CS/ENGRI 172, Fall 2002
10/25/02: Lecture Twenty-Four Handout

Topics: Kleinberg's (1998) hubs and authorities algorithm.

Conventions and notation

We'll ignore repeated links (i.e., if document d has two hyperlinks to document d', we only count one link between them). We'll also ignore self-links (links from a document to itself).

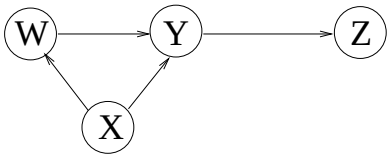
Let d be a document. We'll use the following shorthand notation:

To(d) the set of documents that link to d.

From(d) the set of documents that are linked to by d.

Notice that the in-degree of d is the number of documents in To(d).

Example Here are four documents, W, X, Y, and Z.



We have To(W) consisting of just X, whereas To(Y) is the two documents W and X. From(X) is the two documents W and Y, and From(Z) doesn't contain any documents. Furthermore, note that the in-degree of W is the same as the in-degree of Z, and that the out-degrees of W and Y are equal.

The hubs and authorities algorithm

Here's how the algorithm processes queries.[1] First, we retrieve a *root set* of (hopefully) relevant documents via content-based IR. (One may expand this root set by adding in the documents that link to or are linked from some document in the root set.) Let N be the number of documents in the root set, and for convenience let's call these documents $d_1, d_2, \ldots, d_N$. For each d_j in the root set, we want to compute its *authority score* a_j and its *hub score* h_j.

1. Initialization: For every document d_j, set both a_j and h_j to 1.

2. Repeat the following steps in order until no "significant" change:
3. Update authority scores: For every document d_j, change a_j to $\sum_{d_k \text{ in } \text{To}(d_j)} h_k$.
4. Pseudo-normalize authority scores: For every document d_j, change a_j to $a_j / \sum_{k=1}^{N} a_k$.
5. Update hub scores: For every document d_j, change h_j to $\sum_{d_k \text{ in } \text{From}(d_j)} a_k$.
6. Pseudo-normalize hub scores: For every document d_j, change h_j to $h_j / \sum_{k=1}^{N} h_k$.

Example (cont.)

		W		X		Y		Z	
		auth	(hub)	auth	(hub)	auth	(hub)	auth	(hub)
a.	Init	1	(1)	1	(1)	1	(1)	1	(1)
b.	Update-a	1	(1)	0	(1)	2	(1)	1	(1)
c.	PNorm-a	1/4	(1)	0	(1)	1/2	(1)	1/4	(1)
d.	Update-h	1/4	(1/2)	0	(3/4)	1/2	(1/4)	1/4	(0)
e.	PNorm-h	1/4	(1/3)	0	(1/2)	1/2	(1/6)	1/4	(0)
f.	Update-a	1/2	(1/3)	0	(1/2)	5/6	(1/6)	1/6	(0)
g.	Pnorm-a	1/3	(1/3)	0	(1/2)	5/9	(1/6)	1/9	(0)

[1] We're using pseudo-normalization rather than length-normalization to make the calculations a little easier.