Kaung Htet Myat

Project 1 – Random Graphs Models

CS 4740 – Networks, Crowds, and Markets

Feb 15, 2022

Deliverable 1: Implementing Random Graph in Code.

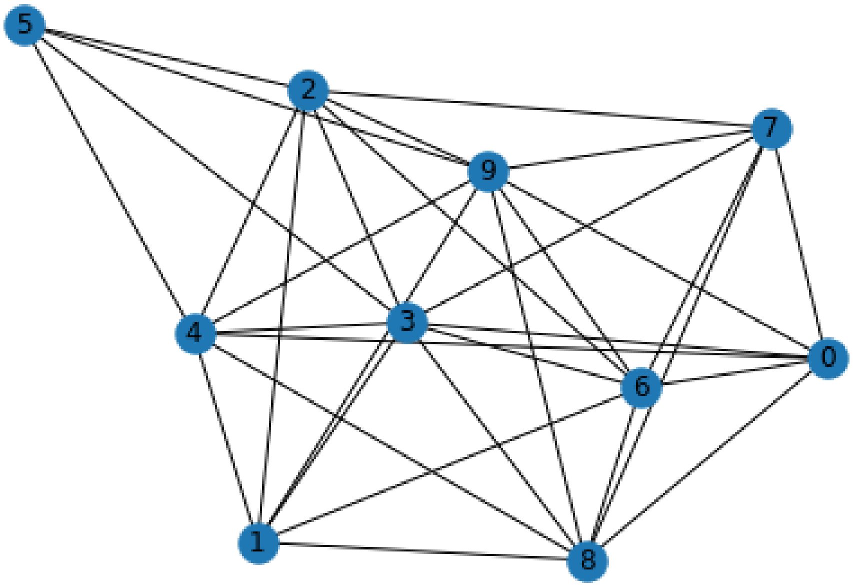
1A. Code

```
import networkx as nx
import matplotlib.pyplot as plt
import random

#n is the number of nodes and p is the probability of having an edge. For
this simple graph program I choose ten nodes with 0.75 chance of having an
 edge.
n = 10
p = 0.75

#create graph. Use for loop to add nodes.
Graph1 = nx.Graph()
for i in range(n):
  Graph1.add_node(i)

list(Graph1.nodes)

#for loop through all pair of nodes and generate a random possibiliy betwe
en 0 and 1. if random number < p, have an edge.
for i in range(n):
  for j in range(i+1, n):
    random_num = random.uniform(0,1)
    if random_num <= p:
      Graph1.add_edge(i,j)

list(Graph1.edges)

#draw graph
nx.draw(Graph1, with_labels = True)
```

1B. Code Explanation.

The library used are matplotlib, networkx and random. For the simple random Graph, I choose to have 10 nodes with possibility of 75% of having an edge. Then I create a graph. I use a for loop to add nodes into the Graph. To add the edges, I use a nested for loop where the first loop loops every node and the second loop loops from the next node to the end node. I generate a random number between 0 and 1 and if the possibility 0.75 greater than the number, I add an edge. Then I graph the graph.

1C. Graph.



1D. Graph Explanation and Analysis.

There are 10 different nodes with names ranging from 0-9. Because there are only 10 nodes, this graph could represent a tight friend group with a few people in the center since the probability is 75%. In a friend group like this, everybody has an equal number of people that they know. There may be a few people like node 3, 6, 4, and 9 that has more connecting with people in the group and are the centers of the group who make major decisions. Yet, node 1, 2, 7, 0, 8, and 5 may be friends that are introduced to the group by a mutual friend and begin having connections with other people in the group.

