Homework 2 Report for EE232E

Lei Ding

Yi Ding

Sonu Mishra

Question 1: Random walk on random networks

(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.

Solution:

The required graph was generated in R using random.graph.game function in igraph package. The degree distribution is shown in figure 1. The diameter of the graph is 6.

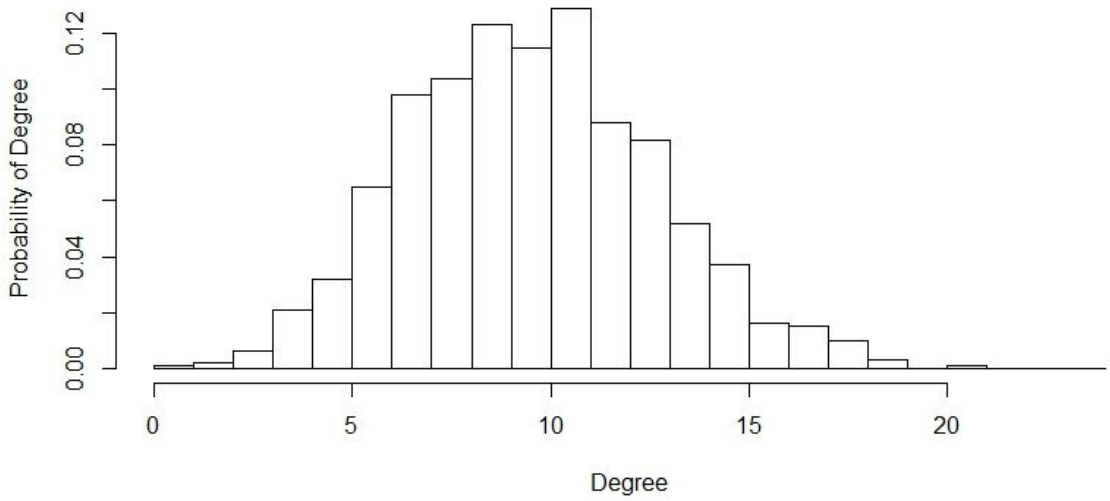

Figure 1.1: Degree distribution of random graph of 1000 nodes

(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 $\sigma^2(t) = \langle (s(t) - \langle s(t) \rangle)^2 \rangle$ *of this distance. Plot <s(t)> v.s. t and* $\sigma^2(t)$ *v.s. 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.*

Solution:

We started with randoms walker on all possible starting nodes and measured at each time step their distances from their respective starting positions. The average, variance and standard deviations of these distances were measured at each time step, are shown in Figure 1.2, 1.3 and 1.4, respectively. It can be seen that average, variance and standard deviation converge to 3.2, 0.46, and 0.68, respectively.

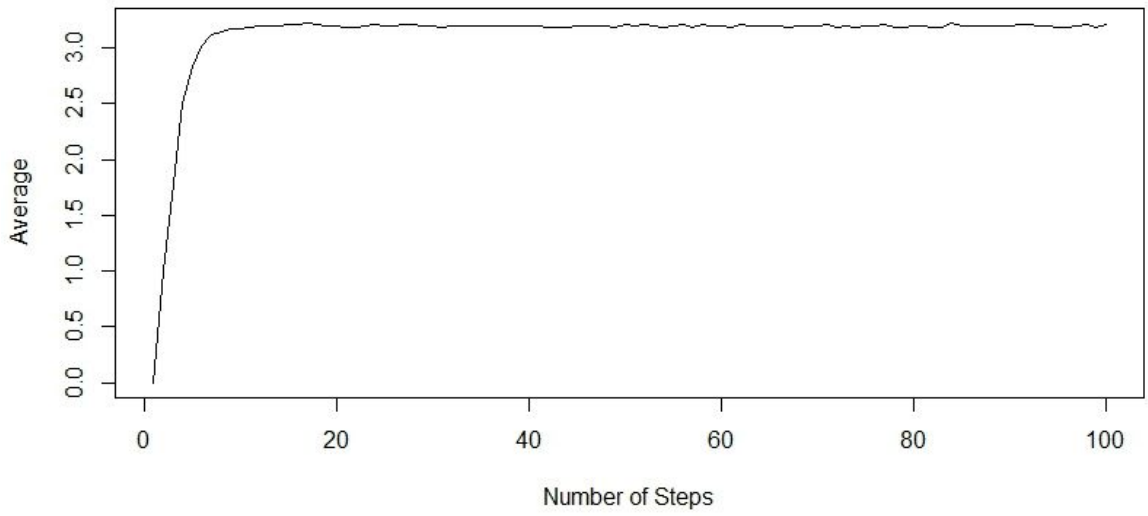

Figure 1.2: Average distance travelled by random walkers vs time in the graph with 1000 nodes

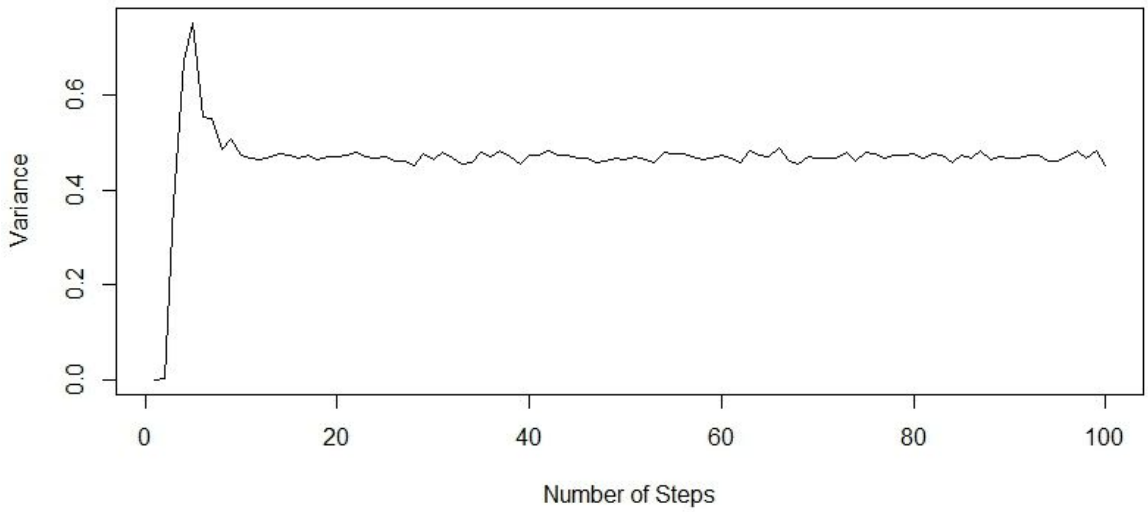

Figure 1.3: Variance of distance travelled by random walkers vs timein the graph with 1000 nodes

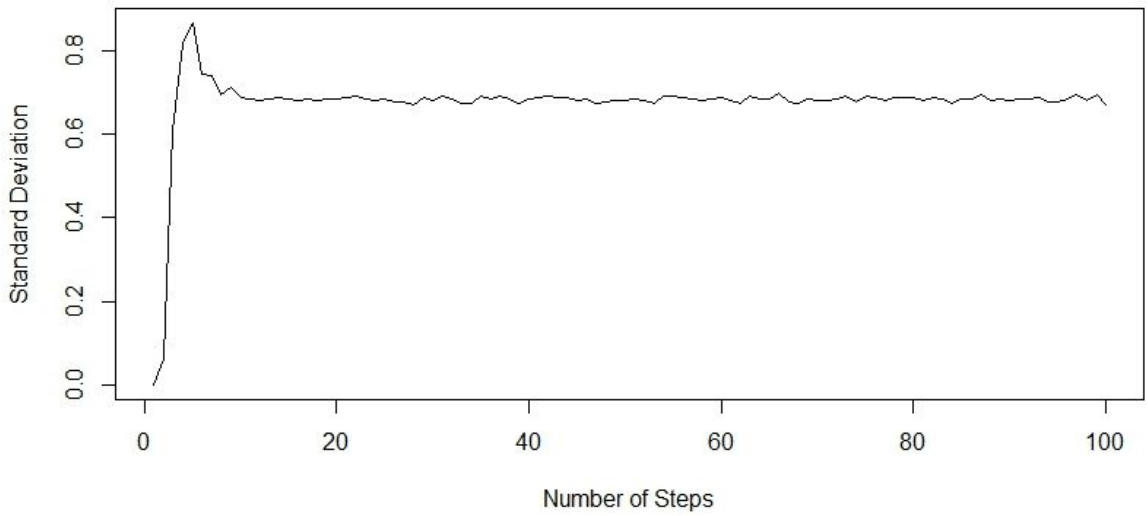

Figure 1.4: Standard deviation of distance travelled by random walkers vs timein the graph with 1000 nodes

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

Solution:

From Figure 1.2, we can see that the average result is not similar to the result of that in d-dimensional space. The reason is that for d-dimensional, the distance could be negative, and also might be symmetric to positive, therefore the average distance in d-dimensional space is likely to be near zero. But for the random graph created here, the distance between every two nodes must be positive, thus the final average distance will converge to a positive number after certain steps of random walk iterations.

