L3S Research Center
Large Scale Data Mining, SS 2016
Dr. Avishek Anand
Assignment 6, due: **14 June 2016**

You must return your assignment sheet and have a correct solution in order to present in the exercise groups. Please write legibily! Do no forget to put your name and matriculation number on your solution!

Problem 1.
Suppose we have a coin, which may not be a fair coin, and we flip it some number of times, seeing h heads and t tails.

1. If the probability p of getting a head on any flip is p, what is the MLE for p, in terms of h and t?
2. Suppose we are told that there is a 90% probability that the coin is fair (i.e., $p = 0.5$), and a 10% chance that $p = 0.1$. For what values of h and t is it more likely that the coin is fair?

Problem 2.

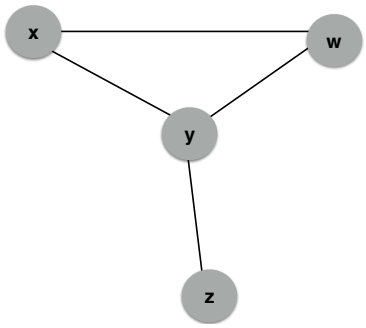

Figure 1: Community Graph

Compute the MLE for the graph in Figure 1 for the following guesses of the memberships of the two communities.

1. $C = \{w, x\}$; $C = \{y, z\}$.
2. $C = \{w, x, y, z\}$; $C = \{x, y, z\}$.

Problem 3. Suppose graphs are generated by picking a probability p and choosing each edge independently with probability p. For the graph of Figure 1 , what value of p gives the maximum likelihood of seeing that graph? What is the probability this graph is generated?

Problem 4. Compute the number of triangles and Clustering coefficient (for each node) of a

1. Complete graph (clique) with n vertices.
2. Complete bi-partite graph with left set with l and right set with m vertices.

Consider a node A in a graph G. A has exactly p neighbors with an edge probability between the neighbors being p. What is the expected value of the clustering coefficient for node A.

Problem 5. For the graph in Figure 2 determine:

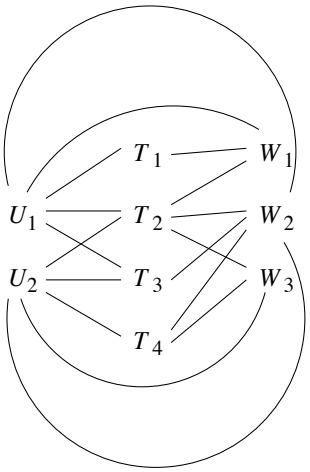

Figure 2: Tripartite Graph

1. What is the minimum degree for a node to be considered a heavy hitter?
2. Which nodes are heavy hitters?
3. Which triangles are heavy-hitter triangles?

Problem 6.

1. Extend the parallel algorithm discussed in the lecture to detect squares. That is for nodes a, b, c, d the edges $(a, b), (b, c), (c, d), (a, d)$ should exist in the graph. Write the pseudo-code map and reduce steps involved.
2. Does your proposed algorithm be extended for arbitrary sized polygons ?
3. Are there computational bottlenecks in the algorithm when there is skew (a power law distribution on the outdegrees) ? Outline rough ideas to overcome them (if at all).