Graphs and Networks **out:** November 10, 2010

Problem Set 4

Lecturer: Daniel A. Spielman **due:** November 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".

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.

If you are doing the experimental problems, it goes without saying that you should write your own code.

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/Mistakes Fixed

1. Fixed the direction of the inequality in problem 2b.
2. Fixed the cost function in problem 3.

4 Problems for all

1. Let G be a directed, strongly-connected, unweighted graph with n vertices (That is, every edge has weight 1). Let $\boldsymbol{\pi}$ be the stable distribution of the random walk on G. Prove that for every vertex u, $\boldsymbol{\pi}(u) \geq n^{-(n+1)}$.

5 Theoretical Problems

2. Consider a directed road network G in which every edge e has a cost function of the form $c_e(f_e) = a_e f_e$. That is, the cost of each edge is proportional to the amount of flow on that edge, with no constant term. Consider traffic flows between two distinguished vertices s and t.

   a. Prove that $\text{Nash}(G) = \text{Opt}(G)$.

   b. Let G' be a network obtained by *removing* an edge from G. Prove that $\text{Nash}(G') \geq \text{Nash}(G)$. That is, Braess's Paradox cannot happen in such networks.

3. Let G be an undirected, connected, weighted graph. Let the weight of edge e be w_e. We can view G as a resistor network in which the resistance on edge e is $r_e = 1/w_e$. We can also view G as a directed traffic network by replacing each undirected edge with two directed edges pointing in the oppposite directions. Make the cost functions of each of these edges be $c_e(f_e) = r_e f_e$.

   a. Establish an isomorphism between the unit electrical flow in the electrical network and the Nash flow in the traffic network.

   b. Use this isomorphism to prove Rayleigh's Monotonicity Principle.

   c. Now, consider an electrical network of resistors and ideal diodes (not Zener diodes). The ideal diodes direct edges: they will not allow any electrical flow in one direction, and they allow arbitrary flows in the other direction with no energy dissipation. The electrical flow between two nodes s and t will be the flow that obeys the directions given by the diodes and minimizes energy dissipation over the resistors. The effective resistance between s and t is still defined to be the energy dissipation of this electrical flow. Prove that if one cuts a resistor or diode in this network, the effective resistance between s and t cannot decrease.

6 Experiments

Your goal in this project is to try doing link prediction on your graph, following the methodology of Liben-Nowell and Kleinberg. The first thing you must decide is which links to use for training and which to use for evaluation. If your graph has time-stamps associated with edges, then break it up this way. If there is nothing as good as time-stamps, then just choose a random subset of edges for evaluation. I suggest choosing at least 1000 edges. If this would be more than one third the edges in your graph, then your graph is too small. Find another.

Your task is to reproduce these experiments, using just a small number of graph metrics. These should include:

1. graph distance

2. Number of common neighbors

3. Jaccard's coefficient
4. The Ademic/Adar measure.
5. The Preferrential attachment measure.
6. If your graph has fewer than 2000 vertices, you should also try the rooted PageRank measure. Note that you can do this efficiently by storing the appropriate matrix (which has 4 million entries, each requiring only 8 bytes).

Explain well how you did the experiments, and explain the results.

Print and submit all code. Make sure it is readable by a human. Add explanations where necesssary.