Homework 2

Due Friday, October 7th, 2011 by 5pm printed in my office or via email.

Problem 1: Shortest path computations

from the in-class quiz In class we saw how to compute single-source shortest path distances by using a matrix-over-a-semiring. Recall the algorithm from class:

$$\boldsymbol{A}^{\circ k}\mathbf{e}_i = \underbrace{\boldsymbol{A} \circ \boldsymbol{A} \circ \cdot \boldsymbol{A}}_{k \text{ times}} \circ \mathbf{e}_i.$$

a) Show how to adapt this to the problem of all-pairs shortest paths.

b) just think about it The following question is open-ended. I don't have an answer, nor do I think there is one right answer.
Just write down some of your thoughts about the question. This is a chance for you to explore the material yourselfs.

If you wanted to approximate all pairs shortest paths less expensively, what would you do? (Please don't describe how to implement the Floyd-Warshall procedure, think about approximating the shorest paths and doing so in a matrix sense.)

Problem 2

Kleinberg's HITS algorithm on a graph computes a hub-score and an authority-score for each vertex. Let $G = (V, E)$ be a directed graph. Then a hub should point to many authorities, and so we set the hub-score of a vertex to:

$$h_i = \sum_{(i,j)\in E} a_j.$$

Likewise, an authority should be pointed to by many hubs:

$$a_i = \sum_{(j,i)\in E} h_j.$$

Let $\boldsymbol{A}$ be the adjacency matrix, then

$$\mathbf{h} = \boldsymbol{A}\mathbf{a} \text{ and } \mathbf{a} = \boldsymbol{A}^T\mathbf{h}.$$

If we iterate these two, we find:

$$\mathbf{h} = \boldsymbol{A}\boldsymbol{A}^T\mathbf{h} \text{ and } \mathbf{a} = \boldsymbol{A}^T\boldsymbol{A}\mathbf{a}.$$

In other words, we are looking for eigenvalues of the matrices $\boldsymbol{A}\boldsymbol{A}^T$ and $\boldsymbol{A}^T\boldsymbol{A}$, respectively. As we've stated these, the eigenvalue would have to be 1. In the HITS algorithm, we keep iterating the updates:

$$\mathbf{h}^{(k+1)} = \boldsymbol{A}\mathbf{a}^{(k)} \text{ and } \mathbf{a}^{(k+1)} = \boldsymbol{A}^T\mathbf{h}^{(k)}$$

and then *normalizing* the results so that $\mathbf{h}^{(k+1)}$ and $\mathbf{a}^{(k+1)}$ have unit 2-norm. What this does is run the *power method* on the matrices $\boldsymbol{AA}^T$ and $\boldsymbol{A}^T\boldsymbol{A}$, simultaneously. We can use the Perron-Frobenius theorem to help us understand when this will have a unique solution.

Recall that a matrix has a unique Perron vector (positive eigenvalue equal to its spectral radius) when it is irreducible and aperiodic.

a) Give an example that shows an irreducible graph where $\boldsymbol{AA}^T$ or $\boldsymbol{A}^T\boldsymbol{A}$ is reducible.

b) Give an example that shows an irreducible, aperiodic graph where $\boldsymbol{AA}^T$ or $\boldsymbol{A}^T\boldsymbol{A}$ is reducible.

c) What this means is that it may be easy for the HITS algorithm to get into trouble with respect a unique solution because the Perron vector of $\boldsymbol{AA}^T$ or $\boldsymbol{A}^T\boldsymbol{A}$ may not be unique, even if it is for $\boldsymbol{A}$. So let's see what happens: Implement the HITS algorithm for graph.

d) Try at least four or five small graphs. Can you find a graph that is sensitive to the starting vector? If not, show that these graphs have a unique Perron vector by computing their eigenvalues.

Problem 3

Read http://www.ces.clemson.edu/~shierd/Shier/markov.pdf about how to check if a graph is aperiodic.

a) Describe an $O(|E|d)$ algorithm to check if a graph is aperiodic where the only possible operation with the graph is a matrix-vector product: $\boldsymbol{Av}$.

b) Describe how to use this algorithm to check whether or not $\boldsymbol{A}^T\boldsymbol{A}$ is irreducible.

c) Design an improved HITS algorithm that returns the hubs and authority vectors along with a flag that indicates whether or not the vectors are unique. Use as little extra work and memory as possible.

b) Open question Think about this question and write down your thoughts, or come chat with me about it. Is there a semi-ring operation that will check if a graph is aperiodic? Note that the operations greatest common divisor $= \otimes$ and least common multiple $= \oplus$ form a semi-ring over the non-negative intergers. At the very least, work out what the 0 and 1 elements are in this semi-ring. See http://marcpouly.ch/pdf/internal_100712.pdf for some other semirings that might be useful.

Problem 4: More semi-rings

In this problem, we'll show how a few things with semi-rings.

a) Show that $a \oplus b = \max(a,b)$ and $a \otimes b = \min(a,b)$ is a semi-ring and identify the 0 and 1 elements.

b) Show that a n-by-n permutation matrix is an orthogonal matrix in any semi-ring.

c) Open question Is there a semi-ring to implement Boruvka's algorithm for finding the minimum spanning tree? This would be a nice algorithm to implement in Hadoop or Pregel. Note that there is a semi-ring algorithm to compute minimum spanning trees using the semi-ring from part (a). This was described in MINIMUM-COST SPANNING TREE AS A PATH-FINDING PROBLEM by Maggs and Plotkin.

Problem 5: Markov chain state space classification

In this problem, you'll get some experience classifying the state of a Markov chain.

I'll update this with the example soon, check back on Tuesday the 27th.

Problem 6: Find a paper

Find a paper on networks and matrix computations that you'd like to present on for 30 minute in-class presentation. Please contact me with questions.