Also, from Figures 1.3 and 1.4, we can also see that the result of variance and standard deviation are also not very similar to that of random walks in d-dimensional space. There is a peak in the beginning of the plot.

(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?

Solution:

We generated two random graphs with 100 and 10000 nodes using random.graph.game function of igraph library. The 100 node graph had a diameter of 10 and the 10000 node had a diameter of 3. This is expected, because in the bigger graph, each node is expected to have more number of neighbours. Therefor, the maximum of the shortest distance between any two nodes decreases.

We repeated (b) for both of these graphs by calculating and plotting the average, variance and standard deviation of the distances with respect to time. The plots are given in Figures 1.5 to 1.10. We found some similarities and some differences.

In both cases, the quantities seem to converge to certain value as the time progresses. The average distance travelled in the 100 node graph is around 1, whereas the average distance travelled in the 10000 node graph is around 2.4. This is due to the fact, the 100 node network is more often disconnected and has multiple small connected components. Therefore the random walkers get trapped in the small connected components and just keep roaming around locally. On the other hand, if the network is connected, the random walkers are more likely to travel farther. Therefore 10000 node network and the 1000 node network in 1b, have higher average distance travelled.

However if we compare the average distance of 10000 node network with the 1000 node network in 1b, we can see that the distance travelled in 1000 node network is more. This is because, unlike 100 node network, 1000 node network is often connected. So the above explanation will not hold here. This has to be explaned using diameter. We know that the diamater of 1000 node is more than the diameter of the 10000 node. Therefore the average distance travelled is more.

Another key point to notice is that the the quantities converge very quickly in the case of larger networks. The smaller networks on the other hand take more time to reach the steady state.

Random network with 100 nodes

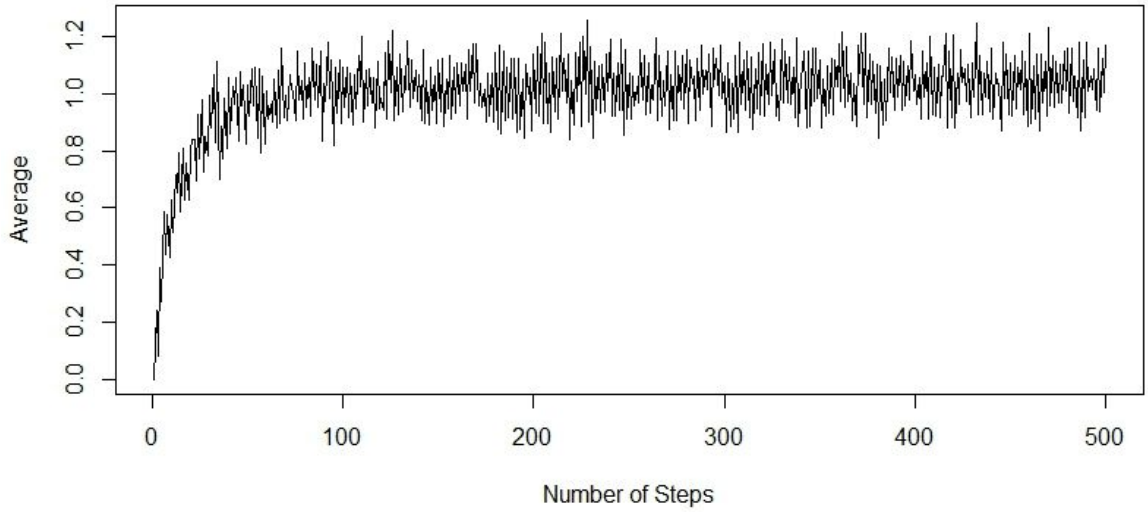

Figure 1.5: Average distance travelled by random walkers vs time in the graph with 100 node

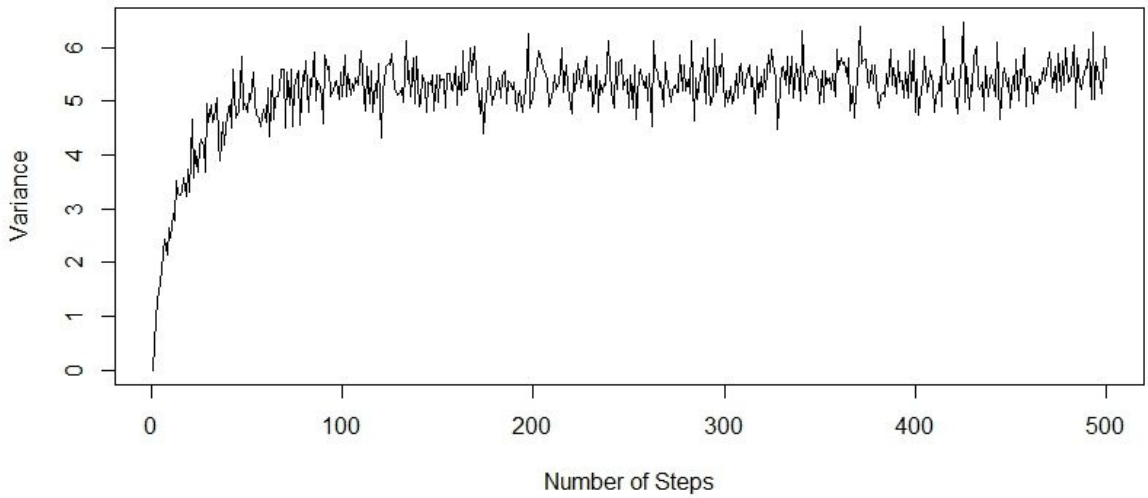

Figure 1.6: Variance of distance travelled by random walkers vs time in the graph with 100 nodes

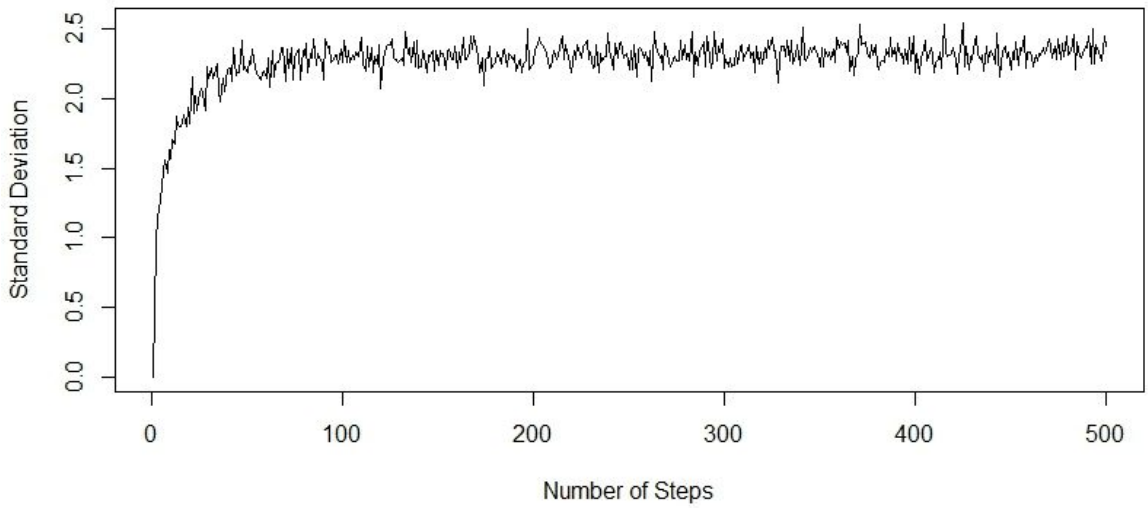

Figure 1.7: Average distance travelled by random walkers vs time in the graph with 100 nodes

Random network with 10,000 nodes

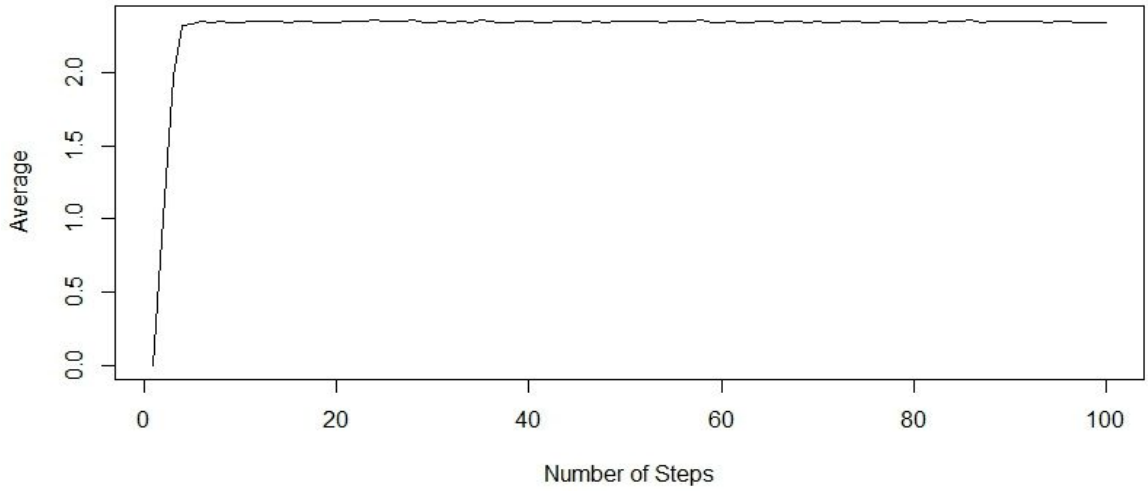

Figure 1.8: Average distance travelled by random walkers vs time in the graph with 100 nodes

