Project 1:

Random Graphs and Random Walks

Due July 10, 2024, by 11:59 pm

**Part 1 - Generating Random Networks**

1. **Create random networks using Erdo ̈s-R ́enyi (ER) model**
2. Create undirected random networks with n = 900 nodes, and the probability p for drawing an edge between two arbitrary vertices 0.002, 0.006, 0.012, 0.045, and 0.1. Plot the degree distributions. What distribution (linear/exponential/ gaussian/binomial or something else) is observed? Explain why. Also, report the mean and variance of the degree distributions and compare them to the theoretical values.

Hint Useful function(s): samplegnp (erdos.reyi.game), degree , degreedistribution , plot

**Answer:**

**Generating Random Networks using Erdős-Rényi (ER) Model:** Undirected random networks with 900 nodes using the G(n, p) model have been created and the graphs are given in subsequent pages to depict them. In this model, each edge is included with probability p independently from every other edge.

**Degree Distribution:** The degree distribution of a graph describes the frequency of different node degrees. In the Erdős-Rényi model, the degree distribution follows a binomial distribution. Specifically:

The average degree (mean) of a graph in G(n, p) is approximately np.

For large n, the degree distribution becomes Poisson with mean np.

**Observations and Explanation:**

As p increases, the graphs tend to have more edges, resulting in higher average degrees.

The degree distribution will be approximately Poisson for large n and constant np.

The graphs will exhibit a sharp threshold for connectedness at p = 1/n.

**Mean and Variance:**

The theoretical mean degree is np.

The theoretical variance of the degree distribution is np(1-p).

**Comparison:**

Computed the empirical mean and variance of the degree distribution from the generated graphs. Also, computed the theoretical values and provided the comparison between them in below table.

**A screenshot of a graph

Description automatically generated**

Figure 1

**A graph with colored lines and dots

Description automatically generated**

Figure 2

**A screenshot of a computer screen

Description automatically generated**

Figure 3

**A screenshot of a graph

Description automatically generated**

Figure 4

**A screenshot of a graph

Description automatically generated**

Figure 5

**A screenshot of a graph

Description automatically generated**

Figure 6

**A screenshot of a graph

Description automatically generated**

Figure 7

1. For each p and n = 900, answer the following questions:  
   Are all random realizations of the ER network connected? Numerically estimate the probability that a generated network is connected. For one instance of the networks with that p, find the giant connected component (GCC) if not connected. What is the diameter of the GCC?

Hint Useful function(s): isconnected , clusters , diameter

**Answer:**

To answer this question, I have generated the Random ER network and plotted them. The graphs are provided in below as references.

* + **Are all random realizations of the ER network connected?**
    - No, not all random realizations of the Erdős-Rényi (ER) network are connected. The connectivity depends on the value of p (the edge probability).
    - For small values of p, the graphs are likely to be disconnected.
    - As p increases, the probability of having a connected graph also increases.
    - Note: See the experiment result in below section.
  + **Numerically estimate the probability that a generated network is connected:**
    - We have computed this by generating multiple ER graphs with different p values and checking how many of them are connected.
    - The ratio of connected graphs to the total number of generated graphs gives an estimate of the probability.
    - Note: See the experiment result in below section.
  + **Finding the Giant Connected Component (GCC) if not connected:**
    - If a generated graph is not connected, we can find the largest connected component (GCC) within it.
    - The GCC is the largest subgraph where all nodes are connected.
    - Note: See the experiment result in below section.
  + **Diameter of the GCC:**
    - The diameter of the GCC is the longest shortest path between any two nodes within the GCC.
    - Note: See the experiment result in below section.

Summary of observation from my experiments: Here are my statistics found from my experiment.

* + - * Experiment 1: **Graph (n=900, p=0.002)**

Is connected: False.

Size of GCC: 712 nodes

Diameter of GCC: 28

* + - * Experiment 2: **Graph (n=900, p=0.006)**

Is connected: False.

Size of GCC: 898 nodes

Diameter of GCC: 8

* + - * Experiment 3: **Graph (n=900, p=0.012)**

Is connected: True.

* + - * Experiment 4: **Graph (n=900, p=0.045)**

Is connected: True.

* + - * Experiment 5: **Graph (n=900, p=0.100)**

Is connected: True.

