same reason as 1(c). In the d-dimensional space, the distance could be negative. But for the graph we generated in (a), there is no negative distance. Thus, the results are not similar with the results of random walks in d-dimensional space.

(d)

Repeat the process for fat-tailed networks with 100 and 10000 nodes.

Figure 2.2 and 2.3 show the average distance and standard deviation of graph with 100 and 10000 nodes, respectively:

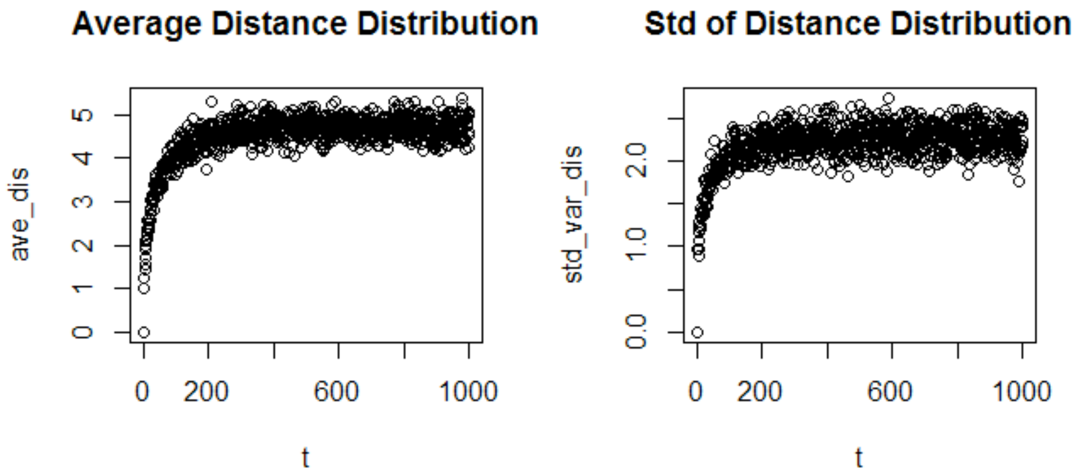

Figure 2.2 Ave_dis and Std_var_dis of random walk in graph with 100 nodes

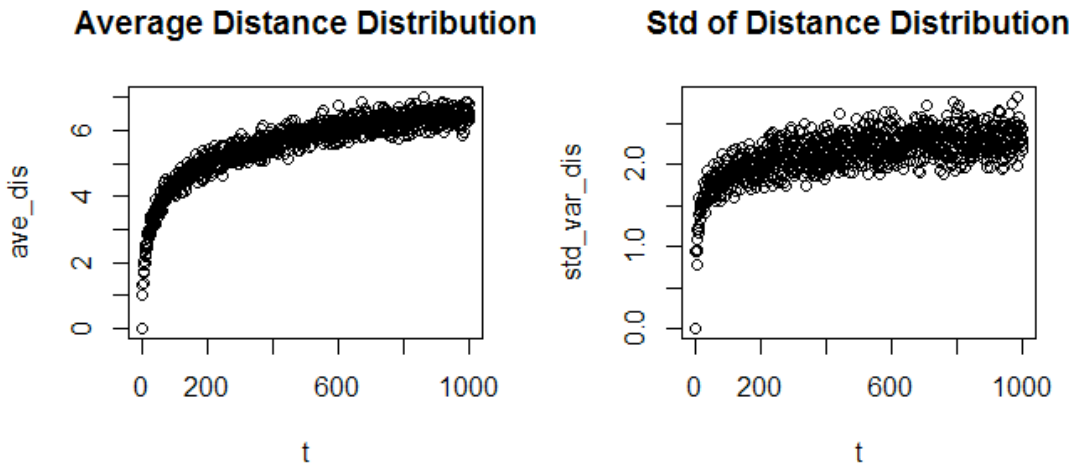

Figure 2.3 Ave_dis and Std_var_dis of random walk in graph with 10000 nodes

Diameter: The diameters of graphs with 100, 1000 and 10000 nodes are 14,16,20, respectively. We could see from the figures, for the graph with 10000 nodes the distribution converge slowly. The smaller the diameter is, the Ave_dis and Std_var_dis converge more quickly.

