EE 232E
Graphs and Network Flows
Homework 1
Winter 2016

Liqiang Yu, Rongjing Bai, Yunwen Zhu
904592975, 204587519, 304588434

04-12-2016

Contents

1 Problem1

In this part, we create random networks with probability p for drawing an edge between two arbitrary vertices and discussion the relationship between p and connectedness of the graph.

1.1 Part a

(i) when p = 0.01, we can get the degree distribution in Figure 1.

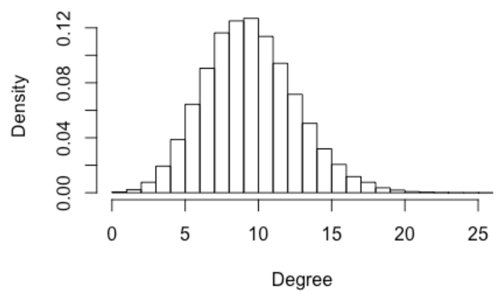

Figure 1: The degree distribution for p = 0.01

(ii) when p = 0.05, we can get the degree distribution in Figure 2.

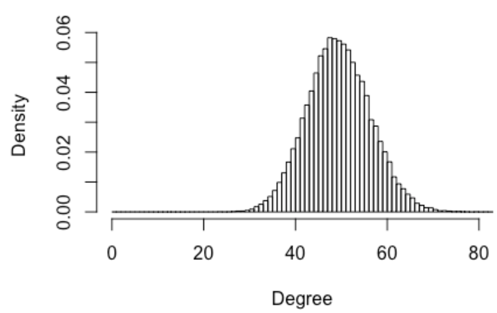

Figure 2: The degree distribution for p = 0.05

(iii) when p = 0.1, we can get the degree distribution in Figure 3.

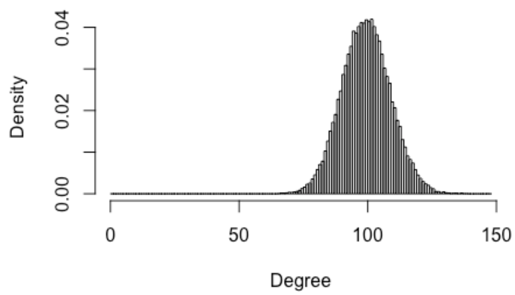

Figure 3: The degree distribution for p = 0.1

Based on the three figures, we can see that as the probability for drawing an edge between two arbitrary vertices increases, the degree increases conrespondingly, which conforms to our thought.

1.2 Part b

To check whether or not these network are connected and the diameter of these networks, we use is.connected() and diameter()function. And as this is the random network, we run the test for 50 times to get a more accurate result. Thus, in this way, we get the ratio of whether the network is connected and average diameter of the networks. The results are shown in Table 1.

Table 1: parameters of random network

	p = 0.01	p = 0.05	p = 0.1
ratio of connectedness	0.9	1	1
diameter	5.4	3	3

Based on the results, we can see that the network has a rather high ration of connectedness. Thus, all of these networks can be regarded as connected.

1.3 Part c

In order to find out the value of p_c, we start from the small value of p until the network become connected. And as this is the random network, we run the test for 100 times to get a more accurate result. Thus, in this way, we get $p_c =$ 0.006371, which roughly corresponds to the value in theory.

2 Problem 3

2.1 Part a

We create a evolving random undirected graph with 1000 nodes, the preferantial attachment exponent is 1, the aging exponent is 0 and the number of bins is 1000. The degree distribution is shown in figure 4.

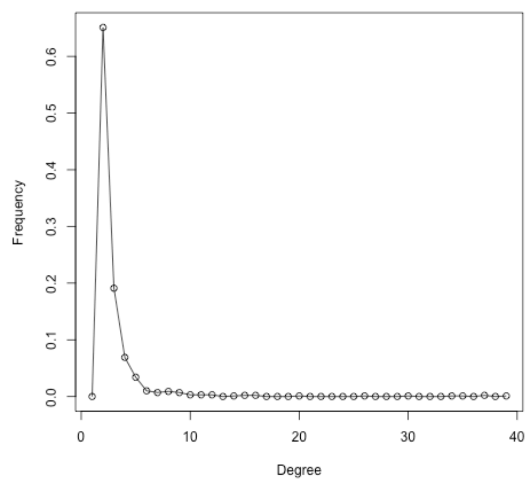

Figure 4: The degree distribution for evolving random graph

2.2 Part b

In order to calculate the modularity, we used the fast greedy algorithm. We ran the test for 100 times and calculated the average of modularity, the average modularity is 0.9225776. One of the community structure is shown in figure 5.

3 Problem 4

3.1 Part a

In this part, we used the forest fire model to simulate a growing network. The network contained 1000 nodes. Again, we ran the test for 100 times and calculated the avarage input and output degree distribution. Figure 6 shows the input degree distribution and figure 7 shows the output degree distribution. The figure 8 shows the result if we put the input and output degree in the same image, from which we can see that the input degree is approximately equal to the output degree.