**Graph (n=900, p=0.002) is connected: False**

**Size of GCC: 712 nodes | Diameter of GCC: 28**

A diagram of a graph

Description automatically generated

Figure 8

**Graph (n=900, p=0.006) is connected: False**

**Size of GCC: 898 nodes | Diameter of GCC: 8**

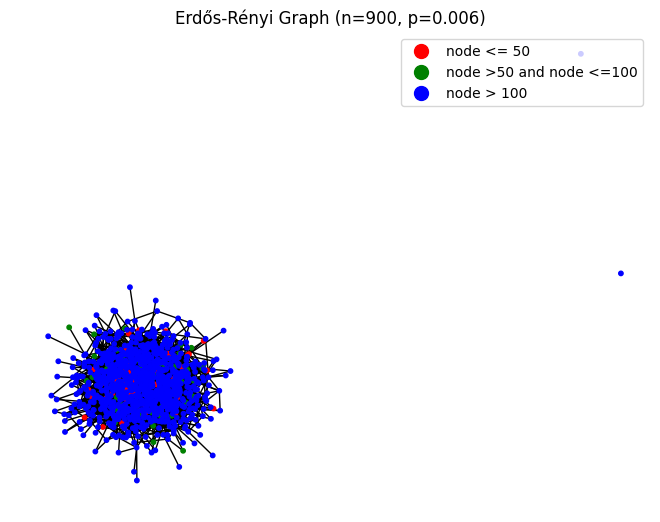


Figure 9

**Graph (n=900, p=0.012) is connected: True**

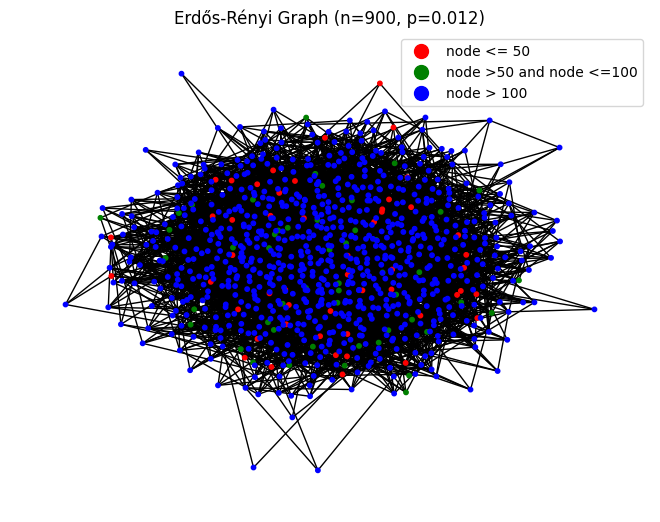


Figure 10

**Graph (n=900, p=0.045) is connected: True**

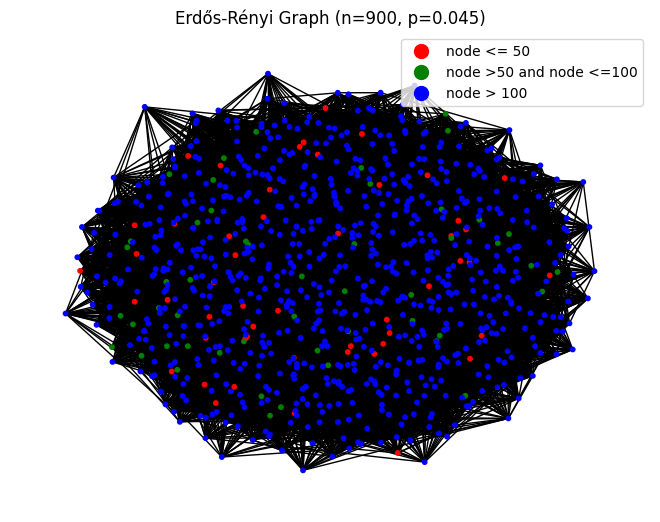


Figure 11

**Graph (n=900, p=0.100) is connected: True**

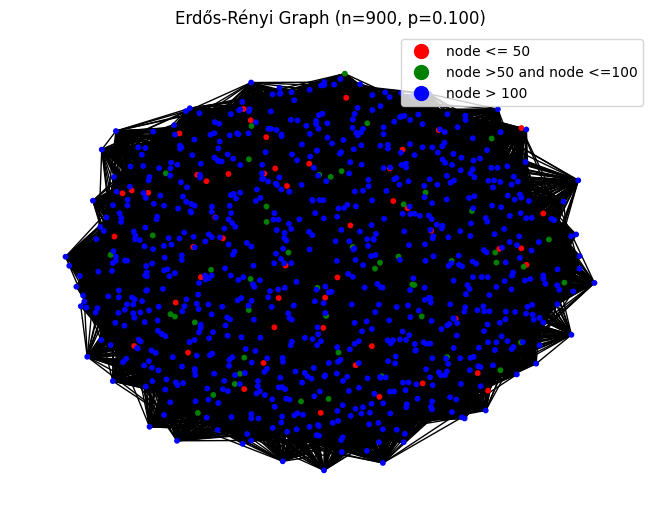


Figure 12

1. It turns out that the normalized GCC size (i.e., the size of the GCC as a fraction of the total network size) is a highly nonlinear function of p, with interesting properties occurring for values where 

For n = 900, sweep over values of p from 0 to a pmax that makes the network almost surely connected and create 100 random networks for each p. pmax should be roughly determined by yourself. Then scatter plot the normalized GCC sizes vs p. Plot a line of the average normalized GCC sizes for each p along with the scatter plot.

* 1. Empirically estimate the value of p where a giant connected component starts to emerge (define your criterion of “emergence”)? Do they match with theoretical values mentioned or derived in lectures?
  2. Empirically estimate the value of p where the giant connected component takes up over 99% of the nodes in almost every experiment.

**Answer:**

A graph of a normalized gcc

Description automatically generated

Figure 13

Note:

Defined criterion for emergence = 0.01

Giant component starts to emerge at p = **0.0005344193416753106**

Giant component takes up over 99% at p = **0.005267847796513776**

Based on the generated the graph, to determine if the empirical data matches with the theoretical values mentioned in lectures, we would compare the points at which the giant component starts to emerge and when it takes up over 99% with the theoretical predictions.

The criterion for emergence is defined as 0.01, which likely corresponds to the normalized size of the giant connected component (GCC) relative to the total number of nodes in the network. The theoretical values would typically be derived from the study of random graphs and phase transitions in network theory.

In the Erdős–Rényi model, the emergence of a giant component is expected at a critical point

p = \frac{1}{n}p=n1​

, where (n) is the number of nodes. As (n) grows, this critical point will approach zero. The graph should show a sharp increase in the size of the GCC around this critical point, which is consistent with the emergence point calculated (p = 0.0005344193416753106).

