E536: Graph Analytics
Homework 1
Release date: Tuesday, 23 Jan 2024.
Due date: Monday, 29 Jan 2024. 11:59pm Eastern time

Submission instruction

What to submit: You need to submit your code (Jupyter notebook). Please make sure your notebooks are executed with saved output cells. For problem 2, you can write answers to a notebook cell. Please see instruction for submitting answer to Problem 3 and 4. You can submit an additional pdf file to show answers. Create a zip file with the contents (if you have multiple files), name it as "E536_LastName_FirstName_HomeworkName.zip" and submit on Canvas. If your answers are included in a notebook, please submit that notebook as "E536_LastName_FirstName_HomeworkName.ipynb" and submit on Canvas. For example, if your name is John Doe and you are submitting HW1, name the zip file as"E536_Doe_John_HW1.zip" or "E536_Doe_John_HW1.ipynb".

You are welcome to discuss HW questions with others. Please use Q&A Community on Canvas for discussions. We are always happy to help you. **Please don't copy answers or code from the internet or from another person. We will use a plagiarism checker. If we find evidence of plagiarism, you will get zero in HW.**

Associate Inspector: The AI responsible for this homework is Ryan Roy. Please visit her office hour with questions.

Late policy: For every late day you will lose 20% of the assigned grade for this HW. That means, if you submit 5 days after the deadline, you will not get any point. However, **We gill give you necessary extensions in case of any personal or health emergencies.**

Problem 1: Local clustering coefficient vs degree (20 points)

1. Let $C(v)$ be the local clustering coefficient of a vertex v. Also assume that V_k is a subset of vertices with degree k. Then, $C_k = \frac{1}{|V_k|} \sum_{v \in V_k} C(v)$ is the average clustering coefficient of all vertices with degree k. Write a Python function to compute C_k for all possible values of k. (hint: you can use clustering() and degree() functions from NetworkX). [5 points]
2. What is the computational complexity of your function? Express in terms of graph parameters (number of vertices n, number of edges m and degree). [5 points]
3. Use your Python code to group local clustering coefficient by degree for the Web-Google network (posted on Canvas). Draw a scatter plot with x-axis denoting degree of vertices and y-axis denoting the average clustering coefficient for a particular degree. Please explain how this plot is different from the clustering coefficient distribution plot shown in the class for the webGoogle hyperlink network. [7+3 points]

Problem 2: Erdős-Rényi vs social networks (10 points)

An Erdős-Rényi (ER) random graph $G(n, p)$ has n vertices, and each edge is added with probability p. In an ER graph, degree distribution follows Binomial distribution.

1. What is the average degree of $G(n, p)$? Express the average degree with n and p. [3 points]
2. If we keep p constant, how does the average degree change as the size of the network (n) increases? Do you expect the same behavior in a social network?. [5+2 points]

Problem 3: Characteristics of real networks (25 points)

Take **two networks** from the SNAP repository (http://snap.stanford.edu/data/index.html). Select these two networks from two different categories. For example, you may select a social network and a road network. Select networks with at least 10,000 vertices, but don't select a graph with more than one million vertices.

Use NetworkX to do the following tasks (we provided sample codes for these tasks):

1. For every network, plot the distributions of (a) degrees, (b) connected components sizes, (c) local clustering coefficients and (d) shortest path lengths. When computing shortest paths, only use the largest connected component. If computing shortest paths is expensive, use random sampling (that is use 10 randomly-selected vertices and compute their shortest paths). We already provided an example on how to use shortest path sampling. [10 points]

2. For every network, compute (a) average degree, (b) size of the large connected component, (c) average local clustering coefficient, and (d) average shortest paths lengths and (e) the diameter of the graph. (use sampling as needed). [12 points]

3. Report the time needed to compute the average statistics mentioned above. You can use the following Python function for timing. [3 points]

```
import time
start = time.time()
#your code
end = time.time()
print(end - start)
```

Problem 4: Characteristics of random networks (25 points)

Create four random networks following (a) Erdős-Rényi model, (b) scale-free model, (c) small-world model, and (d) Holme and Kim algorithm (not discussed in the class). Each network must have at least 10,000 vertices and 50,000 edges. Please select the parameter of the function so that you satisfy the minimum requirements. Use the following functions from NetworkX to generate these graphs:

Model	NetworkX function	Examples with possible parameters
Erdős-Rényi	gnm_random_graph	nx.gnm_random_graph(10000, 50000)
Scale-free	barabasi_albert_graph	nx.barabasi_albert_graph(10000, 5)
Small-world	watts_strogatz_graph	nx.watts_strogatz_graph(10000, 10, .1)
Powerlaw and clustered (Holme and Kim)	powerlaw_cluster_graph	nx.powerlaw_cluster_graph(10000, 5, .8)

Table 1: NetworkX functions for generating random graphs.

Similar to problem 2, use NetworkX to do the following tasks:

1. For every network, plot the distributions of (a) degrees, (b) connected components sizes, (c) local clustering coefficients and (d) shortest path lengths. When computing shortest paths, only use the largest connected component. Use sampling-based single-source shortest path if the connected component is too large. [10 points]

2. For every network, compute (a) average degree, (b) size of the large connected component, (c) average local clustering coefficient, and (d) average shortest paths lengths and (e) the diameter of the graph. (use sampling as needed). [12 points]

3. Report the time needed to compute the average statistics mentioned above. [3 points]

Reporting format for Problem 3 and 4

When solving problem 3 and 4, you will create six graphs (2 real and 4 random). When you report numerical data, please round the floating point numbers to 5 decimal places.

1. For each graph, you will create four plots. Hence, you will report 24 plots. Put plots of the same type from six networks together in one page in a 2x3 or 3x2 grid. That means, degree distribution plots from all graphs should come first, followed by another distribution plots from all six graphs and so on.

2. Create a table for the summary statistics computed in 3(b) and 4(b). Each row of the table stores properties for one graph. Each column stores one property for all graphs. The table shoud look like the following:

Graph	n	m	avg. degree	size of the largest component	avg. local clustering coeff	avg. path lengths	diameter
G1	-	-	-	-	-	-	-
G2	-	-	-	-	-	-	-
G3	-	-	-	-	-	-	-
G4	-	-	-	-	-	-	-
G5	-	-	-	-	-	-	-
G6	-	-	-	-	-	-	-

Table 2: Properties of real and random networks.

3. Based on the above plots and table, what is the best model for each of the real network selected? Why?

4. Create a table similar to Table 2 with the run time of the operations.