CS4980:0005 PEER-TO-PEER AND SOCIAL NETWORKS
HOMEWORK 1
100 POINTS
ASSIGNED 9/14/15 DUE 9/23/15

Download NetworkX, a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. It will simplify your effort in solving the homework problems. This package has a library of various kinds of graphs as well as the implementation of various useful algorithms for social network analysis. Read the notes at the end.

You have to submit one zip file containing (1) a **readme** file describing your strategy for solving the problem, (2) directions to the grader about how to run your program with the proper parameters, and what outputs to expect (3) the codes, and (4) the output or the results as appropriate.

Problem 1 (50 points): The goal of this problem is to generate a social network using *preferential attachment*. Recall how such a graph is generated. Starting with a single node, in each step, add a node to the network using *a single* edge. The probability of choosing the node to which it will attach will be proportional to the degree of that node.

(a) Using this rule, generate three graphs $n = 1000, 2000, 5000$ nodes. Display at least one graph.

(b) What are the largest degrees in each of these graphs?

(c) Show the degree distribution of the generated graph. For this, plot $\log N(k)$ vs. $\log k$, where k is the degree of a node, and $N(k)$ is the number of nodes having degree k. Verify if the power law ($N(k) = c.k^{-r}$) holds, and estimate the value of r.

Problem 2 (50 points): Consider the *Karate Club* network discussed in the class (and also check the major sources, like Easley and Kleinberg's book or Girvan and Newman's paper). Your tasks are as follows:

(a) Compute the edge(s) of highest betweenness in this network.

(b) Remove the edge(s) of highest betweenness, and visualize the graph and see if it has been partitioned into disjoint communities. If not, then repeat the above two steps, until the graph partitions into two components, and then stop.

(c) List the edges that you removed. Display the components

Use the built-in features of NetworkX as much as possible.

Notes

1. How to install python

You can download and install python here: http://www.python.org/getit/

Do not install 3.0 or higher version. You may not be able to use matplotlib (but check this)

2. How to install IDE

If you need IDE for python, explore various options. A good text editor like notepad++, gedit will be enough for python programming

3. How to install networkx

See: http://networkx.github.com/documentation/latest/install.html

4. How to get started

A good resource: http://snap.stanford.edu/class/cs224w-2011/nx_tutorial/nx_tutorial.pdf,

Another good tutorial can be found in networkx's website:

http://networkx.github.io/documentation/latest/contents.html

5. How to generate graph

You may to look at: http://networkx.github.io/documentation/latest/reference/generators.html

When you draw using nx.draw, also call nx.show to open the windows so that you can see it

6. How to find algorithms

See http://networkx.github.io/documentation/latest/reference/algorithms.html