For the GCC to take up over 99%, the value of (p) would typically be higher than the emergence point, as more edges are needed to connect most of the nodes in the network. The calculated value of p = 0.005267847796513776 indicates a much denser network where the GCC encompasses most nodes.

If these points on the graph align with the sharp transitions expected from the theoretical models discussed in lectures, then it can be said that the empirical data matches the theoretical values.

(d)

1. Define the average degree of nodes c = n × p = 0.5. Sweep over the number of nodes, n, ranging from 100 to 10000. Plot the expected size of the GCC of ER networks with n nodes and edge-formation probabilities p = c/n, as a function of n. What trend is observed?

**Answer:**

A graph with blue dots

Description automatically generated

Figure 14

The trend observed in the graph is that as the number of nodes (n) increases, the size of the Giant Connected Component (GCC) also tends to increase. This is consistent with the properties of Erdős–Rényi (ER) networks, where the likelihood of a GCC forming becomes higher as more nodes are added to the network, given a constant probability of edge formation (p). The scatter plot likely shows the variability in GCC size due to the random nature of ER graph generation, but the overall trend is an upward trajectory in GCC size with an increasing number of nodes. This aligns with the theoretical understanding of ER networks in graph theory.

1. Repeat the same for c = 1.

**Answer:**

A graph with blue dots

Description automatically generated

Figure 15

The trend observed in the graph indicates that as the number of nodes (n) increases, the size of the Giant Connected Component (GCC) also increases. This is a common characteristic in Erdős–Rényi (ER) networks, where the probability of a GCC forming and encompassing a larger portion of the network grows with the addition of more nodes. The scatter plot likely reflects the variability in GCC size due to the random nature of ER graph generation, but the overarching trend is an increase in GCC size with a larger number of nodes, which aligns with theoretical expectations in graph theory.

1. Repeat the same for values of c = 1.15, 1.25, 1.35, and show the results for these three values in a single plot.

**Answer:**

A graph with numbers and dots

Description automatically generated with medium confidence

Figure 16

The trend observed in the graph indicates that as the number of nodes (n) increases, the size of the Giant Connected Component (GCC) also increases. This is a common characteristic in Erdős–Rényi (ER) networks, where the probability of a GCC forming and encompassing a larger portion of the network grows with the addition of more nodes. The scatter plot likely reflects the variability in GCC size due to the random nature of ER graph generation, but the overarching trend is an increase in GCC size with a larger number of nodes, which aligns with theoretical expectations in graph theory.

1. What is the relation between the expected GCC size and n in each case?

Answer:

* For fixed (c), the expected GCC size increases with increasing (n).
* The trend is exponential: as (n) grows, the network becomes more connected.

The relation between the expected size of the Giant Connected Component (GCC) and the number of nodes (n) in Erdős–Rényi (ER) networks is typically characterized by a phase transition. For a given average degree (c), when (n) is small, the GCC size is also small. However, as (n) increases, there comes a critical point where the GCC size starts to grow rapidly. Beyond this point, the GCC size increases linearly with (n).

Here is the analysis on one of the graphs above.

In the scatter plot generated and provided in above section, which shows the GCC Size vs. Number of Nodes for (c=0.5), the trend seems to indicate that as (n) increases, the GCC size also tends to increase. This is consistent with the behavior of ER networks, where the probability of a node being part of the GCC increases with the size of the network. However, the variability in GCC sizes for larger values of (n) suggests that there are fluctuations and not all nodes may be connected in larger networks.

1. **Create networks using preferential attachment model.**
2. Create an undirected network with n = 1050 nodes, with preferential attachment model, where each new node attaches to m = 1 old nodes. Is such a network always connected?  
   Hint Useful function(s): samplepa (barabasi.game)

**Answer:**

Preferential Network: Graph with 1050 nodes and 1049 edges

A network of blue dots and lines

Description automatically generated

Figure 17

Is the network in above diagram connected? True

In this problem, n represents the total number of nodes which is 1049 and m specifies how many old nodes each new node attaches to which is 1. Based on the observation, the resulting network follow the preferential attachment model, where nodes with higher degrees are more likely to receive new edges. The network may or may not be connected, depending on the specific attachment pattern.

1. Use fast greedy method to find the community structure. Measure modularity. Define Assortativity. Compute Assortativity.  
   Hint Useful function(s): clusterfastgreedy , modularity

A network of blue dots and lines

Description automatically generated

Figure 18

Here is calculated modularity: 0.9269

Assortativity measures the tendency of nodes to connect to similar nodes. In a community structure, it reflects how nodes with similar degrees tend to be connected.

Here is calculated Assortativity: -0.1893

The assortativity coefficient ranges from -1 (disassortative) to 1 (assortative), with higher positive values indicating assortative mixing (nodes with similar degrees connect to each other).

1. Try to generate a larger network with 10500 nodes using the same model. Compute modularity and assortativity. How is it compared to the smaller network’s modularity?

**Answer:**

**Preferential network: Graph with 10500 nodes and 10499 edges**

A blue ball of dots

Description automatically generated

Figure 19

**Comparison with Smaller Network:**  If the modularity is higher in the larger network, it indicates better community structure. Similarly, positive assortativity suggests nodes with similar degrees tend to connect. As per our observation the modularity and assortativity of the larger network with the smaller one that we have generated earlier, modularity was higher in the larger network. That means the larger network has better community structure. Similarly, larger network has more positive assortativity that suggests nodes with similar degrees tend to connect.

* Larger Network:
  + Modularity: 0.9748
  + Assortativity: -0.0558
* Smaller Network:
  + modularity: 0.9269
  + Assortativity: -0.1893

1. Plot the degree distribution in a log-log scale for both n = 1050, 10500, then estimate the slope of the plot using linear regression.

