Graphs and Networks **out:** September 16, 2010

Problem Set 1

Lecturer: Daniel A. Spielman **due:** September 30, 2010

1 Introduction

There are 3 sections to this problem set. The first contains problems for everyone. The second contains problems for those who are taking the “theoretical track”. The third contains experimental problems for those who are taking the “experimental track”. If you are considering ever taking the experimental track, I suggest you do it now. The reason is that you will build upon previous experiments.

2 Homework Policy

You may discuss the problems with other students. But, you must write your solutions independently, drawing on your own understanding. You should cite any sources that you use on the problem sets other than the textbook, TA and instructor. This means that you should list your collaborators.

You **may not** search the web for solutions to similar problems given out in other classes. If you think this policy needs any clarification, please let me know.

3 Clarifications

1. In problem 2, I mean that the edge should be chosen uniformly at random. That is, each edge with the same probability. The same goes for the endpoint.
2. In problem 4, you should show this for every constant g. That is, show that for every g there is an n_0 such that for all $n \geq n_0$ the statement holds.
3. In problem 3, I should have said that $c \geq 1$.
4. The point of getting meta-data about vertices and edges in the experimental graph is so that we can later define some property P that some vertices (or edges) have and some don't. We want this property to be non-trivial. It would not be interesting if only one vertex or one edge had the property, or if all of them did.

4 Problems for all

1. In Lecture 2, we saw that star graphs (or at least one star graph) have assortativity -1. Present a graph of assortativity -1 in which every vertex has degree at least 2. Justify your computation of the assortativity.

2. Let G be a graph in which the number of vertices of degree i is n_i. Consider picking a random edge in G and then a random endpoint of that edge. What is the probability that the random endpoint has degree d? Express this as a function of d and the n_is. Justify your computation.

5 Theoretical Problems

3. Consider the Galton-Watson branching process in which every cell divides into $k \geq 2$ parts, and in which every part survives to reproduce with probability $p = c/k$. Prove that there is an absolute constant $\beta_c > 0$, depending on c but independent of k, so that the probability the descendants of a cell exist forever is at least β_c.

4. In Lecture 3 we considered random graphs from the distribution $\mathcal{G}(n,p)$ with $p = n^{1/2g-1}$. We then removed a number of vertices from a graph chosen from this distribution. Some students asked whether we could show that the remaining graph still has a lot of edges. You will show that now.

   For a subset of the vertices S, we define $G(S)$ to be the graph having vertex set S and all edges of G that go between vertices of S. Prove that, with high probability for sufficiently large n, for *every* $S \subset V$ with $|S| = n/2$, $G(S)$ contains at least $1/5$ of the edges of G.

5. A 1-out is a directed graph in which the out-degree of every vertex is 1. We may choose a 1-out at random by choosing the endpoint of the edge leaving every vertex uniformly at random among the other vertices. Let G be a random 1-out on n vertices. For a vertex v, let $R(v)$ denote the set of vertices reachable by a directed path from v.

   a. Prove that for each vertex v, for sufficiently large n, $\Pr\left[|R(v)| < \sqrt{n}/10\right] < 1/3$.

   b. Prove that for each vertex v, for sufficiently large n, $\Pr\left[|R(v)| > 10\sqrt{n}\right] < 1/3$.

   Note that these constants are pretty loose. Much tighter bounds are possible.

6 Experiments

This problem can be summarized as "get a useful graph and tell me a little about it". To be more specific, you should find a graph that you are going to use in experimental work throughout the class (although you can switch graphs later if you become unhappy with your original choice). The graph should be from some "real-world" source. Moreover, there should be some information available about the vertices or edges. We will later use this information to assist in validation studies. Some examples of types of useful information are:

1. Dates or timestamps associated with edges or vertices. Maybe when they were created.
2. If vertices are movies, a list of the genres of each.
3. If vertices or edges are articles, the title, subject or classification of each.
4. If vertices are web pages, keywords or the text of the pages.

You really just want to identify some properties that some vertices or edges have and that some don't.

Ideally your graph should have at least 10,000 vertices. If your graph has fewer than 1,000 vertices, get enough graphs so that you have at least 1000 vertices total. We will need these so that we can get some statistical significance when we do experiments later in the course.

The best way to get a graph is probably to write a script to read it or grab it from the web. Interesting sources include databases of papers (arxiv, ncstrl), fragements of the web (wikipedia, links internal to a university, etc.). Wget is a useful tool for downloading a large portion of the web, but it is overkill since you only need the links. This part could take a lot of work. You can use any language that you like. However, I suggest using a language which has a large number of libraries available. I recommend Matlab, Java, Python, C, and maybe R. Your choice of language will have a big impact on the time your computations require.

Here's what you should report:

a. What is your graph, and where did you get it.

b. What additional information have you obtained for your graph.

c. How many nodes are there in your graph?

d. How many edges does it have?

e. What is its assortativity?

f. Compute the clustering coefficient $C^{(2)}$ of your graph. (that is, the average clustering coefficient over the vertices)

Submit all code you've used. I will also create a web form where you should submit those numbers.