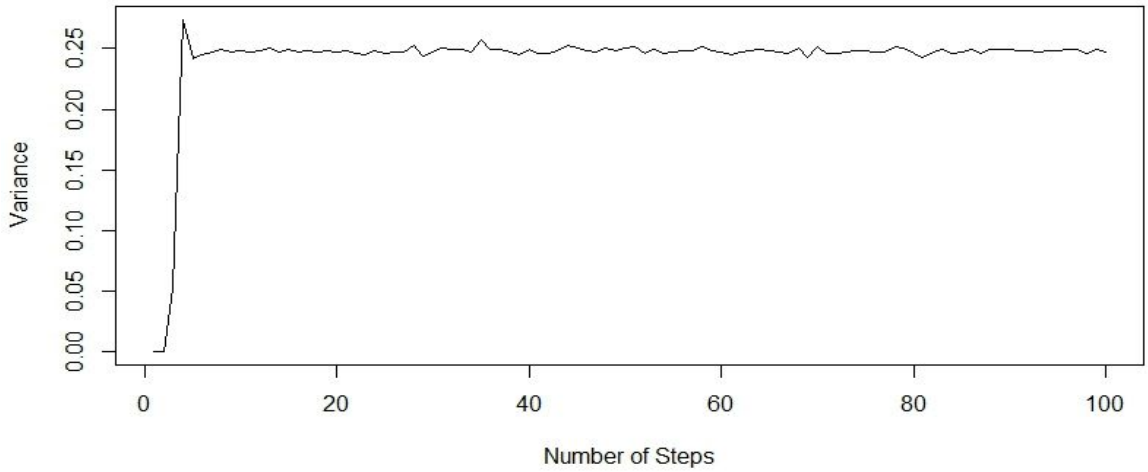

Figure 1.9: Variance of distance travelled by random walkers vs time in the graph with 100 nodes

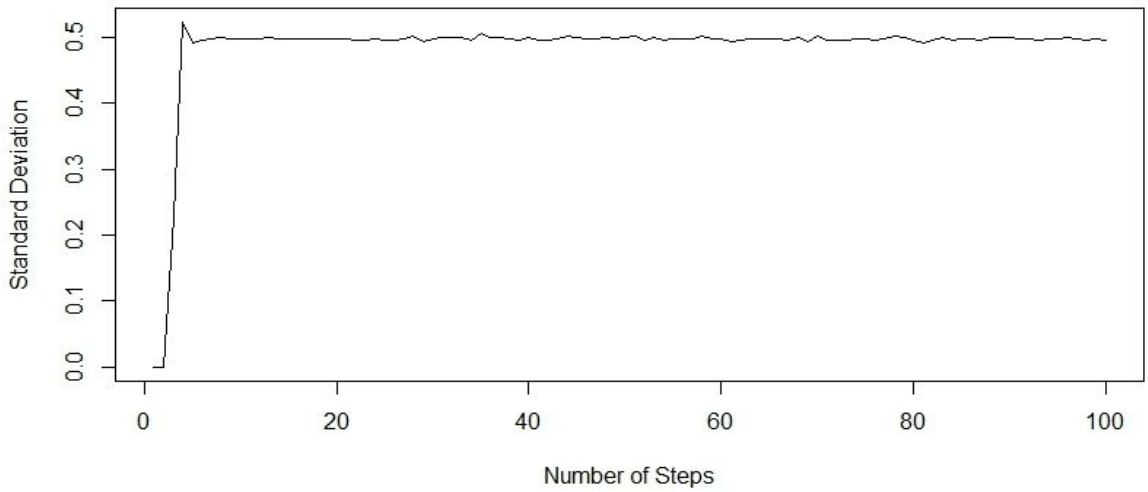

Figure 1.10: Standard deviation of distance travelled by random walkers vs time in the graph with 100 nodes

(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?

Solution:

The degree distribution of the nodes reached at the end of the random walk on the 1000-node network is figure 1.12, together with the original degree distribution figure 1.11. It can be seen that the degree distribution seems to have shifted right by small amount. That is, more proportion of nodes now have higher degree as compared to the original degree distribution. This bolsters the fact that the random walks tend to visit the nodes with higher degrees which is intuitive. A node that has more number of neighbours can be visited from any of those neighbours. On the other hand, the node with only one neighbour will be visited only if the neighbour is visited.

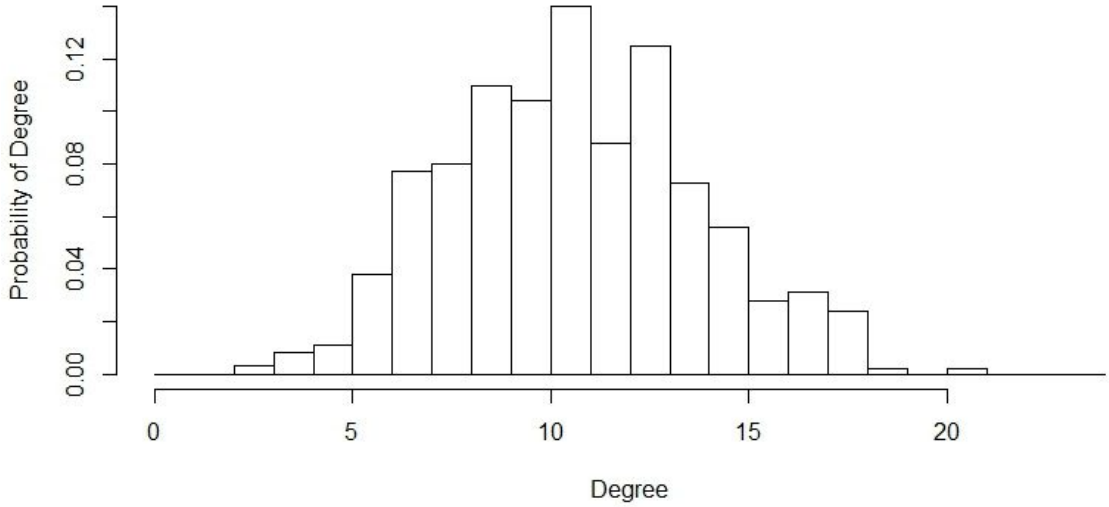

Figure 1.11: Degree distribution of nodes in 1000 node random graph reached at the end of random walk

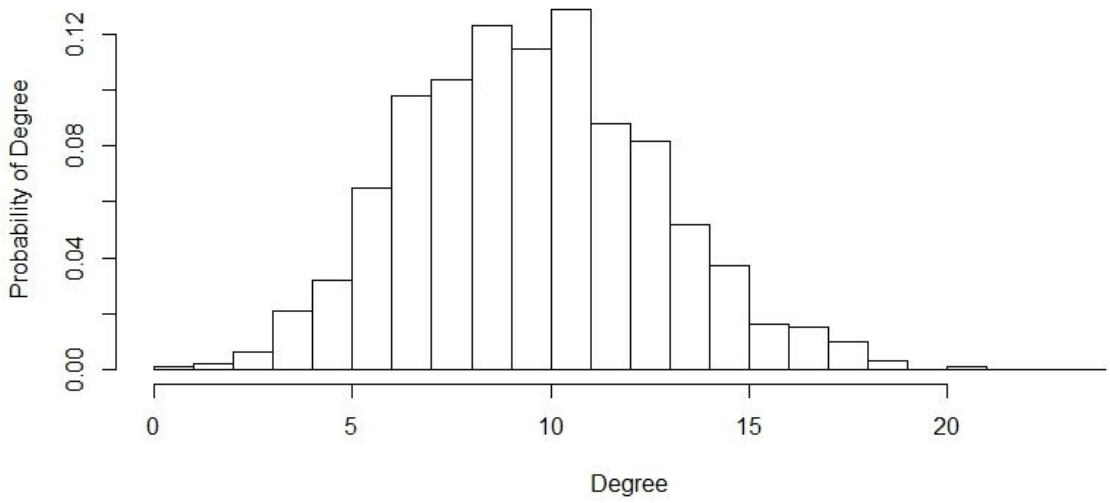

Figure 1.12: Degree distribution of the random graph with 1000 nodes

Question 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}.