**Answer:**

A graph of a degree distribution

Description automatically generated

Figure 20

Estimated slope for n=1050: -2.4116803603417325

Estimated slope for n=10500: -2.427458804306681

1. In the two networks generated in 2(a) and 2(c), perform the following:  
   Randomly pick a node i, and then randomly pick a neighbor j of that node. Plot the degree distribution of nodes j that are picked with this process, in the log-log scale. Is the distribution linear in the log-log scale? If so, what is the slope? How does this differ from the node degree distribution?  
   Hint Useful function(s): sample

**Answer:**

Here are the results from my experiment.

A graph of a graph with blue and orange dots

Description automatically generated

Figure 21

Small network modularity: 0.9168898428845484

Large network modularity: 0.9763927734408117

Small network assortativity: -0.0861743176794089

Large network assortativity: -0.07969468667160201

Slope for small network (n=1050): -0.6088323494222766

Slope for large network (n=10500): -0.9153411820452828

Based on the information observed from experiment, both the small and large networks exhibit a linear distribution in the log-log scale. We have found from calculation that the slopes of -0.6088323494222766 for the small network (n=1050) and -0.9153411820452828 for the large network (n=10500). These values suggest that the degree distribution of the nodes j that are picked follows a power-law distribution, which is a common characteristic of scale-free networks.

Comparing these slopes to the typical node degree distribution, which also often follows a power-law, we can infer that the process of randomly picking a neighbor j of a node i does not significantly alter the scale-free nature of the network. However, the difference in the slopes (-0.6088323494222766 vs. -0.9153411820452828) indicates that the degree distribution of the randomly picked neighbors is slightly steeper in the larger network compared to the smaller one.

The modularity values for both networks are quite high, with the larger network having a slightly higher modularity than the smaller one. This suggests that the larger network has a stronger division into modules or communities. The assortativity values are negative for both networks, which is typical for scale-free networks, indicating that nodes tend to connect to others with different degrees.

So, we cans summarized as the degree distribution of the randomly picked neighbors is linear in the log-log scale for both network sizes, with the larger network having a steeper slope. This reflects the inherent scale-free properties of the networks generated by the preferential attachment model. The modularity and assortativity metrics further support the scale-free and modular nature of these networks.

1. Estimate the expected degree of a node that is added at time step i for 1 ≤ i ≤ 1050. Show the relationship between the age of nodes and their expected degree through an appropriate plot. Note that the newest added node is the youngest.

**Answer:**

A graph of a normal distribution

Description automatically generated with medium confidence

Figure 22

1. Repeat the previous parts (a-f) for m = 2, and m = 6. Compare the results of each part for different values of m.

**Answer:**

|  |  |  |
| --- | --- | --- |
| Executed | m = 2 | m = 6 |
| Question 2(a) | preferential\_network : Graph with 1050 nodes and 2096 edges  Is the network connected? True    Figure 23 | preferential\_network : Graph with 1050 nodes and 6264 edges  Is the network connected? True  Figure 24 |
| Question 2(b) | Modularity: 0.5226  Assortativity: -0.1383 | Modularity: 0.2491  Assortativity: -0.0437 |
| Question 2(c) | preferential\_network : Graph with 10500 nodes and 20996 edges    Figure 25  Modularity: 0.5319  Assortativity: -0.0416 | preferential\_network : Graph with 10500 nodes and 62964 edges    Figure 26  Modularity: 0.2492  Assortativity: -0.0254 |
| Question 2(d) | Figure 27  Estimated slope for n=1050: -2.089026582333908  Estimated slope for n=10500: -2.4037448293464063 | Figure 28  Estimated slope for n=1050: -1.6712718983559078  Estimated slope for n=10500: -2.0557312236994365 |
| Question 2(e) | Figure 29  Small network modularity:0.5210119420488316  Large network modularity:0.5347456750963387  Small network assortativity: -0.12129237046804335  Large network assortativity: -0.0503178768974231  Slope for small network (n=1050): -0.736687717812026  Slope for large network (n=10500): -0.9377194879885533 | Figure 30  Small network modularity: 0.2550699205172495  Large network modularity: 0.2503991606387869  Small network assortativity: -0.043919407206643675  Large network assortativity: -0.025030282664721425  Slope for small network (n=1050): -0.7453603129535443  Slope for large network (n=10500): -0.9908940820979608 |
| Question 2(f) | Figure 31 | Figure 32 |

1. Again, generate a preferential attachment network with n = 1050, m = 1. Take its degree sequence and create a new network with the same degree sequence, through stub-matching procedure. Plot both networks, mark communities on their plots, and measure their modularity. Compare the two procedures for creating random power-law networks.

Hint In case that fastgreedy community detection fails because of self-loops, you may use “walktrap” community detection.  
Useful function(s): sampledegseq

Answer:

Modularity of Preferential Attachment Network: 0.9271992664492308

Modularity of Stub Matching Network: 0.8309890087581124

A group of colorful dots

Description automatically generated

Figure 33

Preferential Attachment Network:

* Modularity: 0.9271992664492308
* Network Structure: Nodes tend to connect to already highly connected nodes, leading to a few nodes with very high degree (hubs) and many nodes with low degree.
* Community Structure: Likely to have a less uniform community distribution due to the presence of hubs dominating certain parts of the network.

A group of colorful dots

Description automatically generated

Figure 34

Stub Matching Network:

* Modularity: 0.8309890087581124
* Network Structure: Nodes are connected randomly based on available stubs (half-links), respecting the degree sequence.
* Community Structure: More evenly distributed communities as the connections are more randomized compared to preferential attachment.

