Network Science

Rubric Assignment 1

Code for the following tasks. Do not use any existing network analysis libraries (such as NetworkX/igraph). All results must be submitted through a well-documented Jupyter Notebook. **Please submit a zip folder containing the data, Jupyter Notebook, and the PNGs of the graph where required. Use the filename convention: GroupNo.1_Assignment1.**

---

1. Identify an **undirected network** of your choice such that N > 1000. **Write a Python script** to:[15]

Submit a jupyter notebook named as Q1 and write the code for all parts a to g.

(a) Represent the network in terms of its 'adjacency matrix' as well as 'edge list'.

Write the code to represent the network-

- Adjacency matrix (1 mark)
- Edge List (1 mark)

(b) Visualize the network and export it as an image (PNG/JPG etc.).

Visualize the network using Cytoscape. Save the image as png. (1 mark)

(c) Computer the 'sparseness' of the network.

Explain the sparseness of the network in 2-3 lines. (1 mark)

(d) Compute its average degree < k >.

Write code to calculate the average degree of network. (1 mark)

(e) Plot its 'scaled/normalized degree distribution', $p_k \times k$

Plot the scaled degree distribution. Label the axis properly. Save and submit the image as png (3 mark)

(f) Compute its Average Path Length (Implement Breadth First Search Algorithm) and plot average path length distribution

- Write code to calculate the average path length using BFS algorithm. (3 marks)
- Plot average path length distribution. Label the axis properly. Save and submit the image as png. (1 mark)

(g) Average Clustering Coefficient and plot it as C(k) × k.

- Code to calculate the average clustering coefficient of the network. (2 marks)
- Plot as required. Label the axis properly. Save and submit the image as png. (1 mark)

2. **Write Python script** for computing in/out-degree for directed graphs. For a **real-world directed network** of N > 1000, **compute and plot** its in- and out- degree distribution. **[3]**

Submit a jupyter notebook named as Q2

- Write the code to calculate the in and out degree of directed graphs. (2 mark)
- Plot the in- and out- degree distribution. Save and submit the image as png (1 mark)

3. How would redefine the notion of 'degree' and 'clustering coefficient' for a weighted network to account for the edge weights? **Implement a Python script** to compute these and, for any relevant **real-world, weighted** graph, plot (a) 'weighted degree distribution' and (b) 'Clustering Coefficient' versus 'Degree'. **[6]**

Submit a jupyter notebook named as Q3

- Define the degree and clustering coefficient for weighted network (1 mark)
- Code to calculate the degree and clustering coefficient for weighted network (2 marks)
- Plot weighted degree distribution. Save and submit the image as png (1.5 mark)
- Plot clustering coefficient v/s degree of network. Save and submit the image as png (1.5 mark)

4. **Write a Python script** to create a Gilbert random graph corresponding to an undirected and unweighted real-world network that was used in Question 1. **Plot and compare** their 'scaled/normalized degree distributions'. Compute the degree distribution of the random graph over 100 instances. **[3]**

Submit a jupyter notebook named as Q4

- Code to create a Gilbert random graph for undirected and unweighted network. (1 marks)
- Plot the degree distributions. Save and submit the image as png. Compare them. (1 mark)
- Implement Gilbert random graph algorithm for 100 instances and Plot the degree distribution for 100 instances. (1 mark) Save and submit the image as png.

5. Load the real-world networks studied in the above examples in Cytoscape and visualize them using various layouts. Export the images in PNG/JPG or similar format. **[3]**

Visualize the network using various layouts. Save and submit the images. (at least 3; 1 mark per layout)