(e)

We set 500 step as the end of random walk.

Figure 2.4 shows the comparison between the degree distribution of the graph and the degree distribution of the nodes reached at the end of the random walk on the graph.

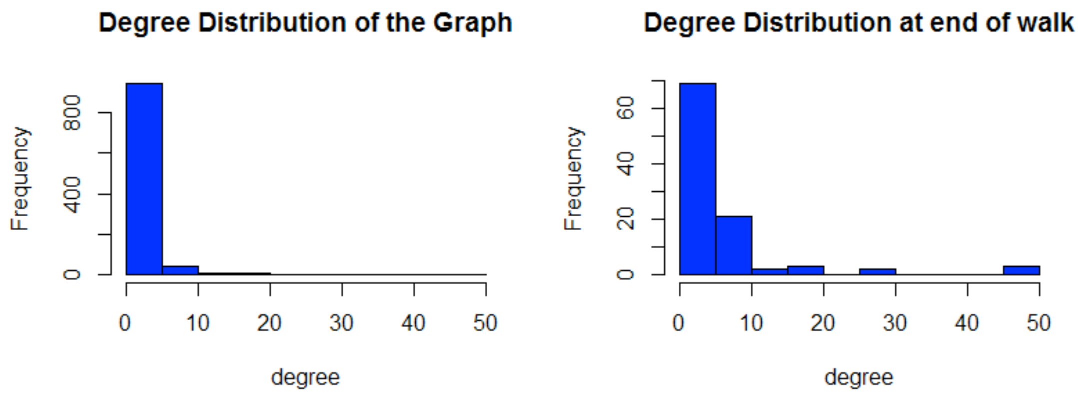

Figure 2.4 Comparison between two degree distribution

After running several times, we could see from the figure that after certain large number of steps of random walk, the two degree distributions are similar.

2.3 Problem 3 – Page Rank

(a)

Figure 3.1 shows the relationship between the probability and the degree of the nodes.

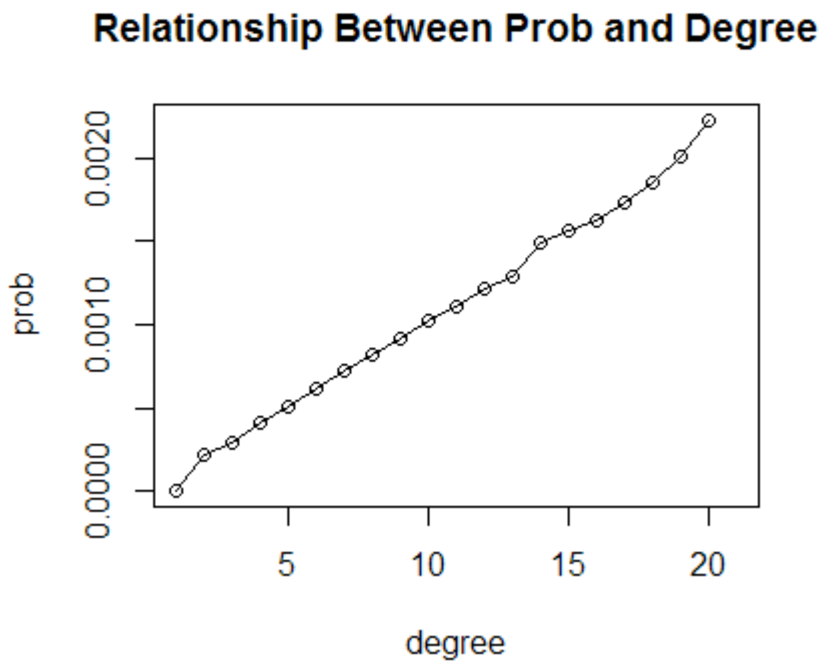

Figure 3.1 Relationship between probability and degree

From the figure, we could find out that the probability and degree are linearly related, which could also shown from the correlation of these two parameters. The correlation value is 0.945678, calculated by the function cor.