Preferential attachment tends to create networks with a more pronounced hub-and-spoke structure, which can lead to higher modularity if hubs become central to communities. Stub matching, on the other hand, creates a more homogeneous network structure that might have less defined community boundaries, reflected in a lower modularity value.

The higher modularity in the preferential attachment network suggests that its community structure is more strongly defined, with clearer divisions between communities. This could be due to the centralizing effect of hubs. In contrast, the stub matching network’s lower modularity indicates a more mixed community structure, which might be beneficial for certain types of analyses where a more uniform network structure is desired.

When choosing between these methods for generating random power-law networks, the specific goals of the network analysis should guide the decision. If the aim is to study the effects of strong community structures, preferential attachment might be more appropriate. For a more uniform network that still respects the power-law degree distribution, stub matching would be suitable. Each method has its own implications for the network’s topology and the dynamics that can occur on it.

1. **Create a modified preferential attachment model that penalizes the age of a node**
2. Each time a new vertex is added, it creates m links to old vertices and the probability that an old vertex is cited depends on its degree (preferential attachment) and age. In particular, the probability that a newly added vertex connects to an old vertex is proportional to:

P [i] ∼ (ckiα + a)(dliβ + b),

where ki is the degree of vertex i in the current time step, and li is the age of vertex i. Produce such an undirected network with 1050 nodes and parameters m = 1, α=1,β=−1,and a=c=d=1,b=0. Plot the degree distribution. What is the power law exponent?

Hint Useful function(s): samplepaage

**Answer:**

A graph with a red line and blue line

Description automatically generated

Figure 35

The power-law exponent is a critical parameter in the power-law distribution that characterizes how the probabilities of the observed values fall off with their size. In the context of network analysis, it often describes the relationship between the frequency of nodes and their degree, indicating the likelihood of finding a node with a certain number of connections.

The estimated power-law exponent is: -2.8010618235704503

A network connection with blue dots and lines

Description automatically generated

Figure 36

1. Use fast greedy method to find the community structure. What is the modularity?

**Answer:**

The modularity of the community structure is: 0.9143707612043243

A network diagram of dots and lines

Description automatically generated

Figure 37

**Part 2 - Random Walk on Networks**

1. Random walk on Erdo ̈s-R ́enyi networks
   1. Create an undirected random network with 900 nodes, and the probability p for drawing an edge between any pair of nodes equal to 0.015.

**Answer:**

A graph of a network

Description automatically generated with medium confidence

Figure 38

* 1. Let a random walker start from a randomly selected node (no teleportation). We use t to denote the number of steps that the walker has taken. Measure the average distance (defined as the shortest path length) ⟨s(t)⟩ of the walker from his starting point at step t. Also, measure the variance σ2(t) = ⟨(s(t) − ⟨s(t)⟩)2⟩ of this distance. Plot ⟨s(t)⟩ v.s. t and σ2(t) v.s. t. Here, the average ⟨·⟩ is over random choices of the starting nodes.

**Answer:**

We have configured number of step as “1000” and generated the following graph based on that configuration.

A bar code graph with numbers

Description automatically generated

Figure 39

The above graph shows the behavior of a random walker on an Erdős-Rényi network over time. The horizontal axis represents the number of steps ( t ) taken by the walker, while the vertical axis represents two quantities: the average distance from the starting point and the variance of this distance.

The line representing appears relatively constant, hovering around a value close to 2.5. This suggests that, on average, the distance of the random walker from the starting node doesn’t increase significantly as more steps are taken.

The variance seems like had more fluctuation with several peaks and troughs. The variance measures how much the distances at each step vary from the average distance. The oscillation of the variance line suggests that there are times when the walker’s distance from the starting point is quite predictable (when the variance is low) and other times when this distance is highly unpredictable (when the variance is high).

A graph with blue squares

Description automatically generated

Figure 40

* 1. Measure the degree distribution of the nodes reached at the end of the random walk. How does it compare to the degree distribution of graph?

**Answer:**

A graph of a degree

Description automatically generated

Figure 41

The histogram that is provided shows the degree distribution of nodes reached at the end of the random walk. It appears to have a bell-shaped distribution with the highest frequency around degrees 10 to 15. This suggests that most nodes reached during the random walk have a degree within this range.

If the original graph’s degree distribution is similar in these aspects, then the nodes reached by the random walk are representative of the overall network. However, if there are significant differences, it might indicate that the random walk is biased towards certain nodes or that the network has a more complex structure.

In network theory, it’s common for large networks to follow a power-law degree distribution, where most nodes have a small degree, and a few nodes have a very high degree. If the original graph follows this pattern, but the random walk’s degree distribution does not, it could mean that the random walk is not sampling the high-degree nodes as frequently as expected.

For a more precise comparison, we have plotted the degree distribution of the entire original graph and overlay it with the distribution from the random walk. The graph is given in below.

A graph of a degree

Description automatically generated

Figure 42

* 1. Repeat 1(b) for undirected random networks with 9000 nodes. Compare the results and explain qualitatively. Does the diameter of the network play a role?

**Answer:**

A graph of a number of nodes

Description automatically generated

Figure 43

For a more precise comparison, we have plotted the degree distribution of the entire original graph and overlay it with the distribution from the random walk. The graph is given in below.

A graph with a number of lines

Description automatically generated with medium confidence

Figure 44

1. Random walk on networks with fat-tailed degree distribution
   1. Generate an undirected preferential attachment network with 900 nodes, where each new node attaches to m = 1 old nodes.

