

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:

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

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

Solution Just compute $\mathbf{A}^{\circ k}$ where k is the diameter of the graph.

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 yourselves.

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 shortest paths and doing so in a matrix sense.)

Solution Anything involving an approach that would work for matrices was what was required for full points. But almost anything that wasn't Floyd-Warshall was given full. Note that a quick solution is to compute $\mathbf{A}^{\circ k}$ recursively via $\mathbf{A}^{\circ k/2} \circ \mathbf{A}^{\circ k/2}$.

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 \mathbf{A} be the adjacency matrix, then

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

If we iterate these two, we find:

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

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

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

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 $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$, 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 $\mathbf{A}\mathbf{A}^T$ or $\mathbf{A}^T\mathbf{A}$ is reducible.

Solution Many examples, the two-node directed cycle works.

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

Solution Many examples, a three-node directed cycle with a self-loop works.

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 $\mathbf{A}\mathbf{A}^T$ or $\mathbf{A}^T\mathbf{A}$ may not be unique, even if it is for \mathbf{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.

Solution The idea was to cook-up a graph that had two eigenvalues equal to the spectral radius, and then show that different starting vectors could pick any. For other types of non-uniqueness see: [Farahat, LoFaro, Miller, Rae, and Ward. Authority Rankings from HITS, PageRank, and SALSA: Existence, Uniqueness, and Effect of Initialization](#)

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: $\mathbf{A}\mathbf{v}$.

Solution First compute a breadth-first search using matvecs as in class. However, you need to identify all the non-tree edges.

To do this, you must ensure that the matvec only proceeds from nodes at level j at each iteration. Any node discovered at levels less than j are non-tree edges.

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

Solution We just record one extra vector of any non-zero result of a mat-vec we ever see. If, after the previous algorithm finishes, that vector is not completely filled, then the graph is reducible.

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.

Solution The simple solution is just to put these two algorithms together. A slightly better answer is to note that a periodic symmetric matrix has only one structure: period two. Thus, we can classify nodes into even or odd distance from the root. Note also that any matrix power $\mathbf{A}^k x$ must be in either an even-or-odd class if this structure holds.

We can get this structure by doing mat-vec products and the elements have to fall into one of two classes. So, we just check if we ever violate even/odd classes. This also means that the hits problem must start with a unit vector.

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 integers. 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.

Solution This works over any closed set. The zero element is the minimum of the set, and the one element is the maximum.

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

Solution the proof that a permutation matrix is orthogonal given in class only depends on the properties of the 0 and 1 element, namely: $0 \times x = 0$ and $1 \times 1 = 1$, which also hold in a 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.