(b)

For a directed graph, we should measure both the in-degree and the out-degree.

Figure 3.2 and 3.3 show the relationship between the probability and the in and out degree of the nodes, respectively.

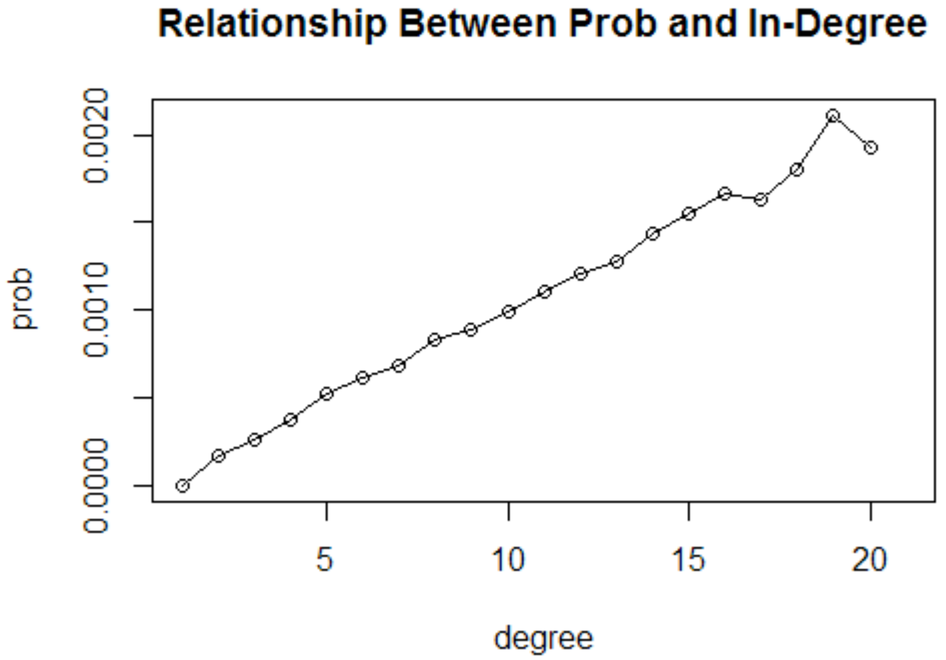

Figure 3.2 Relationship between probability and in-degree

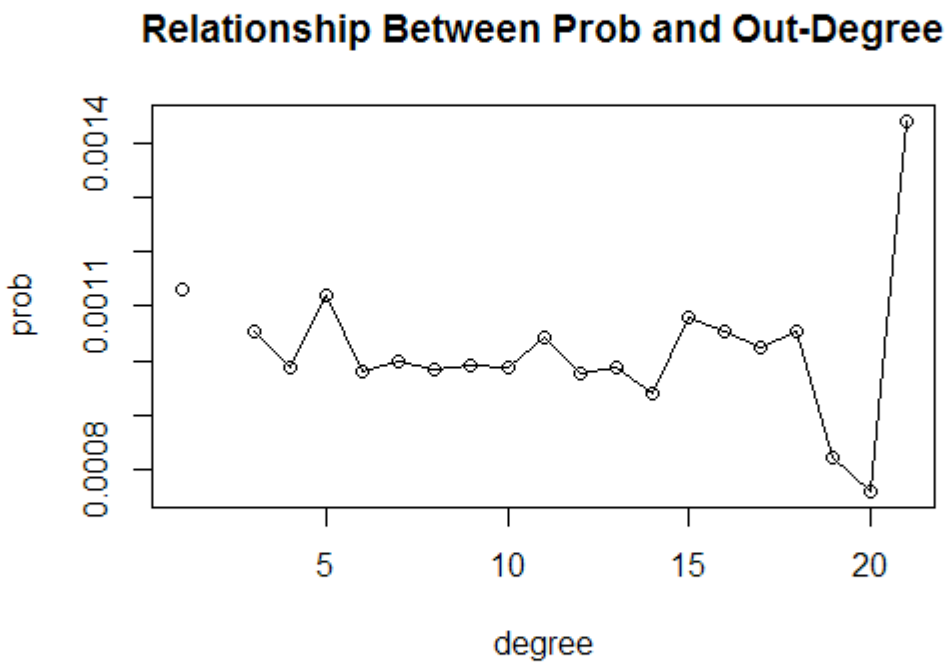

Figure 3.3 Relationship between probability and Out-degree

We could see from the figures that the in-degree is linearly related to the probability while the out-degree seems not related to the probability. The correlation value between the in-degree and probability is 0.8658934, and the correlation value between out-degree and probability is -0.01402701. From the correlation values, we could also get the same result.