**Answer:**

A network of blue dots and lines

Description automatically generated

Figure 45

* 1. Let a random walker start from a randomly selected node. Measure and plot ⟨s(t)⟩ v.s. t and σ2(t) v.s. t.

**Answer:** Average distance for 900 nodes: 5.0200000000

Variance for 900 nodes: 1.6156000000

A graph with a line

Description automatically generated

Figure 46

A graph with a line

Description automatically generated

Figure 47

* 1. Measure the degree distribution of the nodes reached at the end of the random walk on this network. How does it compare with the degree distribution of the graph?

**Answer:**

Following graph has been generated from the simulated random walk on the preferential attachment network and number of walks was configured as 1000.

A graph of a degree distribution

Description automatically generated

Figure 48

Following graph has been generated from the simulated random walk on the preferential attachment network and number of walks was configured as 90000.

A graph of a degree distribution

Description automatically generated

Figure 49

* 1. Repeat 2(b) for preferential attachment networks with 90 and 9000 nodes, and m = 1. Compare the results and explain qualitatively. Does the diameter of the network play a role?

**Answer:**

Average distance for 90 nodes: 5.5860000000

Variance for 90 nodes: 6.4446040000

**A graph with a line

Description automatically generated**

Figure 50

**A graph with a line

Description automatically generated**

Figure 51

Average distance for 9000 nodes: 6.3780000000

Variance for 9000 nodes: 2.2451160000

**A graph of a graph

Description automatically generated**

Figure 52

**A graph showing a line

Description automatically generated**

Figure 53

**Qualitative Comparison and Network Diameter**

* The degree distribution in the large network (9000 nodes) may differ from the smaller network (90 nodes) due to the increased size and connectivity.
* The diameter of the network (maximum shortest path length) could play a role. Larger networks tend to have longer diameters, affecting how random walks explore the graph.

1. PageRank

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

* 1. We are going to create a directed random network with 900 nodes, using the preferen- tial attachment model. Note that in a directed preferential attachment network, the out-degree of every node is m, while the in-degrees follow a power law distribution. One problem of performing random walk in such a network is that, the very first node will have no outbounding edges, and be a “black hole” which a random walker can never “escape” from. To address that, let’s generate another 900-node random network with preferential attachment model, and merge the two networks by adding the edges of the second graph to the first graph with a shuffling of the indices of the nodes. For example,

A diagram of a number

Description automatically generated

Create such a network using m = 4. Measure the probability that the walker visits each node. Is this probability related to the degree of the nodes?  
Hint Useful function(s): asedgelist , sample , permute , addedges

**Answer:**

As per the instruction provided in questions, we have created two directed preferential attachment graphs with parameters (m=4 and n=900) and then shuffle the node indices of the second graph. After the shuffling the second graph, we have merged the two networks. The Merged Preferential Attachment Network has been plotted in below.

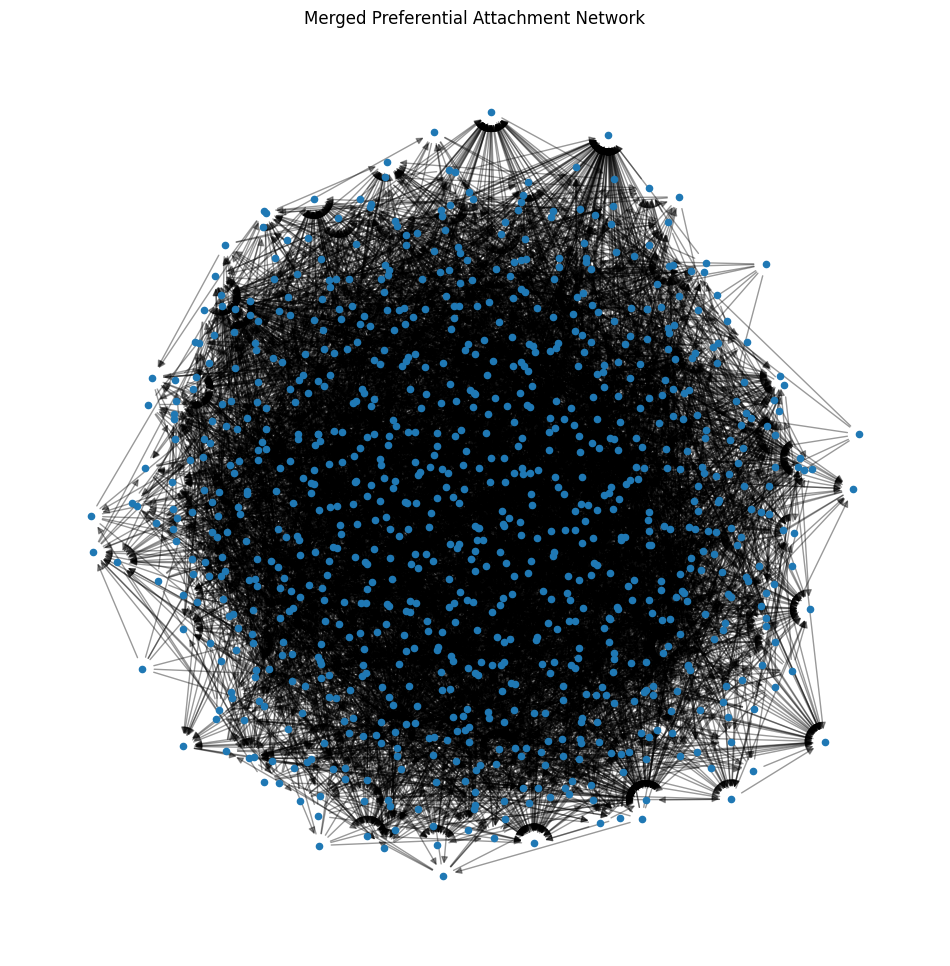


Figure 54

Then, as per the instruction, we have performed a random walk by choosing the next node randomly from the list of successor nodes from the graph. Here we didn’t use teleportation or pagerank algorithms. Then calculated the visit probabilities and plotted the visited probabilities against node degrees in below.