Deliverable 2: Number of Components and Probability

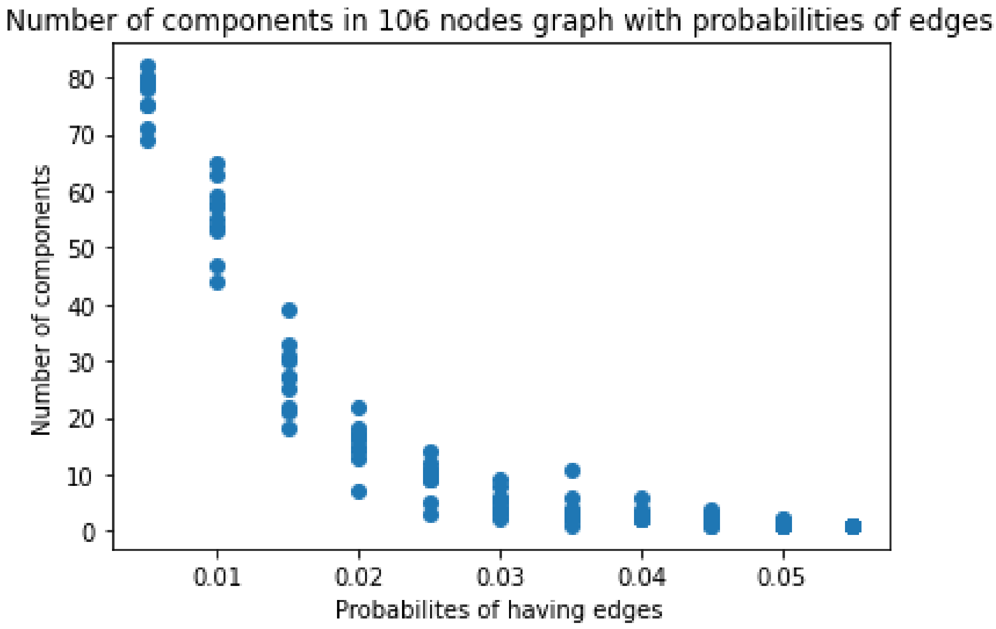
2A. Implementation Analysis

There are a total of 106 nodes. The possibility p value starts at 0.005 and 10 random graphs are generated for each value. The possibilities are increased by 0.005 value until they reach 0.055. For every value of p the 10 random graphs are generated and the total number of connected components are recorded. Then I graph the total number of components (all 10 values recorded) by p value. Secondly, I computed the average and graph the average number of components by the p value.