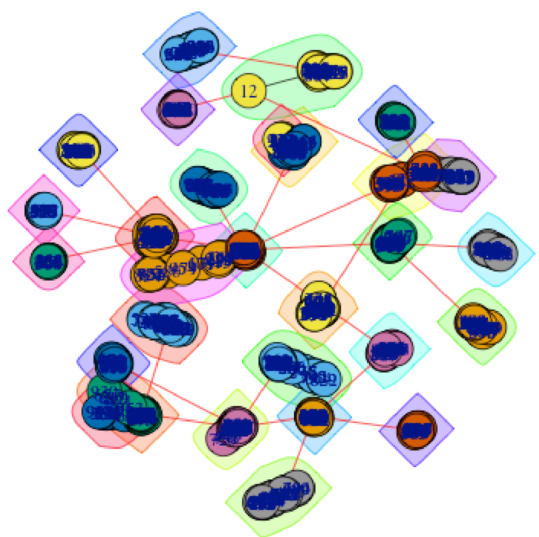

Figure 5: The community structure of the graph

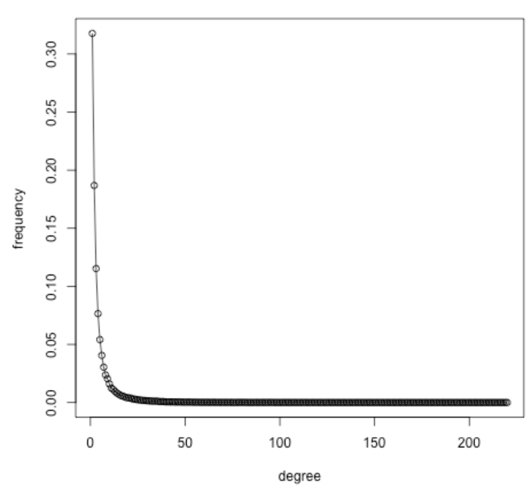

Figure 6: The input degree distribution

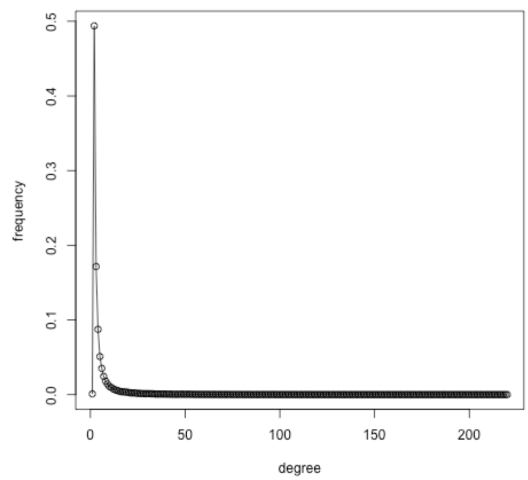

Figure 7: The output degree distribution

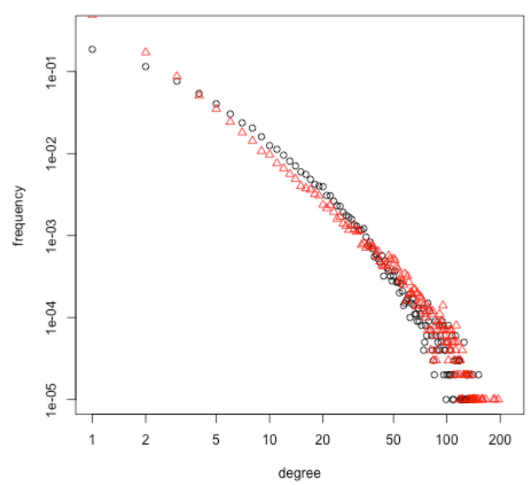

Figure 8: The intput and output degree distribution

3.2 Part b

We calculated the avarage diameter of the graph through 100 test and the diameter result is 10.46.

3.3 Part c

We chose to use random walks algorithm to calculate the community because it is more fit for directed graph. The tests were also run for 100 times and one of the community structures is shown in figure 9 and the average modularity is 0.42.

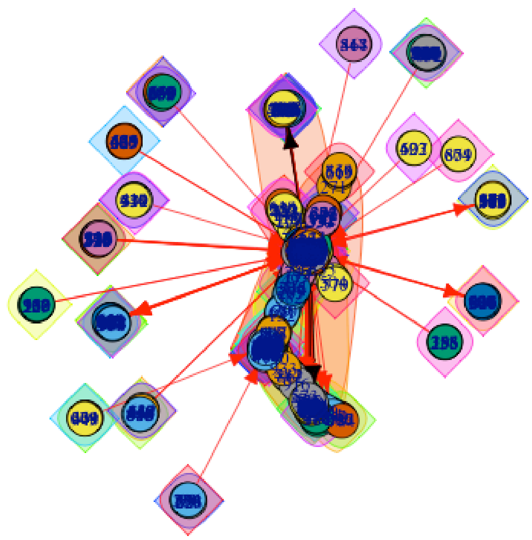

Figure 9: The community structure of the growing graph