CSCI-4974/6971: Homework 1
<v1.0> updated January 24, 2024

Graph Connectivity

Due Date: Friday 9 February 2024, 11:59pm via Submitty

For this assignment, we're going to analyze the connectivity and properties of several graphs. For background material and reference, use:

- https://en.wikipedia.org/wiki/Strongly_connected_component
- https://en.wikipedia.org/wiki/Power_law#Maximum_likelihood
- https://en.wikipedia.org/wiki/Gini_coefficient#Definition
- https://www.cs.cornell.edu/home/kleinber/networks-book/networks-book.pdf – Ch. 13 - discussing the web structure

We will be using Submitty for autograding. Upload a single *.py file that outputs responses for all of the below. There will be separate gradeables listed for the 4974/6971 sections. Pay careful attention to output formatting. **You can use any NetworkX functionality you wish, but do not use any other external libraries**.

First, if you haven't done so already, download and install Python, NetworkX, and any other dependencies you might need. Work through the examples we've done in class to get a feel for working in the Python+NetworkX environment. Next, we'll be using a couple datasets for this analysis, so download the following:

- Simple Wiki: http://cs.rpi.edu/~slotag/classes/SP24m/hw/out.link-dynamic-simplewiki.data
  - Set of hyperlinks from the *Simple English Wikipedia*
  - Vertices: pages, Edges: hyperlinks
- p2p-Gnutella: http://cs.rpi.edu/~slotag/classes/SP24m/hw/p2p-Gnutella31.data
  - Crawler snapshot of the Gnutella social network
  - Vertices: peers, Edges: connections between peers

1. **Bow-tie Structure:** For the first two graphs (Simple Wiki and p2p-Gnutella), let's first see how their structure compares to the "bow-tie" structure observed on the Internet. For Simple Wiki, you should ignore the timestamp data on the edges. Compute the following quantities for each graph (see pg. 389 from EK) and output as is given in the template code file:

   (a) Number of weakly connected components.

   (b) Number of strongly connected components.

   (c) Number of trivial strongly connected components.

   (d) Number of vertices in each of SCC, IN, and OUT.

   (e) Number of vertices in each of Tendrils and Tubes.

   (f) Number Tendrils and number of Tubes.

   (g) **CSCI-6964 only:** As covered in class, "trivial components" are components consisting of a single vertex. Consider the notion of "trivial Tubes" and "Trivial Tendrils", which we'll define as Tubes and Tendrils that are composed only of a directed path on single vertices, each with out and in degrees of only 1. E.g., Tendril ($IN \to A \to B \to C \to D$) is trivial, but it would not be if edge ($A \to D$) or edges ($A \to E, E \to D$) also exists. Implement an approach to find Trivial Tendrils and Trivial Tubes, and output how many of each that these networks contain.

   We'll use the following definitions for each structural aspect:

   - SCC – Vertices in the largest SCC.
   - IN – Vertices that can reach SCC.
   - OUT – Vertices that can be reached from SCC.
   - Tubes – Vertices that can be reached from IN and can reach OUT, but are not in SCC.
   - Tendrils – Can reach or can be reached by vertices in the above sets, as well as other vertices that can reach or be reached by vertice in Tendrils.

   When determining the counts of tendrils and tubes, note that each tendril and each tube should be weakly connected. E.g., in some $G' = G \setminus \{IN, SCC, OUT, Tubes\}$, the number of weak components remaining from the original massive weak component would be the number of tendrils.

2. **Measurements:** Let's look at calculating various graph measurements to compare the various graph types we're considering, relative to the "real-world" graph properties we've discussed in class. As Simple Wiki and p2p Gnutella are directed, use the sum of the in+out degrees as a "degree" measure.

   (a) **Degree Skew:** Determine the skew of the degree distribution for each graph by

i. Estimating the power-law coefficient using the *maximum likelihood* method.

ii. Computing the Gini coefficient.

(b) **Hubs:** Let's define graph hubbedness (probably not a real word) based on the ratio of vertices that are in the tail of the degree distribution to the total number of vertices. The cutoff for a vertex v to be classified in the tail is if $d(v) > \ln |V|$.

(c) **Small-world:** Estimate the average shortest paths length. As this is computationally difficult to do exactly, use the following approximation scheme: uniformly select 100 vertices at random and run single source shortest paths from these vertices. The averaged average path length from these vertices will be the estimate of the average shortest path length. Only consider the largest component in each graph.

(d) **CSCI-6964 only:** Recall that calculating a graph's diameter is also deceptively difficult. Implement the approximation scheme we discussed in class, where you'll iterate 100 times in the largest component.