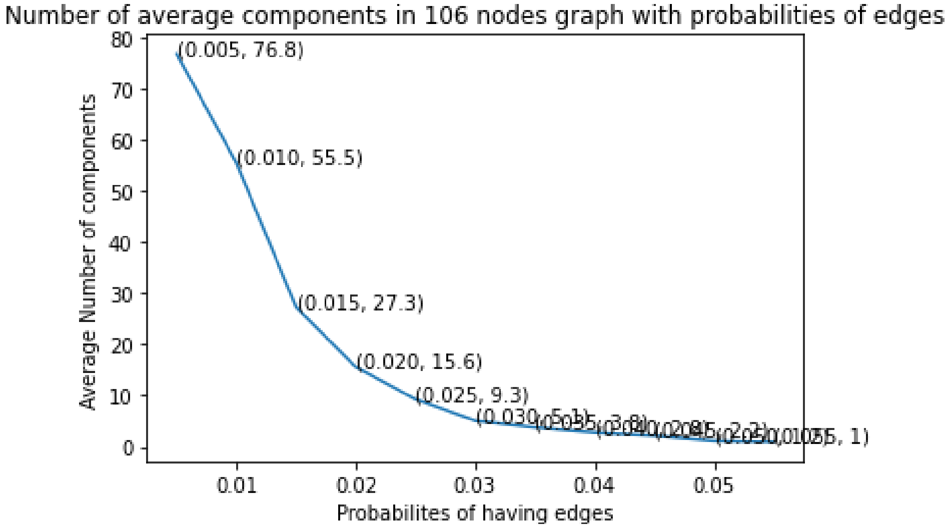
This is the number of components recorded by the value of p.

```
For p = 0.005              [79, 78, 75, 71, 69, 80, 82, 79, 80, 75]
For p = 0.010              [59, 54, 65, 47, 57, 63, 44, 58, 53, 55]
For p = 0.015              [31, 39, 33, 25, 27, 22, 30, 27, 21, 18]
For p = 0.020              [15, 18, 16, 17, 17, 13, 22, 17, 14, 7]
For p = 0.025              [11, 9, 12, 11, 10, 3, 14, 9, 5, 9]
For p = 0.030              [9, 4, 3, 8, 4, 5, 2, 6, 6, 4]
For p = 0.035              [6, 4, 3, 3, 2, 4, 2, 11, 2, 1]
For p = 0.040              [3, 2, 2, 3, 2, 2, 4, 6, 2, 2]
For p = 0.045              [3, 2, 1, 2, 4, 2, 1, 1, 3, 3]
For p = 0.050              [1, 1, 2, 1, 1, 1, 2, 1, 1, 1]
For p = 0.055              [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

2B. Number of components by Possibilities Plot



2C. Average number of Components graph



2D. Analysis

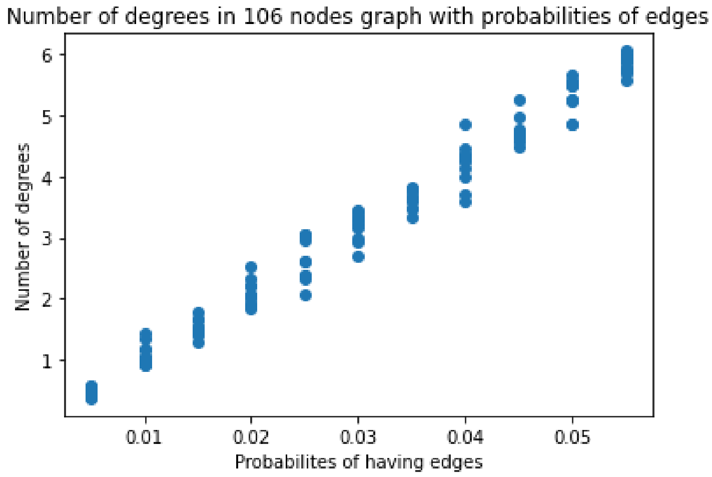
As seen in the two graphs above, the number of components decreases as probabilities of having edges increases. The number of components converges to 1 at around 0.055 which is around 5.5%. The reason to why the number converges at such a low possibility of 5.5 is that there are a larger number of nodes. In this case, the number of nodes could be people. Mathematically 5.5 would result in around 5-6 friends for each node. As a result, although the number of friends circle is small, the larger population with random friend group results a connected graph. One important thing to note in the graph is that the graph is random which is also the reason to why a low possibility of 5.5% result in connected graph. In real society, people tend to make friends with people who have similar interest, racial and social-economic background. This would result in groups in society with bridges in between the different groups rather than random friends.

Deliverable 3: Degrees of Nodes and Probability

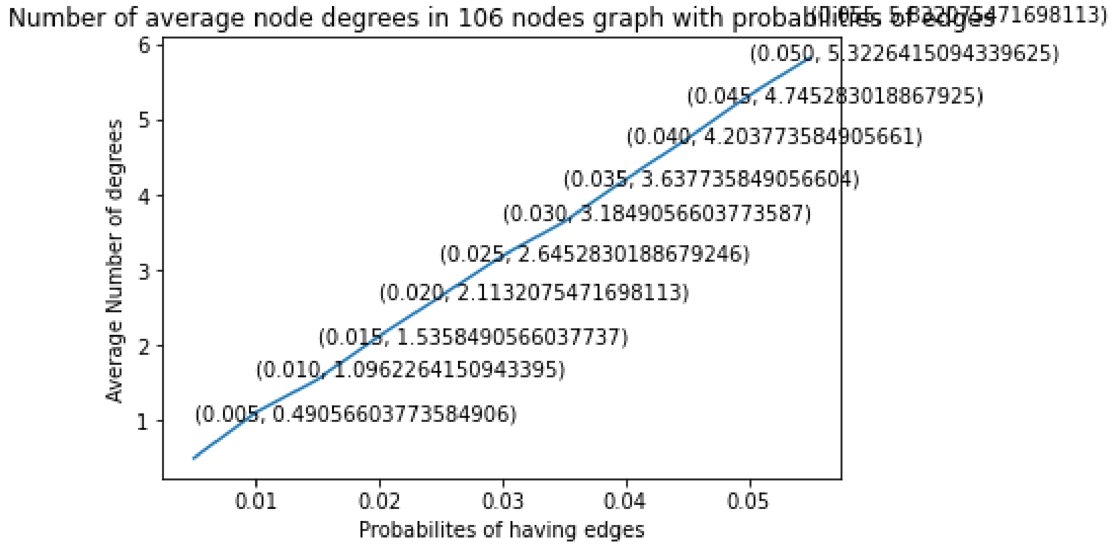
3A. Implementation Analysis

There are a total of 106 nodes. The possibility p value starts at 0.005 and 10 random graphs are generated for each value. The possibilities are increased by 0.005 value until they reach 0.055. For every value of p the 10 random graphs are generated and the total number of degrees for each node are recorded. Then I calculated the average by total degrees/ number of nodes. Then I graph the total number of degrees (all 10 values recorded) by p value. Secondly, I computed the average and graph the average number of degrees by the p value.

3B. Number of degrees by Possibilities Plot



3C. Average number of Degrees graph



3D. Analysis

Comparing the average component graph and the average number of degree graph, we can see that it takes only about 5-6 connection from every node to create a connected graph. From the graph at 0.005 p value, the average degree of node is 0.49. However, at 0.055 p value, the average degree of node is 5.8. This proves that it only needs around 5.5% value of p to reach a fully connected graph. As p increase, the degree of nodes increases. Therefore, we can conclude that it requires around p value of 0.055 to reach a fully connected graph where there are around 6 degrees.