Solution:

Using barabasi.game in igraph package, the fat-tailed network with 1000 nodes is created. The degree distribution is shown in the following figure. The diameter of the graph is 16.

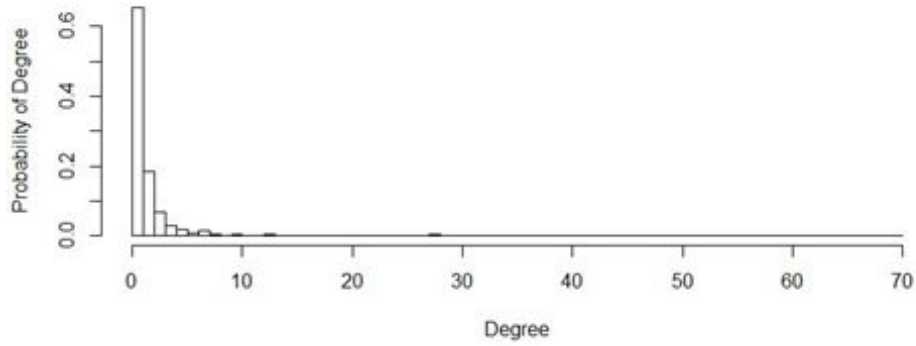

Figure 2.1: Degree distribution of the fat-tailed graph 1000 nodes

(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.

Solution:

Using a random walker, we measure and plot $\langle s(t) \rangle$ v.s. *t and* $\sigma^2(t)$ v.s. *t*, plots are shown in following figures. We plot the average, variance and standard deviation as well.

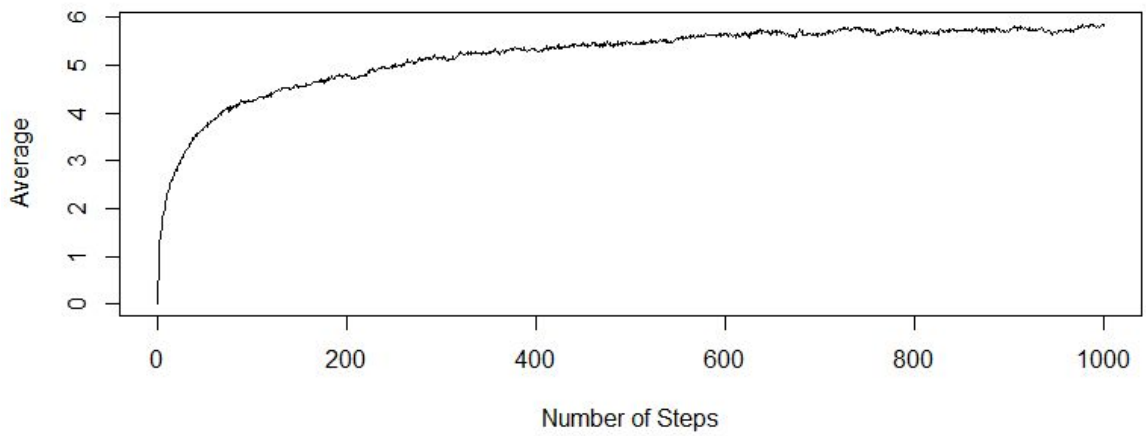

Figure 2.2: Average v.s. number of steps

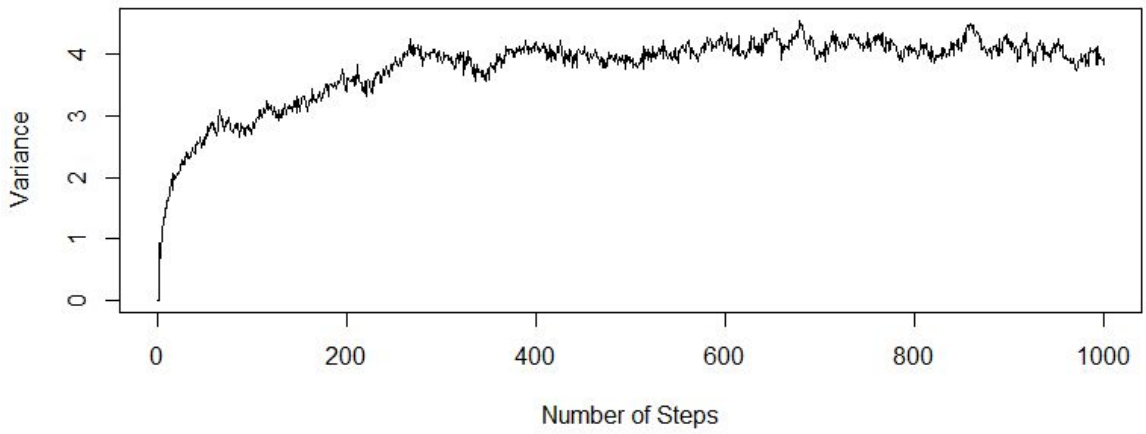

Figure 2.3: Variance v.s. number of steps

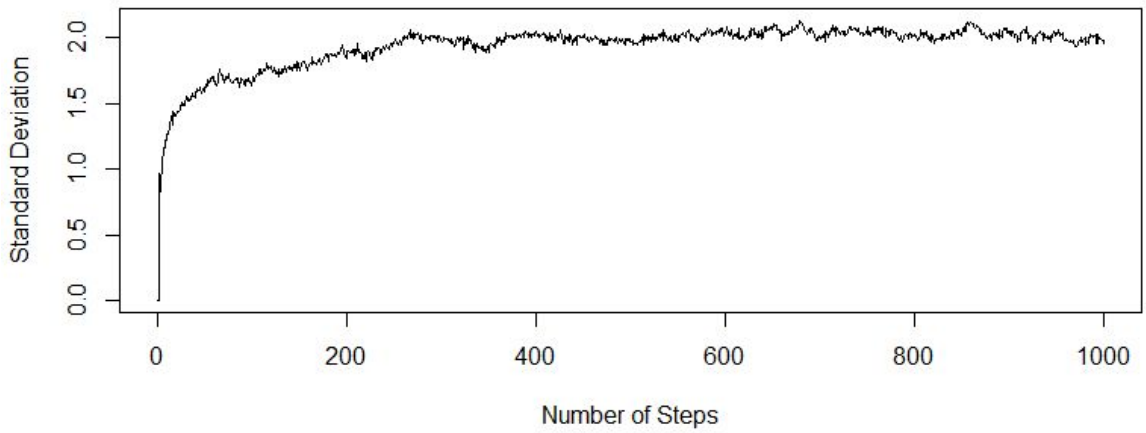

Figure 2.4: Standard Deviation v.s. number of steps

From the figures, we can see that the average distance, variance and standard deviation will converge to a positive number after certain number of steps. The average distance will converge to 6, while variance will converge to 4 and standard deviation to 2.

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

Solution:

From the figures, we can see that the average result is not similar to the result of that in d-dimensional space. The reason is that for d-dimensional, the distance could be negative, and also might be symmetric to which is positive, thus the average distance in d-dimensional space is likely to be near zero. But for the fat-tailed graph created here, the distance between every two nodes must be positive, thus the final average distance will converge to a positive number after certain steps of random walk iterations.

Also, from the figures we can also see that the result of variance and standard deviation are very similar to that of random walks in d-dimensional space. The figure shows nearly, which is similar to d-dimensional space.

(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?

Solution:

1) We first repeat (b) for a fat-tailed graph with 100 nodes and plot the figures as in question (b). The average, variance and standard deviation plots are shown in the following figures.

