CSCI-4974/6971: Homework 3
<v1.0> updated March 31, 2024

Community Detection and Null Graph Models

Due Date: Friday 12 April 2024, 11:59pm via Submitty

For this assignment, we're going to do what is called an LFR-style community detection benchmarking experiment, as well as some subgraph frequency analysis using random graph models. We will be using the following datasets as a baseline for these experiments:

- Edits of .ng Wikipedia: `http://cs.rpi.edu/~slotag/classes/SP24m/hw/edit-ngwiki.data`
- Dolphin Interaction: `http://cs.rpi.edu/~slotag/classes/SP24m/hw/dolphins.data`

We will be using Submitty for collecting homeworks. 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. Your code should be runable as a script on the command line (via `bash$ python3 hw03.py`). **You can use any NetworkX/NumPy/SciPy functionality you wish, but do not use any other external libraries**.

1. We will first be performing an LFR-style experiment to evaluate the differences in community detection performance between the Louvain algorithm and the Label Propagation algorithm. CSCI-4974 students will simply be using the NetworkX LFR graph generator, while CSCI-6971 students will be developing their own generator using Chung-Lu graphs. See the template code file for more details. A basic overview is below.

    - Read in the baseline dataset and extract the needed parameters.
    - For each various value of mu (recall: the fraction of edges in the community graph that are external to a community), you'll generate 10 random graphs for the experiment.
    - For each graph, run label propagation and louvain.
    - Compute the NMI values of these outputs versus the 'ground truth'.
    - Output these NMI values.

2. For the second part of the assigment, we will be doing a subgraph frequency analysis, comparing the subgraph frequencies in real data to a few different potential 'null model' random graphs. You'll read in the real graph and determine the number of vertices, number of edges, and the degree distribution. You will then use these values to generate 10 instances each of 4 random graphs:

   - Erdos-Renyi $G(n, m)$ graphs
   - Chung-Lu graphs
   - Barabasi-Albert graphs
   - Watts-Strogatz graphs with $p = 0.25$

   Next, you'll use generated instances of all possible 3- and 4-vertex connected subgraphs created from the methodology as given in class. For each network and subgraph, determine the average number of counts over each of the 10 generations for a network type. You'll output those averages for comparison of the 'null models' against the real network.