Graphs and Networks **out:** October 19, 2010

Problem Set 3

Lecturer: Daniel A. Spielman **due:** November 4, 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 Problems for all

1. Let $G = (V, E, w)$ be a weighted graph, with positive edges weights given by the function w. If we view this graph as a resistor network, then the resistance of edge e, r_e, is $1/w_e$.

   In class, we considered what happens if we fix the potentials (voltages) of some vertices. We observed that the voltages as the remaining nodes may be determined by solving a system of equations in the Laplacian matrix of the graph, subject to the given voltages. Your job is to write this as an ordinary system of linear equations in the unknown voltages.

   That is, let $W \subset V$ be the set of vertices at which voltages are fixed, and let them be fixed according to the function f. So, $\boldsymbol{v}(x) = \boldsymbol{f}(x)$ for all $x \in W$. Let $U = V - W$ be the set of vertices for which the voltages are unknown. You need to solve for $\boldsymbol{v}(x)$ for each $x \in U$.

   Let $\boldsymbol{u} \in \mathbb{R}^U$ be the vector of voltages for which you want to solve. That is $\boldsymbol{v}(x) = \boldsymbol{u}(x)$ for $x \in U$. This is a linear problem, and so it can be written in the form

$$M\boldsymbol{u} = \boldsymbol{b},$$

where M is a matrix with rows and columns indexed by U and $\boldsymbol{b}$ is a vector indexed by U.

Derive a clean closed form for the entries of M and $\boldsymbol{b}$ in terms of G, W and $\boldsymbol{f}$. That is, you should write a formula for $M(x,y)$ and for $\boldsymbol{b}(x)$. Make sure that your matrix M is *symmetric*.

4 Theoretical Problems

2. Let G be a connected, undirected graph and let u and v be two vertices in G. Fix α and let $\boldsymbol{p}_u$ and $\boldsymbol{p}_v$ be the Personal PageRank vectors starting at u and v respectively, with teleportation probability α. Prove that
$$\frac{\boldsymbol{p}_v(u)}{d(u)} = \frac{\boldsymbol{p}_u(v)}{d(v)}.$$

3. **Symmetry.**

   Let G be a connected, undirected, unweighted graph. Alternatively, you can think of every edge in G as having resistance 1. Treat G as a resistor network.

   Let (x,y) and (w,z) be two different edges of G. Let $\boldsymbol{i}_{x,y}(w,z)$ be the amount of current that flows over the edge (w,z) in the electrical flow that sends one unit of current from x to y. Prove that
$$\left|\boldsymbol{i}_{x,y}(w,z)\right| = \left|\boldsymbol{i}_{w,z}(x,y)\right|.$$

4. **Small Changes, Big Effects.**

   Sometimes it is possible to substantially change the flow on an edge in a resistor network by making very small changes to the resistances. Find a graph $G = (V,E)$, two distinguished nodes s and t, two sets of resistances $\boldsymbol{r}$ and $\boldsymbol{r}'$, and an edge (a,b), such that

   a. When one unit of current flows from s to t, the flow over edge (a,b) at least twice as large with resistances $\boldsymbol{r}'$ as with resistances $\boldsymbol{r}$; but,

   b. for all edges e,
$$\boldsymbol{r}(e) \leq \boldsymbol{r}'(e) \leq (11/10)\boldsymbol{r}(e).$$

   Justify your analysis.

5. Show that if you just increase the resistance on an edge e then you will not increase the flow on that edge.

   That is, let $G = (V,E)$ be an undirected, connected graph and let $\boldsymbol{r}$ be the edge resistances. Let e be an edge of E and let $\boldsymbol{r}'$ be another set of resistances such that
$$\begin{aligned}\boldsymbol{r}'(f) &= \boldsymbol{r}(f), \text{for all } f \neq e, \text{ and}\\ \boldsymbol{r}'(e) &\geq \boldsymbol{r}(e).\end{aligned}$$

   Let s and t be any two vertices of V. Prove that when we flow one unit of current from s to t, the amount of current that flows on edge e under resistances $\boldsymbol{r}'$ is at most the amount that flows under resistances $\boldsymbol{r}$.