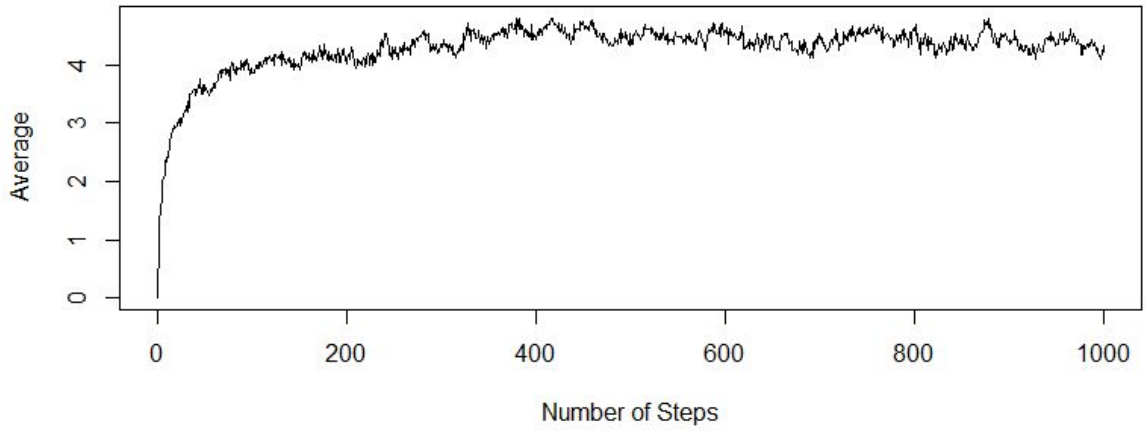

Figure 2.5: Average v.s. number of steps

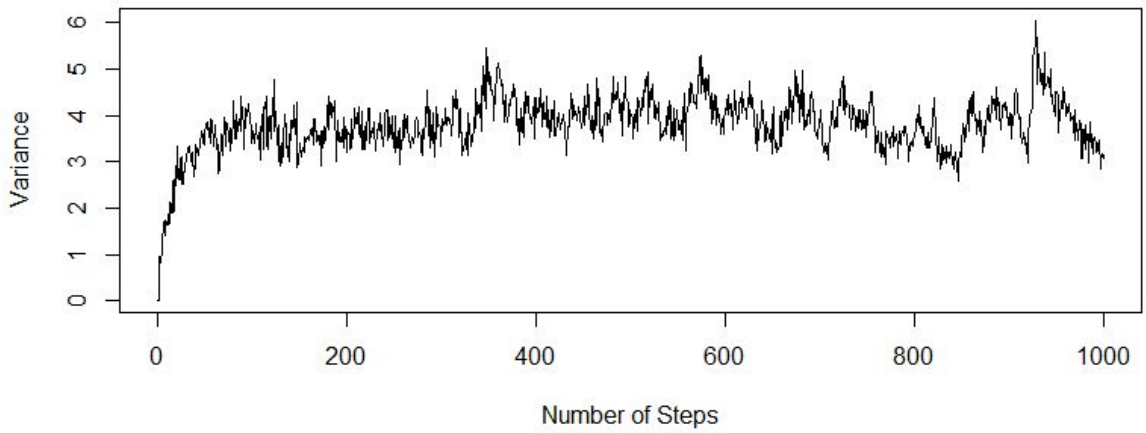

Figure 2.6: Variance v.s. number of steps

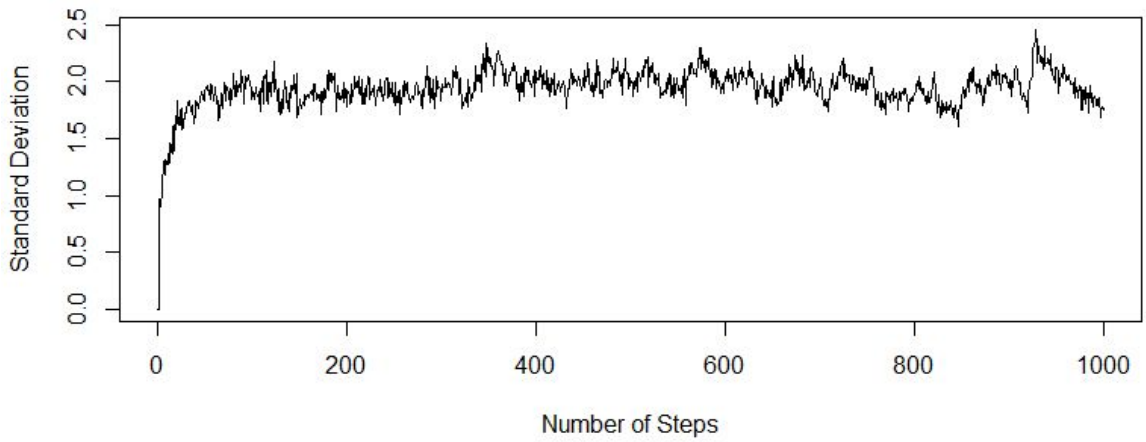

Figure 2.7: Standard Deviation v.s. number of steps

2) We then repeat (b) for a fat-tailed graph with 10000 nodes and plot the figures as in question (b). The average, variance and standard deviation plots are shown in the following figures.

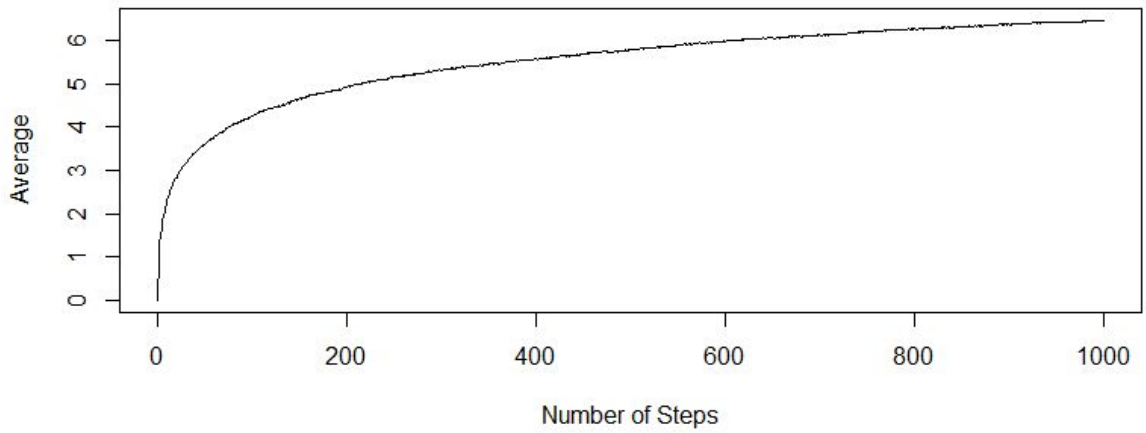

Figure 2.8: Average v.s. number of steps

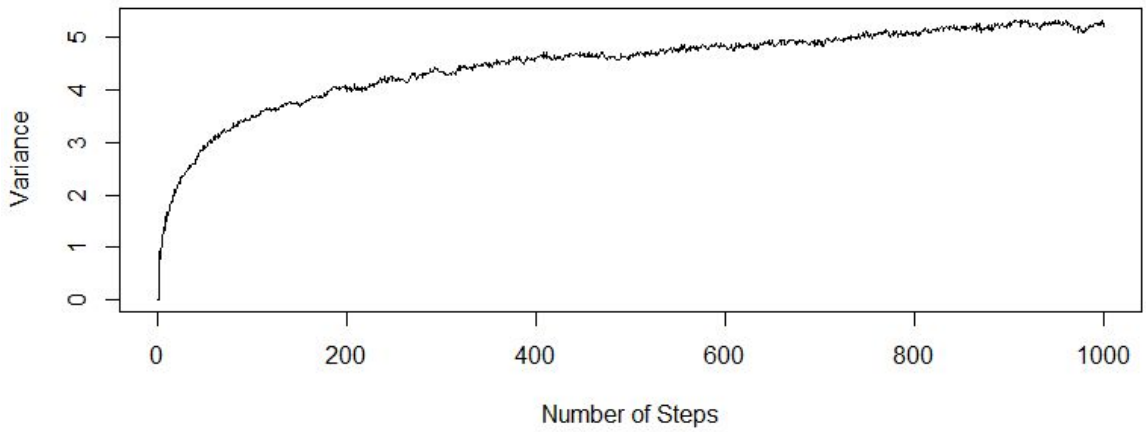

Figure 2.9: Variance v.s. number of steps

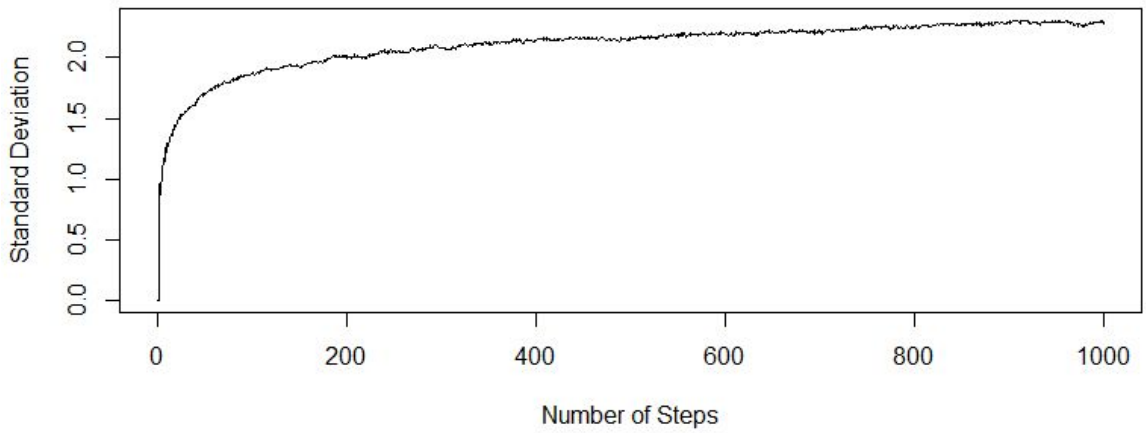

Figure 2.10: Standard Deviation v.s. number of steps

The diameter of fat-tailed graph with 100 nodes, 1000 nodes and 10000 nodes are 11, 16 and 23, which increases as the number of nodes grows. From the figures, we can see that graph with 100 nodes and 1000 nodes converge quite quickly while graph with 10000 nodes converge much slowly. Thus, we may conclude that this is due to the diameter. Graph with smaller diameter tends to converge quickly while graph with larger diameter tends to converge slowly, which is similar to the question 1. Besides, all the average, variance and standard deviation will have smaller fluctuations when the number of nodes grows and will fluctuate seriously when the number of nodes is small.