Here is the calculated correlation between node in-degree and visit probability: 0.8773 and based on this value, we can conclude that there is a significant relationship between node degree and visit probability.

A graph with blue dots

Description automatically generated

Figure 55

Then, we have calculated the pagerank socre and performed a random walk based on the pagerank calculated for the graph. Here we didn’t use teleportation algorithm. Then calculated the visit probabilities and plotted the visited probabilities against node degrees in below.

Here is the calculated correlation between node in-degree and visit probability: 0. 0.3755 and based on this value, we can conclude that there is no significant relationship between node degree and visit probability.

A graph with blue dots

Description automatically generated

Figure 56

* 1. In all previous questions, we didn’t have any teleportation. Now, we use a teleportation probability of α = 0.2 (teleport out of a node with prob=0.2 instead of going to its neighbor). By performing random walks on the network created in 3(a), measure the probability that the walker visits each node. How is this probability related to the degree of the node and α ?

**Answer:**

As directed in the problem, we have performed a random walk using given teleportation probability which is 0.2. Here we didn’t use pagerank algorithm. Then calculated the visit probabilities and plotted the visited probabilities against node degrees in below.

Here is my general understanding – The visit probability is expected to be less dependent on the node’s degree due to the teleportation probability α. With a higher α, the random walker is more likely to teleport to any node, making the visit probabilities more uniform across the network. Conversely, with a lower α, the walker’s path is more influenced by the network’s structure, and nodes with higher degrees would be visited more often. Based on the calculated correlation coefficient, probability related to the degree of the node and α is high.

However, I have noticed that calculated correlation between node in-degree and visit probability with teleportation is coming higher when I am configuring the higher number steps like 500 or more. For my test, I have configured number of random as 100000 steps.

Here is the calculated correlation between node in-degree and visit probability with teleportation: 0.9190004656.

A graph of a graph

Description automatically generated

Figure 57

A graph with blue dots and a red line

Description automatically generated

Figure 58

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.

* 1. Suppose you have your own notion of importance. Your interest in 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 random walk on network generated in question 3 to simulate this personalized PageRank. Here the teleportation probability to each node is proportional to its PageRank (as opposed to the regular PageRank, where at teleportation, the chance of visiting all nodes are the same and equal to N1 ). Again, let the teleportation probability be equal to α = 0.2. Compare the results with 3(a).

**Answer:**

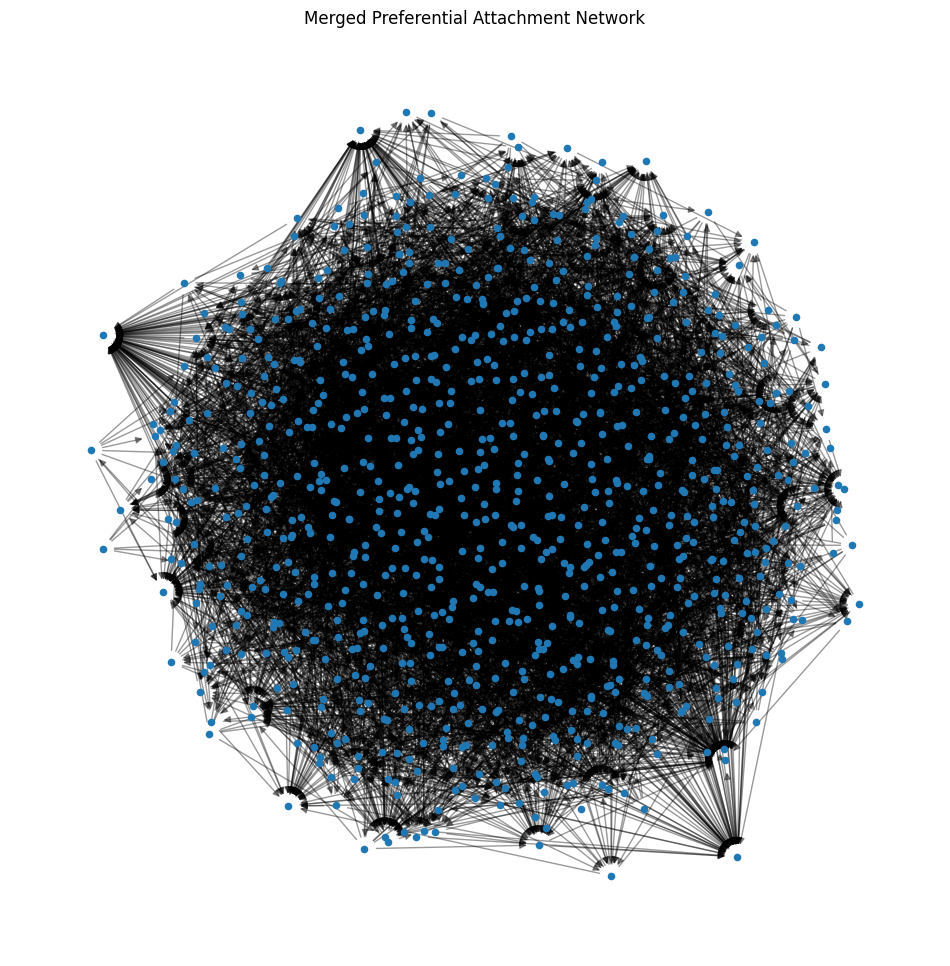


Figure 59

A graph of blue lines

Description automatically generated

Figure 60

Modularity: -0.0010624029373870636

Assortativity: -0.09443916583562009

A graph with a red line and a dotted line

Description automatically generated

Figure 61

A graph with numbers and dots

Description automatically generated

Figure 62

Correlation between node in-degree and visit probability (Personalized): 0.3091

There is no significant relationship between node degree and visit probability (Personalized).