(c)

Change the damping parameter d=0.85.

Figure 3.4 shows the relationship between probability and degree with damping factor 0.85

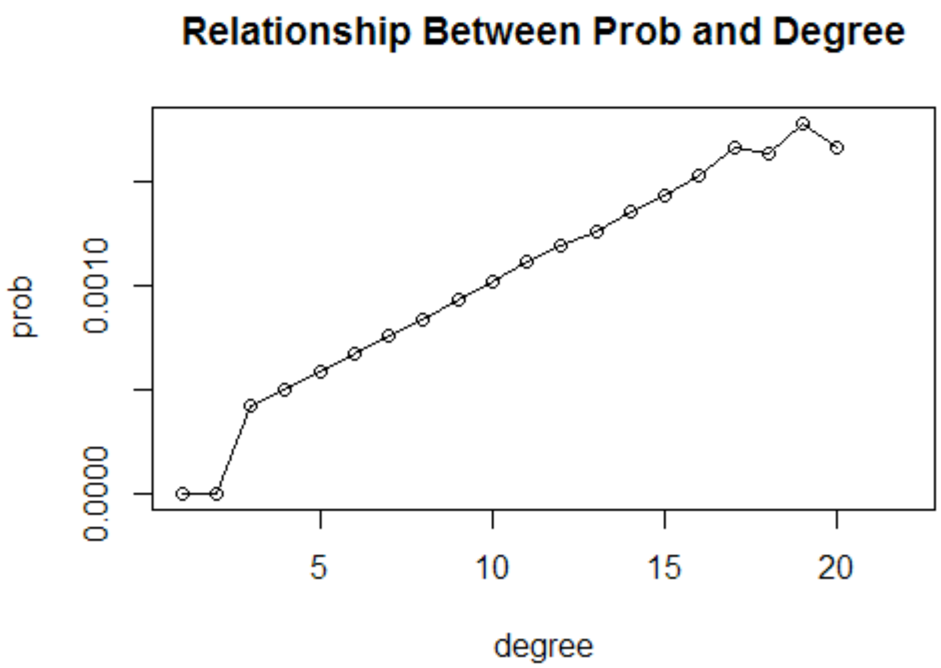

Figure 3.3 Relationship between probability and Out-degree with DP=0.85

Compare to the previous one with DP=1 the correlation value = 0.9269036 is a little bit small, but the probability and the degree are still linearly related, which demonstrated by the figure 3.3.

2.4. Problem 4 – Personalized Page Rank

(a)

Using the page.rank function in igraph to simulate the PageRank of the nodes.

Figure 4.1 shows the pagerank of all the nodes(pages)