(e) 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?

Solution:

The degree distribution of the nodes reached at the end of the random walk on the 1000-node fat-tailed network is shown in the following figure, together with the original degree distribution.

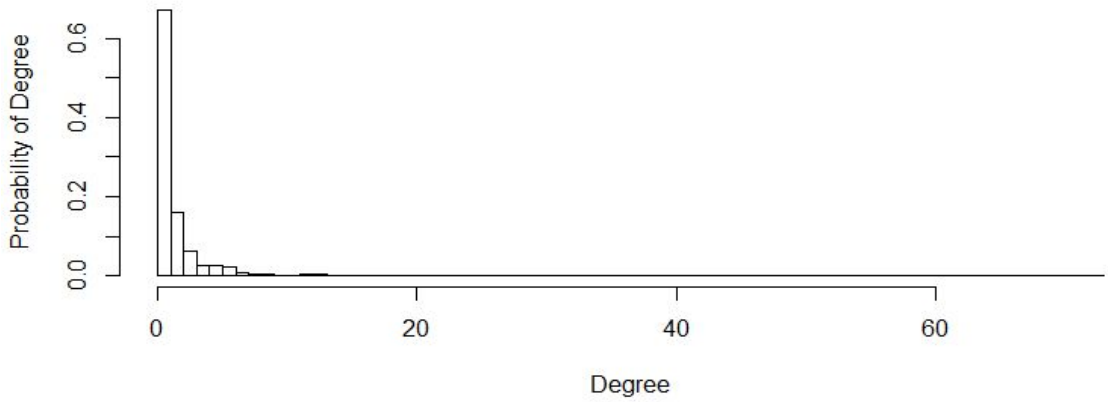

Figure 2.11: Original degree distribution

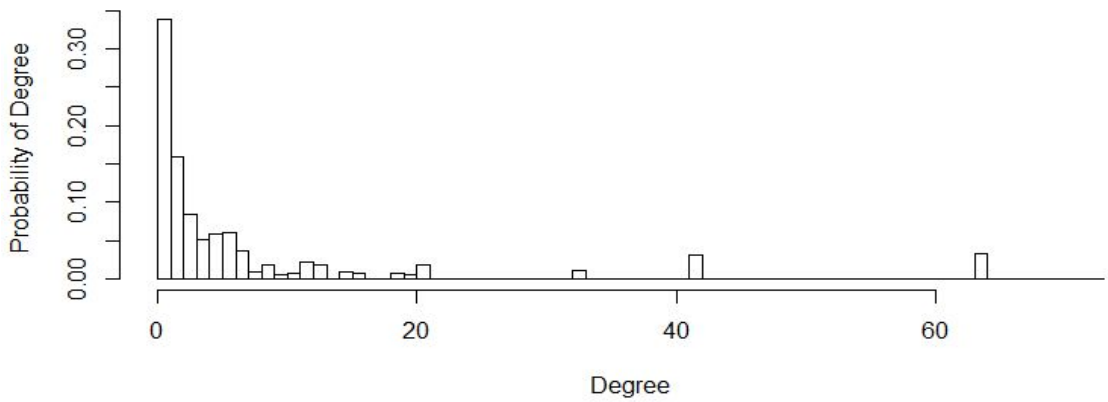

Figure 2.12: Degree distribution after random walk

The degree distribution after random walks is very similar to that before random walks, but similar to part (e) in question 1, the distribution tends to shift right and the probability of

large degree nodes tends to grow. Thus we can conclude that, for fat-tailed graph, random walk also tends to visit the node with higher degrees.

Question 3: 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?

Solution:

In this part, we first use the *random.graph.game* to generate an undirected graph with 1000 nodes, the probability for drawing an edge between any pair of nodes as 0.01 and use netrw() to generate random walk. After that, we compare the degrees of the graph and the probabilities that a walker visits each node with the correlated coefficient. The correlated coefficient value is 0.93977, which can show that this probability is positively related to the degree of the nodes. And the relation is close to linear. It is easy to understand because the graph is undirected and if a node has more edges, it is more possible for this node to be visited. The figure of the probability that the walker visits each node is shown in Figure 3.1. And the figure of the probability-degree is shown in Figure 3.2.

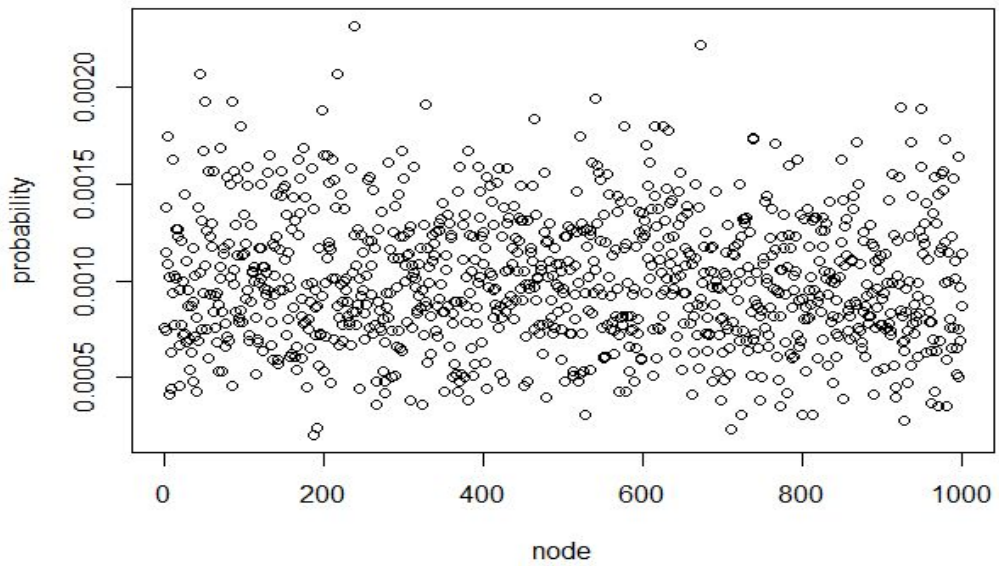

Figure 3.1 Probability that a walker visits each node for undirected graph with damping = 1

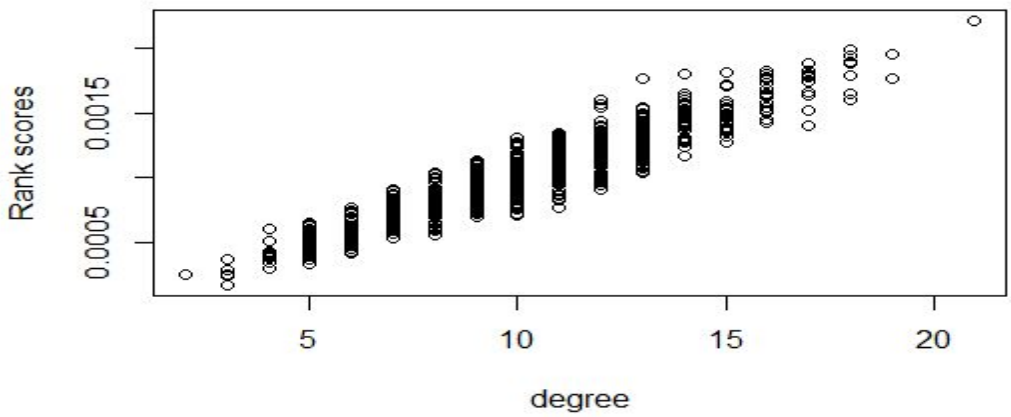

Figure 3.2 Probility-Degree for for undirected graph with damping = 1

(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?

Solution:

Compared to part (a), this part changes the undirected graph to directed graph. So we use the same functions and same parameters except 'directed' value which is set to TRUE. In this case, we get the figure of the probability that the walker visits each node shown in Figure 3.3. And the probability-degree figure is shown in Figure 3.4. And the correlated efficient for the graph's in-degrees and the probability that the walker visits each node is 0.86696, which is less than (a). It shows that such probability is positively related to graph's degree to some extent but that linearly related. This is because when the graph becomes directed, it has in-degree and out-degree. So the probability is affected by both in-degree and out-degree. We can see it directly from Figure 3.2 and 3.4.

