HW 3: due Thursday, October 6

For this assignment, submit your answers for both problems as a single pdf file named `hwk3.pdf`, alongside any code you may have written, using Gradescope. Submit your pdf to the assignment named "Homework 3 PDF". Submit your code and any generated networks to the assignment named "Homework 3 Networks and Code".

This assignment is far more coding-heavy than the previous assignments. Please come to office hours or reach out early if you have issues implementing the solutions to problem 1, as an understanding of how to traverse and explore digital representations networks will be critical for problem 2.

Problem 1

For this problem, you will need to download the yeast PPI physical interaction network, available as the PPI data set on the course site.

(http://www.cs.tufts.edu/comp/142/private/datasets.html)

This file includes 76025 yeast protein protein interactions (PPI) among 5001 proteins. The file is tab-delimited. Each row corresponds to one interaction and has three columns: the first and the second columns are the two interacting proteins, while the third entry is an number in (0, 1] that shows the confidence of the existence of the interaction.

(a) Construct a simple undirected, unweighted graph from this protein-protein interaction network. Each graph node should correspond to a portein in the protein interaction network, and an undirected edge should exist between two nodes if there is an interaction between the two proteins in the PPI dataset. Export this network as `p1a.csv`.

(b) Many computational biologists have concluded that PPI networks are scale-free. Compute the degree distribution for the 5001 nodes and add it as a histogram to your pdf.

(c) Compute the local clustering coefficient for each node in the graph. Submit a tab-delimited file named `p1c.csv` containing two columns, where the first column is the protein ID and the second column is the clustering coefficient.

Proteins YGR296W and YPL098C both have 5 interacting partners; which one has the higher clustering coefficient? Explain briefly what the difference means.

(d) Implement a function that counts the number of triangles, or 3-cliques, in the graph. A k-clique is defined as a graph with k nodes with an edge between every pair of nodes. In general, we consider that a clique forms a functional module in the PPI network; thus, it will be interesting to find all of them. How many 3-cliques are there in the PPI network? What is the global clustering coefficient of this network? Please do not use `networkx`'s `triangles` function for this question, although you may use it to check your work.

(e) In some applications, we seek to find "close" pairs of proteins based on the network and study the similarity between such "close" pairs. A simple way to define the "closeness" is to use shortest path distance.

In order to estimate the path length distribution for this graph, sample 1000 nodes at random from the graph, and compute the distribution of shortest path distances between these 1000 nodes. Please submit a histogram showing the distribution of shortest path distances you observe. How does this compare to your expectations for a protein-protein interaction network?

(f) Estimate the diameter of the network based on your answer to part e. How does what you found relate to the results of part d?

Problem 2

For this problem, you will be using the same PPI network from the previous question, but you will also use the MIPS annotations provided alongside the data set. These MIPS annotations are functional annotations – they describe the function of a protein. We have MIPS annotations for a subset of proteins in the PPI network, but not all of them, and you will implement an algorithm for predicting the function of un-annotated proteins.

(a) Implement the majority vote algorithm. Given a yeast protein, have each of its labeled neighbors vote once for each of their labels, and assign that protein

the single label that got the most votes (if there is a tie, break it numerically by MIPS number, assigning it to the lower-numbered MIPS term). Repeat this process until all nodes are annotated. Submit a comma-delimited file of protein-annotation pairs as p2a.csv . (You may either have one protein and one annotation per line, or one protein with all of its annotations in a single line.)

(b) Implement a different algorithm, or a variation of this algorithm to accomplish the same task. Explain carefully what you did, and why. Submit the final annotations in a comma-delimited file named p2b.csv.

(c) Measure the performance of both algorithms using *leave one out cross validation.*

Leave-one-out cross-validation is performed as follows: Take a single, labelled node (a protein you have MIPS annotations for) and temporarily remove its label. Perform your labelling algorithm. After completion, if the node you temporarily un-labelled was given a label it had originally (in the "truth" data), this test is considered a success. Now, repeat this for every node for which there are known labels (only removing one label per run!), and summarize the number of test runs that successfully recovered one of the original labels and the number of runs where this did not happen.

(d) In the previous question, you were asked to predict a single correct functional label, but some proteins have been assigned multiple labels. Say a few words about how you might extend the majority vote algorithm to allow the assignment of multiple labels. How might you measure performance in this setting?