MCS-381 Social Computing (Fall 2023) Coding Assignment 1: Using the Preferential Attachment Model to Generate a Scale-free network

Start: Thursday, 9/14/2023
Due: Tuesday, 10/10/2023 by the start of the class (8am)
10% grade deduction penalty per day late.

First, please note that all coding assignments from this course are left to be somewhat open-ended, much like the work you are going to encounter after you graduate. I will give you the objective of the coding assignment and what I expect as the outcome. You are to provide a report documenting your process of solving the problem, as well as give me the outcome. What I won't specify, is the methods and approaches you should use in your implementation (or even the languages you should use), you will look those up yourself (and document it in your report).

Objective:

The power law is a mathematical concept that describe the distribution of events or quantities according to the scale-invariant relationship between the frequency of an event and its magnitude. They are commonly used in many different fields, including physics, economics, and sociology, to describe the distribution of wealth, the frequency of natural disasters, and many other phenomena.

With regards to the development of the web, power laws have been used to describe the distribution of links among websites, the popularity of websites, and the growth of online communities. The idea behind this is that the web is a complex system that exhibits many of the characteristics of a power-law distribution, including the presence of a few highly connected or popular nodes and a long tail of less connected or less popular nodes.

For a description of the power law, you should read (ahead on) Chapter 18.1 and 18.2 of your textbook. Another useful description of the power law is here: https://www.science.org/doi/10.1126/science.287.5461.2115a. I will also give you a description of the power law distributions in networks during the briefing of this assignment.

A network with degree distributions (in-degree and/or out-degree for directed graphs. In the case of our coding assignment, we're generating an un-directed graph, so it's just degree) that follow the power law is called a scale-free network. The most popular model based on preferential attachment described by Barabasi and Albert (1999). In this model, a new node is created at each time step and connected to existing nodes according to the "preferential attachment" principle. At a given time step, the probability *p* of creating an edge between an existing node *u* and the new node is `p=[(degree(u)+1)/(|E|+|V|)],` where *V* is a set of nodes and *E* is the set of edges between nodes (the probability is basically saying that the higher a node's degree is the more likely it is to attract new attachments, a "rich get richer" effect). The algorithm starts with some parameters such as the number of steps that the algorithm will iterate, the number of nodes that the graph should start with, and the number of edges that should be attached from the new node to preexisting nodes at each time step. The Barabasi model of network formation produces a *scale-free network*, a network where the node degree distribution follows a power-law principle. Scale-free networks produce small number of components, small-diameter, heavy-tailed distribution, and low clustering.

The preferential attachment model captures the "rich get richer" principle that could be the underlying mechanism behind the formation of most social networks and the world-wide web.

Here is a simulation of the preferential attachment model and the scale-free network it generates: https://ccl.northwestern.edu/netlogo/models/PreferentialAttachment, I will give you a more thorough introduction on the importance of the preferential attachment model during the briefing of this assignment (so do not miss these lectures).

Deliverables:

For this coding assignment, you, and your partner (in a team of two) are to implement the preferential attachment model. The easiest method to implement this is to use a direction (in Java or in Python) where the key is the node ID, and the value is the number of edges the node has so far. However, I will leave it up to you to decide on your method of implementation.

Use your implementation of the preferential attachment model to generate a graph of 2000, 4000, and 6000 nodes. For each graph, plot the log-log plot of the degree distribution and include it in a report. Your implementation of the preferential attachment model needs to be able to generate large graphs, so please make sure that your implementation is fast and efficient. For each of the log-log plot, please justify in your report if it follows the power law. If your graph follows the power law, give the slop of the power.

Submit your implementation via Moodle as well as a report documenting your implementation process and result (the three graphs). You should also give a brief analysis of each of the three graphs.

The correctness of your implementation if worth 60% while your report is worth 40%. In your report, your justification of the three graphs is worth 10% for each graph while the other components (introductions, conclusion, and other explanations) are worth 10%.