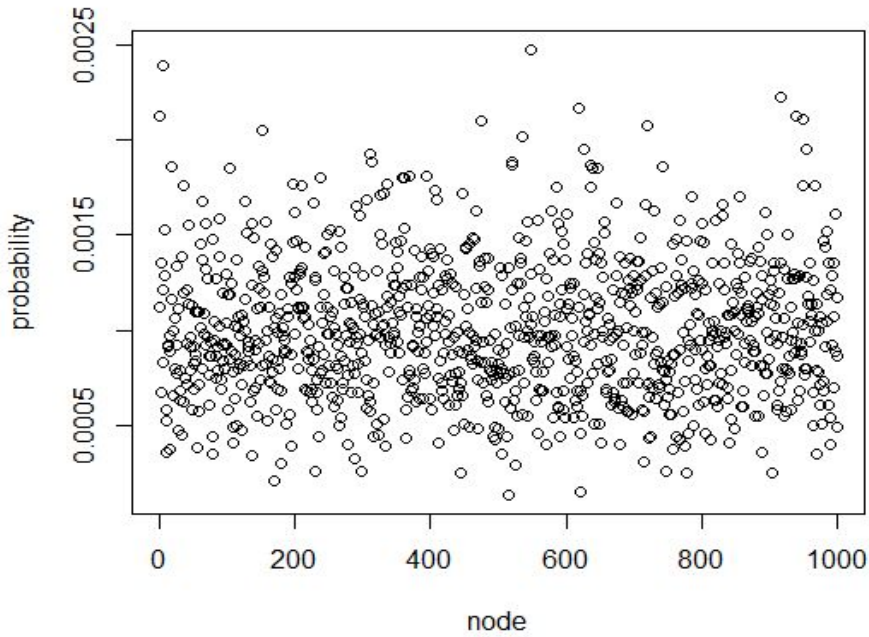

Figure 3.3 Probability that a walker visits each node for directed graph with damping = 1

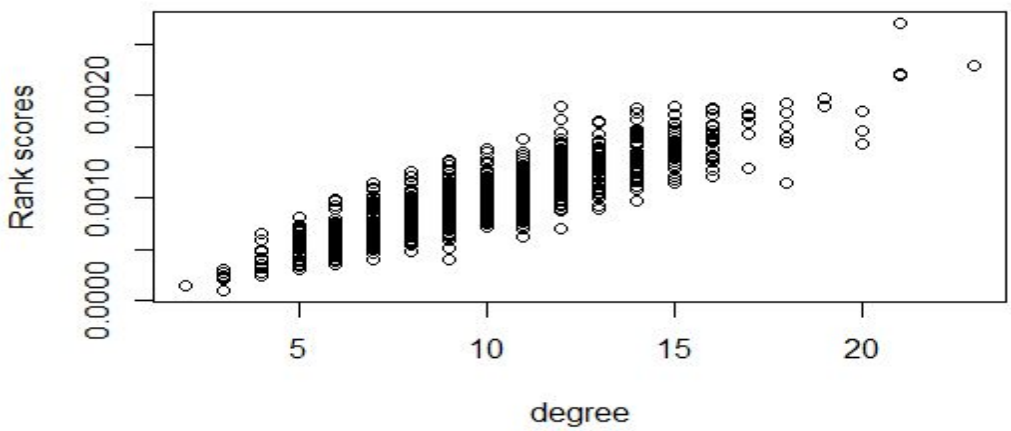

Figure 3.4 Probability-Degree directed graph with damping = 1

(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?

Solution:

If we set the damping value to 0.85, then for each step, the walker will have a probability equal to 0.85 of continuing random walking. So this will reduce the probability that a node will be visited by some walker, which in turn makes the probability that each node is visited less related to the degree of the node. So in this situation, we get the correlation coefficient as 0.91372 which is slightly less then (a). It is reasonable. And the figure of the probability that the walker visits each node shown in Figure 3.5. And the figure of the probability-degree is shown in Figure 3.6.

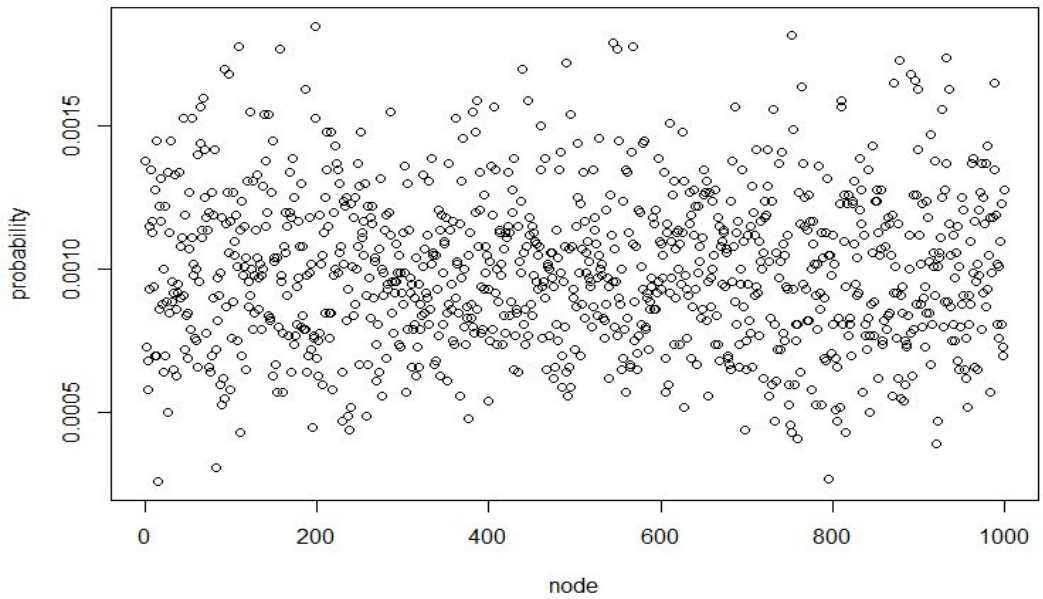

Figure 3.3 Probability that a walker visits each node for undirected graph with damping = 0.85

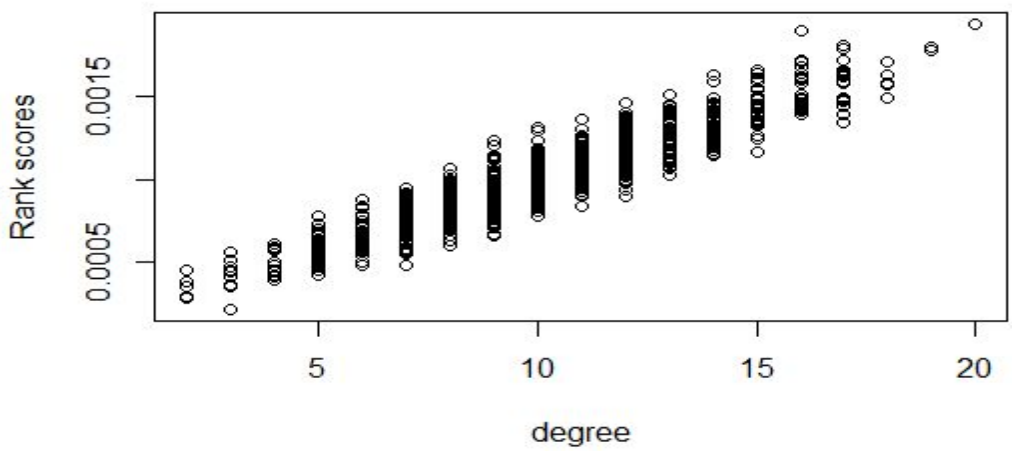

Figure 3.6 Probability-Degree for undirected graph with damping = 0.85

Question 4: Personalized PageRank

(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.

Solution:

We can see that this problem is very similar to *Question 3*. We can first generate a graph with p = 0.01 and number of nodes = 1000. And then generate random walk with damping parameter equal to 0.85. Finally, we can get the probability that a walker visits each node and use this probability value to represent the page rank metrics. We can see that if a page has a higher probability of being visited, it can be ranked higher. The rank scores which are represented by probability for each node are shown in Figure 4.1. And the pagerank-to-degree figure is shown in Figure 4.2. The correlation efficient for page rank and degrees is 0.870401. The variance for page rank is 9.19746e-8.

