CCI 688-02, MATH 410-01: INTERNET ALGORITHMS & ECON

Homework 1

Spring 2016

Anke van Zuylen

Due date: Thursday February 4

1. The data set for this problem is a bibliography of computer science papers maintained by Joel Seiferas at the University of Rochester. The file collab_data.txt contains the information I downloaded from `ftp:\/\/ftp.cs.rochester.edu\/pub\/u\/joel\/papers.lst.`. Each line in the file contains a distinct paper.

   Your job is to create a co-authorship graph, which contains one node for each author, and an edge between two authors if they are co-authors on a paper in the file. The bibliography contains some authors that have the same name. For the purposes of this problem set, these simply get mapped to a single node, and you may therefore find that your graph will have self-loops.

   (EC) For an extra challenge, write a script that assigns a unique number to each author, and converts the file collab_data.txt to a file that has a line for each paper, where the line lists the numbers of the authors on the paper. NB: this really is a challenge: like any real data set, our data set has typo's, and inconsistencies in its formatting!

   If you do not feel up to this challenge, you can use the files created by Van Dimopoulos (one of the students who came before you) for the remaining questions:

   - The file authorlist.txt contains the authors that appear in the bibliography, numbered 1 to 34740.
   - In the files output.txt and output2.txt, each line again represents a distinct paper. File output2.txt lists for each paper both its authors and the numbers for the authors. Finally, output.txt contains for each paper the numbers for the authors on the paper (followed by 0's to make each line have the same number of entries – not needed but makes it easy to read into Matlab (if you should wish to do so)). So the line 1 5 12 0 0 0 0 0 0 0 would represent a paper written by authors 1, 5 and 12 (the zeros are there only to make each line have the same number of entries.

   (a) (**Creating the graph**) Create a simple undirected graph that has a node for every author (i.e., your network will have 34740 nodes) and an edge between a pair of authors, if they appear together on some paper in the bibliography. You should create only one edge if two authors appear together on multiple papers, i.e., your graph should be *simple* and not a multigraph. Report the number of edges (reporting the number of self-loops, and the number of edges between distinct nodes separately) in your graph.

(b) (**Node degrees**) The degree of a node is the number of edges it's incident to. We start by considering how the degrees of the nodes are distributed.

Thus, for a number j, let n_j denote the number of nodes with degree exactly j. Let $d_{\max}$ be the maximum degree of any node in the network. (This is the maximum total number of co-authors that any one author has – the maximum j for which $n_j > 0$.)

i. For each j from 0 to d, output the number n_j.

ii. Produce a scatterplot in the plane of the ordered pairs $(\log j, \log n_j)$ for those j such that both $j > 0$ and $n_j > 0$.

(c) (**Connected components**)

i. Let n^* be the number of nodes in the largest connected component, and let n be the number of nodes in graph overall. Report these two quantities and their ratio.

ii. Let k_j denote the number of connected components of size j. Report j and k_j for each j such that $k_j > 0$.

iii. Produce a scatterplot in the plane of the ordered pairs $(\log j, \log k_j)$ for those j such that $k_j > 0$.

(d) (**Node-to-Node Distances**) We now restrict ourselves to the largest connected component.

i. We first focus on the node for "Hartmanis" (node 3439). Report the maximum distance between Hartmanis and any other author in the largest connected component.

ii. Find the maximum distance between "Bornberger-Bauer" (node 7881) and any other author in the largest connected component.

iii. Give an upper and lower bound on the diameter of the largest connected component.

2. One of the goals of network analysis is to find mathematical models that characterize real world networks and that can then be used to generate new networks with similar properties. In this problem, we will explore two famous models– Erös-Rényi and Small World –and compare them to the real-world data from the previous part.

- $G(n, p)$ Random Network: To construct this undirected network, use $n = 34740$ nodes and pick $m = 111632$ edges at random.

  Make sure you do not create multi-edges, but you may allow self-loops, as we allowed these in our real-world graph as well. (How many self-loops do you create in expectation? How does this compare to what you found in 1(a)?)

- Small-World Network: You can generate this undirected graph as follows: Begin with $n = 34740$ nodes arranged as a ring, i.e., imagine the nodes form a circle and each node is connected to its two direct neighbors (e.g., node 399 is connected

to nodes 398 and 400), giving us 34740 edges. Next, connect each node to the neighbors of its neighbors (e.g., node 399 is also connected to nodes 397 and 401). This gives us another 34740 edges.

Finally, randomly select 42152 pairs of nodes not yet connected and add an edge between them. In total, this will make $m = 34740 \cdot 2 + 42152 = 111632$ edges.

(a) (**Degree Distribution**) Generate the above two networks. Plot and compare the log-log degree distribution of the two $G(n, p)$ Random Network, the Small-World Network and the Collaboriation Graph from 1(b). Superimpose the plots of the three networks. Briefly describe the shape of each distribution and explain the differences between the networks.

(b) (**Connected components**) Report the size of the largest connected component for the $G(n, p)$ Random Network and the Small-World Network.

(c) (**Node-to-Node Distances**) For each of the two networks, pick a node u in the largest connected component, and find the maximum distance between u and any other node v in the largest connected component.