CS550 **Problem Set 3**
Spring 2020 **Due:** Apr 18, 2020, 11:59 p.m.

This homework has a total of 70 points, it will be rescaled to 10 points eventually.

Submission instructions: These questions require thought but do not require long answers. Please be as concise as possible. You should submit your answers as a writeup in PDF format, for those questions that require coding, write your code for a question in a single source code file, and name the file as the question number (e.g., question_1.java or question_1.py). Finally, put your PDF answer file and all the code files in a folder named as your NetID (e.g., ab123), compress the folder as a zip file (e.g., ab123.zip), and submit the zip file via Sakai. For the answer writeup PDF file, we have provided both a word template and a latex template for you, after you finished the writing, save the file as a PDF file, and submit both the original file (word or latex) and the PDF file.

1. **Modularity [15pt]**

   In (Newman 2006, PNAS 103(23): 8577–8582)[1] Mark Newman defines the modularity of a network divided into two components as (see paper or course slides for specification on notation):

   $$Q = \frac{1}{4m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) s_i s_j \tag{1}$$

   We will now get a better intuition on what this quantity means. Consider the network in the figure 1 below:

   (a) [5pt] If we remove edge (A, G) and partition the graph into two communities, calculate the modularity of this partition.

   (b) [5pt] Now, consider the original network from the figure and the groups identified in (a). Add a link between nodes E and H and recalculate modularity Q. Did the modularity Q go up or down? Why?

   (c) [5pt] Consider the original network from the figure and the groups identified in (a). Now add a link between nodes F and A and recalculate modularity Q. Did Q go up or down? Why?

2. **Spectral Clustering [30pt]**

   Still consider the graph in Figure 1, assume that any edge in this graph has an equal weight 1. We run spectral clustering to partition the graph into two communities.

   (a) [10pt] Provide the adjacency matrix A, degree matrix D, and Laplacian matrix L of the graph.

[1] Newman ME. Modularity and community structure in networks. Proceedings of the national academy of sciences. 2006 Jun 6;103(23):8577-82.

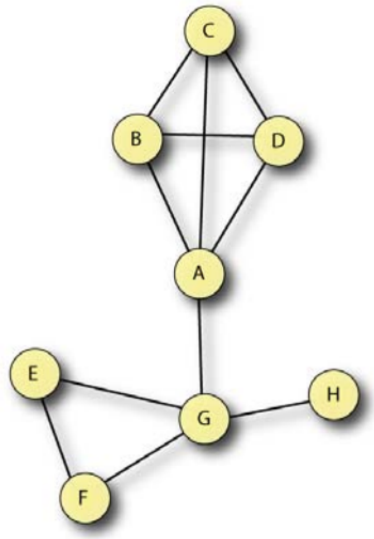

Figure 1: Figure of problem 1 and problem 2

(b) [10pt] Using Matlab or Python, compute the eigen values and the corresponding eigen vectors of the Laplacian matrix. Rank the eigen values in ascending order. [You may refer to the problem 1 in homework 2 for some hints of using Python to compute eigen values and eigen vectors]

(c) [10pt] What is the eigen vector corresponding to the second smallest eigen values? Using 0 as the boundary, partition the graph into two communities, what is the graph partitioning result?

What to submit:

i. The matrices A, D and L in (a).

ii. The eigen values and eigen vectors in (b), as well as the code for computing them.

iii. The graph partitioning result in (c).

3. **Clique-Based Communities [25pt]**

Imagine an undirected graph G with nodes $2, 3, 4, ..., 1000000$. (Note that there is no node 1.) There is an edge between nodes i and j if and only if i and j have a common factor other than 1. Put another way, the only edges that are missing are those between nodes that are relatively prime; e.g., there is no edge between 15 and 56.

We want to find communities by starting with a clique (not a bi-clique) and growing it by adding nodes. However, when we grow a clique, we want to keep the density of edges at 1; i.e., the set of nodes remains a clique at all times. A maximal clique is a clique for which it is impossible to add a node and still retain the property of being a clique; i.e., a clique C is maximal if every node not in C is missing an edge to at least one member of C.

(a) [5pt] Prove that if i is any integer greater than 1, then the set C_i of nodes of G that are divisible by i is a clique.

(b) [10pt] Under what circumstances is C_i a maximal clique? Prove that your conditions are both necessary and sufficient. (Trivial conditions, like "C_i is a maximal clique if and only if C_i is a maximal clique", will receive no credit.)

(c) [10pt] Prove that C_2 is the unique maximal clique. That is, it is larger than any other clique.