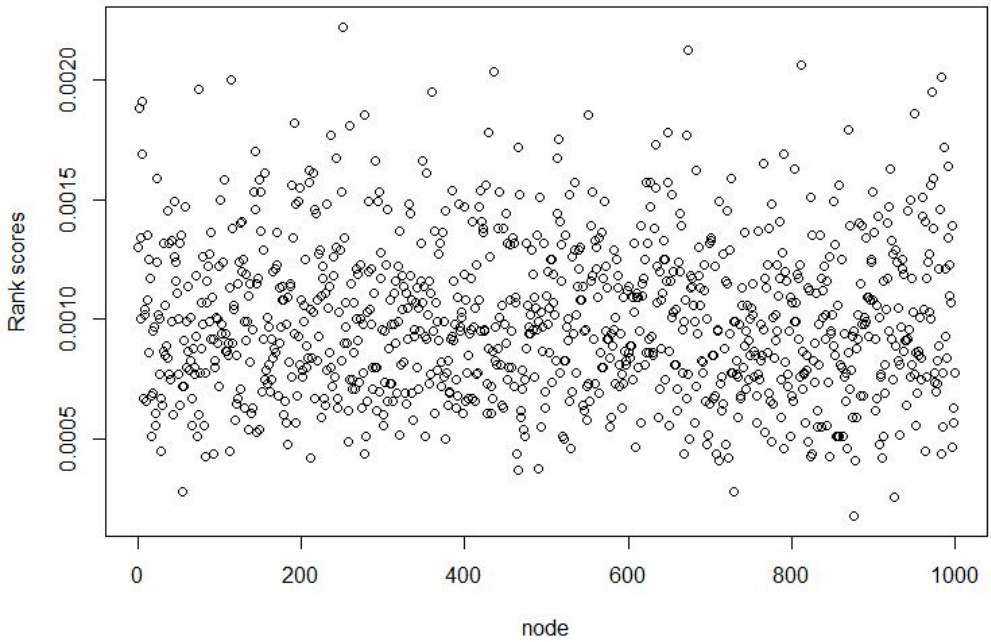

Figure 4.1 Rank scores for each node for pages with damping = 0.85

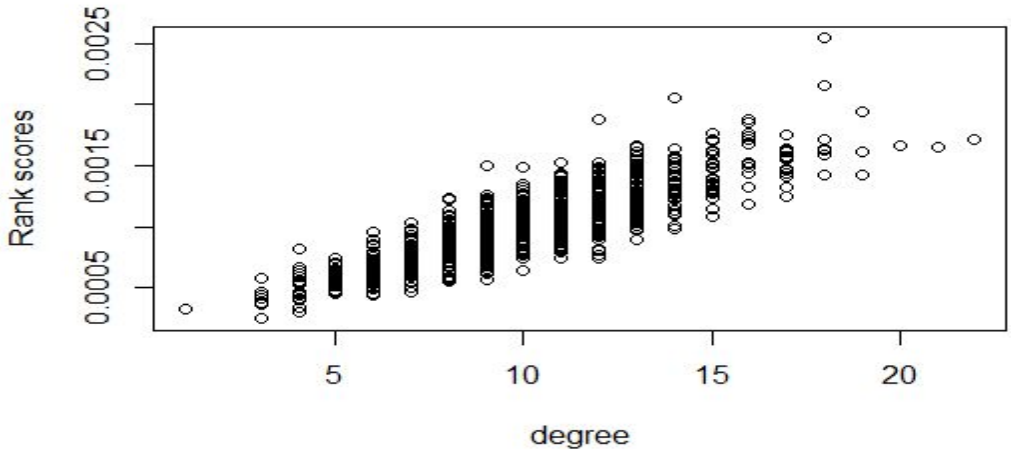

Figure 4.2 PageRank-Degree for pages with damping = 0.85

(b) Suppose you have your own notion of importance. You 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 teleprotation probability to each node is proportional to its PageRank (As opposed to the regular PageRank, where teleportation probability to all nodes are the dame and equal to 1/N). The damping parameter is equal to d = 0.85. Compare the results with (a).

Solution:

In this problem, we add our own notice of importance as a reference for rank scores. The teleportation probability to each node is proportional to its PageRank. So we first generate our page rank (as in (a)) without personalization. And then use this page rank scores (also known as probabilities) as teleportation probability to generate our own personalized PageRank. And with the damping parameter = 0.85, we get the final page rank scores for each node as shown in Figure 4.3.The figure of pagerank-degree is shown in Figure 4.4. The correlation coefficient for page rank and degrees is 0.85084. The variance of the page rank is 12.213e-8.

Compared with (a), we find that this correlation coefficient is slightly smaller. We think this is because we add personalization, which makes PageRank more complex. So it is more like nonlinear relationship between page rank and degrees, which makes the page rank less positively related to degrees. And as to variance, we can also see that due to personalization, the variance of page rank gets larger.

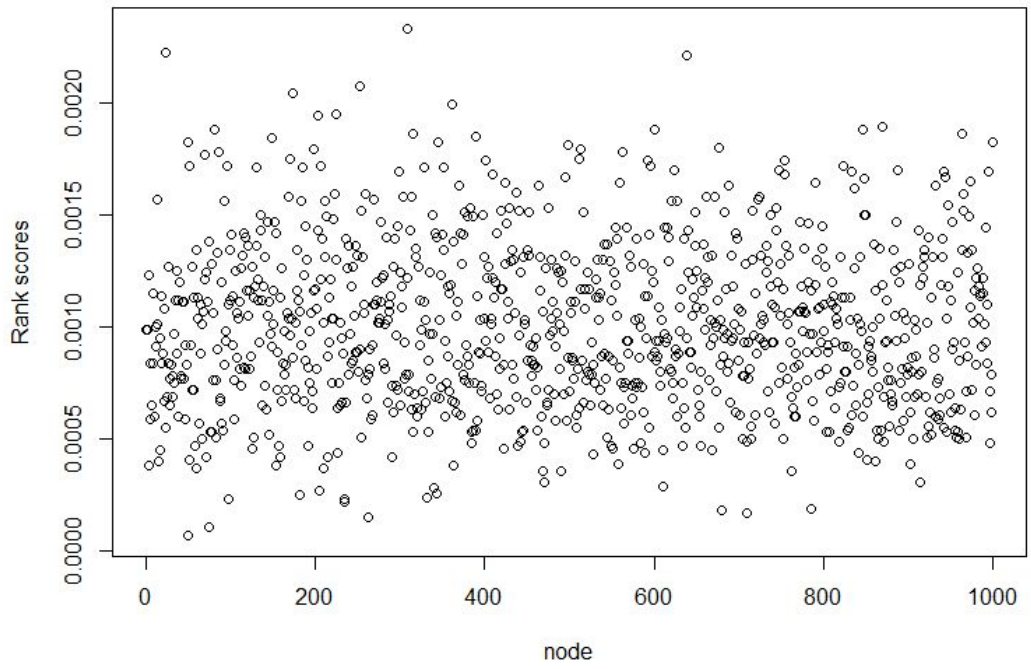

Figure 4.2 Personalized Rank scores for each node for pages with damping = 0.85

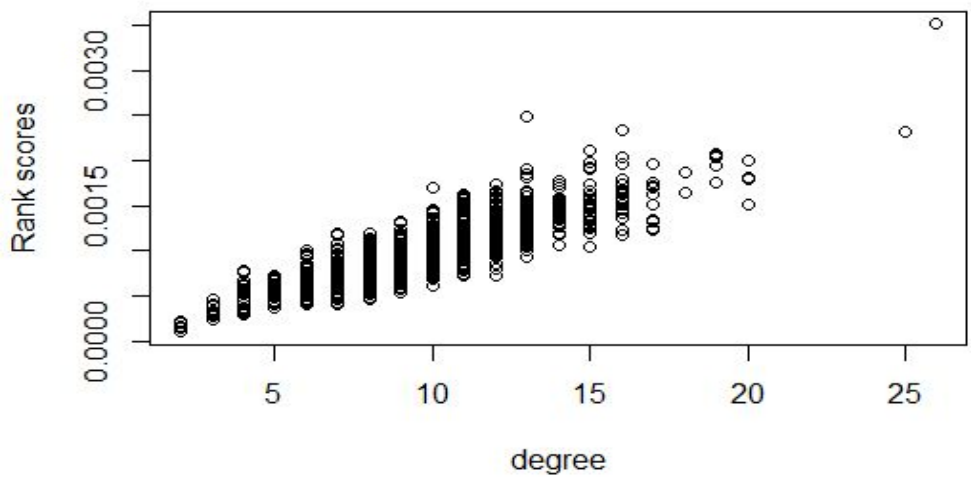

Figure 4.4 Personalized Rank score-degree for each node for pages with damping = 0.85

(c) More or less, (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 take into account the effect of this self-enforcement and adjust the PageRank equation?

Solution:

Taking into account the self-enforcement in (b), we can get the new equation for PageRank as below:

Let's assume that we have got the regular PageRank vector $\mathbf{P} = [P_1, P_2, ..., P_N]^T$ and the network matrix **M**, where if $M_{ij} = 1$, P_j points to P_i, if $M_{ij} = 0$, P_j doesn't point to P_i. So set the personalized Page Rank as a vector $\mathbf{PP} = [PP_1, PP_2, ..., PP_N]^T$. And we define some matrix calculation symbol:

$\mathbf{I_N} = [1,1, ..., 1]^T$, the length of $\mathbf{I_N}$ is N.

.*: $\mathbf{C} = \mathbf{A}.*\mathbf{B}$ means $c_{ij} = a_{ij} * b_{ij}$

./: $\mathbf{C} = \mathbf{A}./\mathbf{B}$ means $c_{ij} = a_{ij} / b_{ij}$

So the equation is:

$$\mathbf{PP} = (1\text{-}d) * \mathbf{P} + d * [\mathbf{P} * \mathbf{I_N}^T .* \mathbf{M} ./ (\mathbf{I_N} * (\mathbf{M}^T\mathbf{P})^T)] * \mathbf{PP}$$