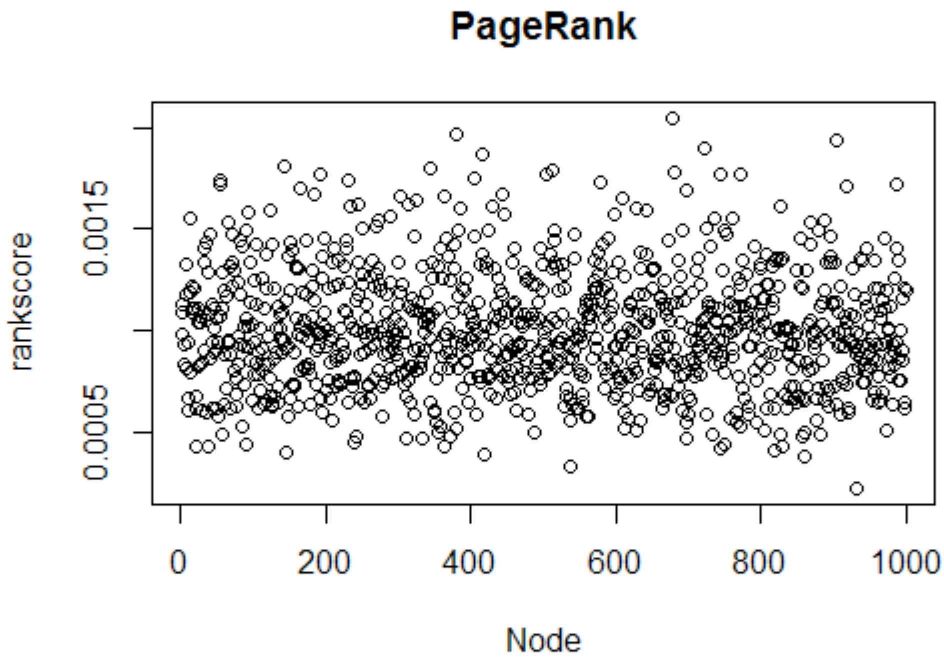

Figure 4.1 PageRank

(b)

Using the function personalized.pagerank

Figure 4.2 shows the personalized pagerank of all the nodes(pages)

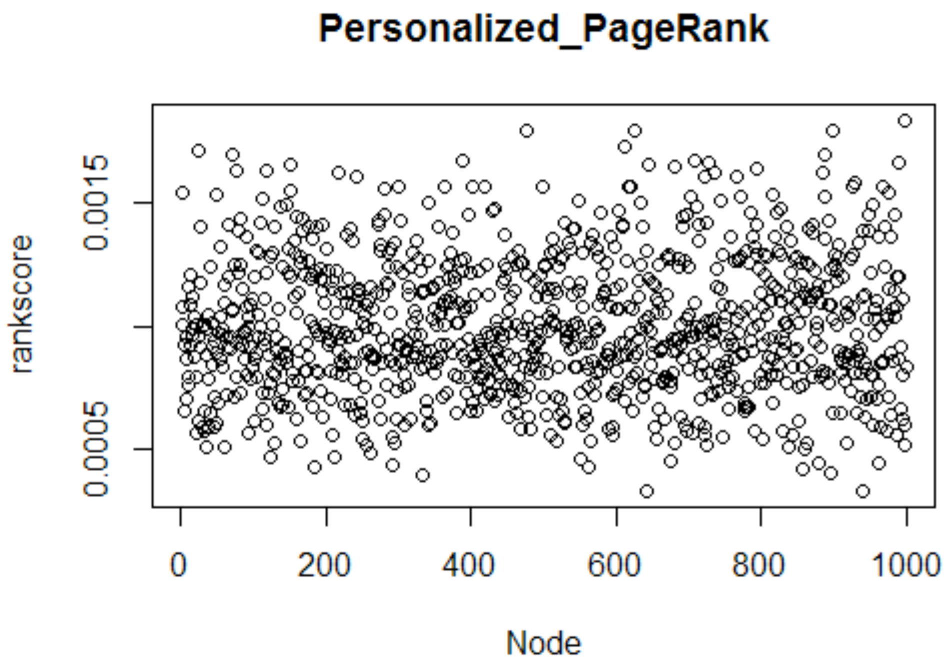

Figure 4.2 Personalized PageRank

(c)

PageRank is the density of random walker on it at stationarity. So the equation $PR(A) = (1-d)/N + d\sum_{T_{in}} \frac{PR(T_{in})}{C(T_{in})}$ is the regular page rank, so change the teleportation probability from 1/N to a number, which is proportional to PageRank.

3. Difficulty encountered

At the first of this project, we could not successfully import the netrw package in R with version 3.0, so we tried hard to find previous version. However, in Mac OX and Ubuntu, the access of previous version is denied. So we finally used virtual machine of windows to install R 2.15.3 and get the library imported.

What’s more, it took a long time to understand the exact meaning of each input and output of the netrw function. However, it deserved the effort. The package was useful and our other work goes quickly.