CS/ENGRI 172, Fall 2003: Computation, Information, and Intelligence
10/27/03: Hubs and Authorites Algorithm

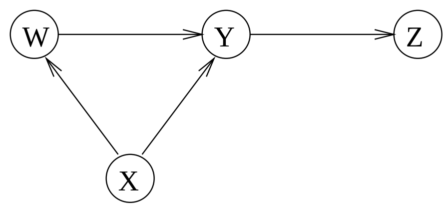
Conventions and Notation

Let d be a document; we'll use the following shorthand notation to describe the link structure surrounding d:

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

Running Example: This graph shows the link structure between 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. Futhermore, 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.

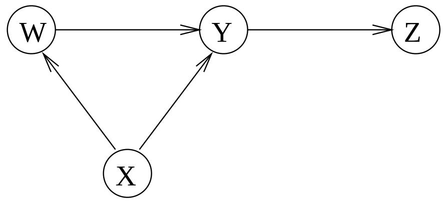
Hubs and Authorities Algorithm

The algorithm processes queries in the following manner. 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 changes occur:
3. Update authority scores: For every document d_j, change a_j to $\sum_{d_k \text{in To}(d_j)} h_k$
4. Pseudo-normalize[1] 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 $\sum_{d_k \text{in From}(d_j)} a_k$
6. Pseudo-normalize hub scores: For every document d_j, change h_j to $h_j / \sum_{k=1}^{N} h_k$

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

Running Example, Computing Scores:



The following table computes the authority and hub scores for the four nodes in our example graph, for the first two iterations of the hubs and authorities algorithm:

	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)
h. Update-h	1/3	(5/9)	0	(8/9)	5/9	(1/9)	1/9	(0)
i. Update-h	1/3	(5/14)	0	(4/7)	5/9	(1/14)	1/9	(0)

Row e shows the result of running one iteration of the algorithm; row i shows the result of running two iterations of the algorithm.