   This problem might be hard.

5 Experiments

In this homework, we will solve systems of linear equations in matrices related to our graphs. So, you will need to get hold of some code for solving systems of linear equations. **Start early**, because this can be difficult.

The simplest way to solve a system of linear equations in a matrix is, of course, to invert the matrix. But, this can take a long time if the matrix is big. An alternative, which works if you matrix is positive semi-definite, is to use the Conjugate Gradient algorithm. This algorithm iteratively finds better and better approximations to the solution by repeatedly *multiplying* by the matrix. This can be much faster. But, you should check your answers. If your routine supplys an $\boldsymbol{x}$ for which you expect $A\boldsymbol{x} = \boldsymbol{b}$, you should check how close $A\boldsymbol{x}$ actually is to $\boldsymbol{b}$. If it is not too close, then you should run the algorithm for more iterations. You can find implementations of Conjugate Gradient (often called CG) in Matlab, the Python package SciPy, the C++ package TAUCS, and many many other places.

If your matrix is mostly zeros, you should try sparse matrix routines.

If you **start early** and your code is too slow, I will help you figure out how to solve your systems faster. This offer is only good if you have tried the above ideas at least one week before this homework is due.

2. Perform experiments analogous to those in problems 4 and 5 from the experimental part of Problem Set 2, but using Personal PageRank vectors. You may compute these either iteratively or by solving systems of linear equations.

   Note that there is a parameter α that you must choose while doing these experiments. You should try many different values of α. As you decrease α, the Personal PageRank vectors will become more spread out. You can measure how spread out a vector $\boldsymbol{p}$ is by summing its largest $n/2$ entries. Let T be the set of $n/2$ vertices as which $\boldsymbol{p}$ is largest. We will say that $\boldsymbol{p}$ is *well-spread* if
   $$\boldsymbol{p}(T) \leq \frac{3}{2} \frac{d(T)}{d(V)}.$$
   Start with $\alpha = 1/2$, and then keep halving α until you get a vector $\boldsymbol{p}$ that is well-spread.

   For each α use the vector you obtain to try to find a cut of low conductance. Report the least conductance found and the size of the set that has that least conductance as a function of α.

   Don't forget to run this many times (as many as on the last homework), starting with different seed vertices each time.

   Find some reasonable way to report this data.

3. In this problem, we will try to use electrical flow to interpolate a function on a graph.

   To begin, you should use your meta-data to find an interesting function f on the vertices of your graph. For example, if you have timestamps associated with nodes, the times would be a natural function. If your metadata identifies a particular set of notes S, you could use the characteristic function of this set S.

Either way, this function must be known at all nodes.

In a typical experiment, we will choose some set W of nodes at which we will reveal the value of the function f. We will then use an electrical flow (as in problem 1) to guess the values of the function on the nodes $U = V - W$. Call this guess $\boldsymbol{u}$. We will evaluate the quality of this guess by the *root mean squared* error:

$$\left(\frac{1}{|U|} \sum_{x \in U} (\boldsymbol{f}(x) - \boldsymbol{u}(x))^2 \right)^{1/2}.$$

A more naive algorithm would be to guess the mean:

$$\boldsymbol{u}(x) = \frac{1}{|W|} \sum_{y \in W} \boldsymbol{f}(y).$$

We will compare to the error of the mean guess.

a. Repeat the following 100 times (or 10 times if your graph has more than 100,000 edges or more than 10,000 nodes). Set W to be a randomly chosen ten percent of the nodes. Perform the experiment above. Report both how well $\boldsymbol{u}$ approximated $\boldsymbol{f}$ on U, and compare to the performance of the mean.

   If your function f is the characteristic function of a set S, you should make sure that in every run W contains elements both of S and $V - S$.

b. Repeat exeperiment a, but this time setting W to be a random half of the nodes.

c. Do it again, but this time with W being a random 90 percent of the nodes.

Submit all code. Make sure it is readable by a human. Add explanations where necesssary.