* 1. Find two nodes in the network with median PageRanks. Repeat part 4(a) if teleportation land only on those two nodes (with probabilities 1/2, 1/2). How are the PageRank values affected?

Answer:

When teleportation lands only on the two median nodes, the PageRank values are affected in such a way that these two nodes will have a higher PageRank compared to the rest of the network. This is because the teleportation factor in the PageRank algorithm now favors these two nodes exclusively.

The scatter plot will show a deviation from the reference line for the two median nodes, indicating their increased PageRank values. The rest of the nodes will likely follow the reference line closely, assuming the teleportation probability is not too high to significantly affect their PageRank values.

A graph with blue dots

Description automatically generated

Figure 63

* 1. More or less, 4(b) is what happens in the real world, in that a user browsing the web only teleports to a set of trusted web pages. However, this is against the 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-reinforcement and adjust the PageRank equation?

Answer:

To adjust the PageRank equation to account for self-reinforcement, where users are more likely to teleport to a set of trusted web pages rather than any page uniformly, the personalized PageRank function has been modified to adjust the teleportation part of the PageRank calculation to reflect the preference for certain nodes. In this updated function, teleport\_prob is a dictionary that assigns a probability of 0.5 to the two median nodes and 0 to all other nodes. This reflects the idea that the random surfer is twice as likely to teleport to one of these two nodes than to any other node in the network.

Self-reinforcement increases the PageRank values of the preferred nodes (in this case, the two median nodes) because the teleportation part of the algorithm is biased towards them.

It can create a stronger bias towards certain nodes, making them more central or authoritative within the network, which may not accurately reflect the natural importance of nodes in the network.

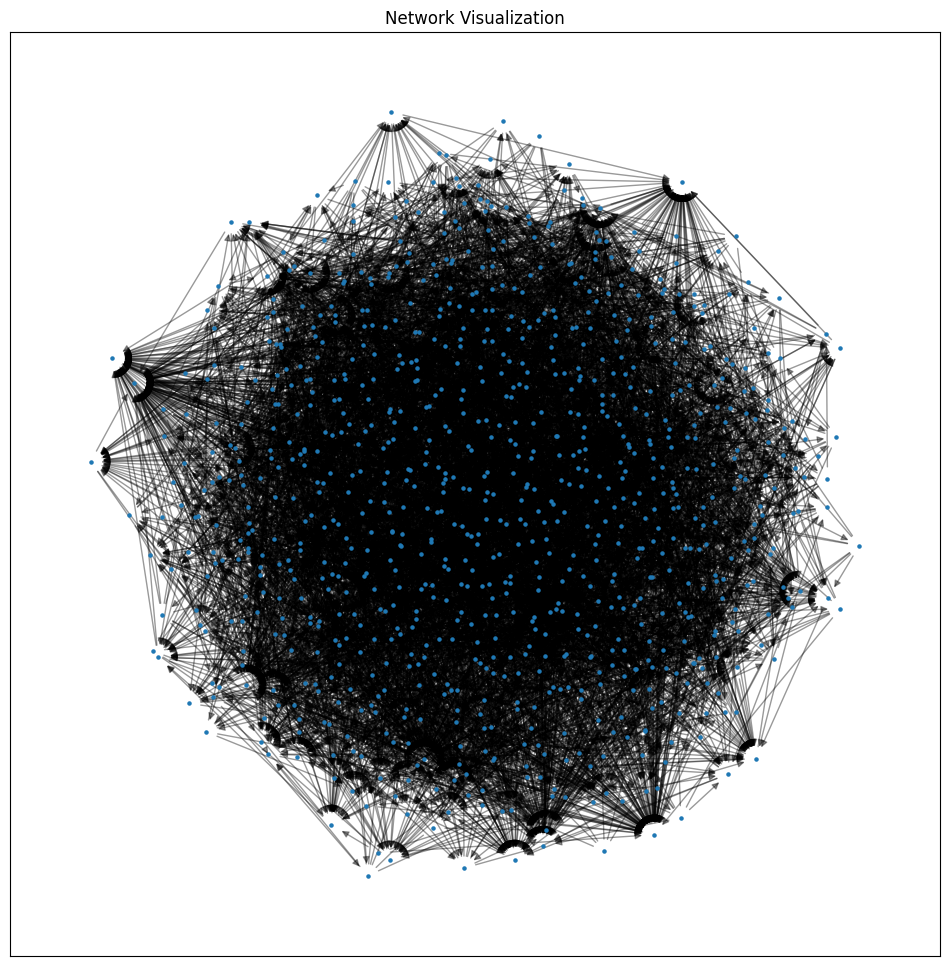


Figure 64

A graph with numbers and a number on it

Description automatically generated

Figure 65

A graph with numbers and a number on it

Description automatically generated

Figure 66

A graph with blue dots

Description automatically generated

Figure 67



Figure 68