CS 559: Algorithmic Aspects of Computer Networks		Fall 2007
Lecture 17 — November 7		
Lecturer: John Byers	*BOSTON UNIVERSITY*	*Scribe: Flavio Esposito*

In this lecture, the last part of the **PageRank** paper has been presented [3] and a discussion about the **Hubs and Authorities** paper [2] begun.

17.1 The Bow-tie model

Definition 1. Given a directed graph $G = G(V, E)$, the **fan out or out-degree** of a vertex V in G is the size of the subset F of E of edges that are going from V to all the other vertices.

Definition 2. Given a directed graph $G = G(V, E)$, the **fan in or in-degree** of a vertex V in G is the size of the subset H of E of edges that are going to V from all the other vertices.

Note that a vertex V can be highly ranked in a graph G if it has either (both) high fan out or (and) high fan in. The insight is the fan in is more important than the fan out but both brings up in the ranking a web page.

Recall that in the PageRank approach the graph $G(V, E)$ has as vertex the web pages and as edges the hyperlinks.

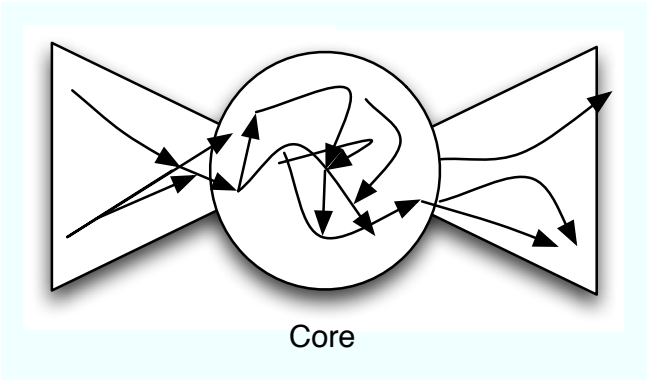

Figure 17.1. bow-tie model: If we only consider the graph bounded by the clique (core), we can prove that an equilibrium point exist: if we have also tends parts, i.e. vertex with fan in zero (left tend) or with fan out zero (right tend) then during a walk we can get stuck (dead end).

Recalling the PageRank markovian approach, the idea is to find an equilibrium vector P of probabilities of a random walker being in vertex V after k steps when $k \to \infty$. The process of finding the vector P ends when P does not change more than a given δ. So If we consider all the bow tie, we might have to wait for a bigger amount of steps to find the final P, or we can have no equilibrium point at all [1].

We use then *random jumps* in the graph instead of a linear walk to visit the all nodes. A random jump approach has two properties:

- Order persevering respect to a vertex V as starting point. First of all, if I change a starting point the order has to be the same. Moreover, if I sort the elements of the vector P, without jumps, and then on the same graph I sort the element of the P found, I might find different values of p_i but the bigger is the same, the second bigger is the same, and so on.
- Speed up convergence for sampling based methods, i.e. $P_{k_1} < P_{k_2}$ where P_{k_2} is the final equilibrium point after k_2 steps without random jumps and P_{k_1} is with random jumps, given the same δ.

17.1.1 Boost the ranking

The idea is that, a page is visited more often if I am being linked from more popular websites.

In other words, a way to increase the probability of getting hit by a *random walker* surfing the web is by getting a link from a very popular website and or to create link to popular websites. (so increase the fan in and the fan out of one page boost the ranking of a page). Another approach could be to create a set of nodes (web pages) all linked one another in a fully connected graph fashion. In this manner, the probability of an user hitting the set of node (pages) connected by edges (pages) will be higher.

17.2 Hubs and Authorities

Assume that we have a topic T (represented by the set of pages in the cloud), and a set of pages i.e. a region related to T.

Definition 3. Given a directed graph $G = G(V, E)$ and a topic T, a vertex V is called **hub for** T if there is a significant number of edges starting from V and ending into a vertex (page) of T.

Definition 4. Given a directed graph $G = G(V, E)$ and a topic T, a vertex V is called **authority for** T if there is a significant number of edges starting from any vertex in T and ending in V.

Observation : Like is depicted in figure, hubs can have also edges that go out of T and an authority can have edges that comes from vertex outside from T but we are not interested in this edges

Kleinberg idea to rank the web pages is based on two main steps:

1. Find a collection of URL^s on a topic T.
2. Derive an algorithm to identify hubs and authorities from the previous step(1.).

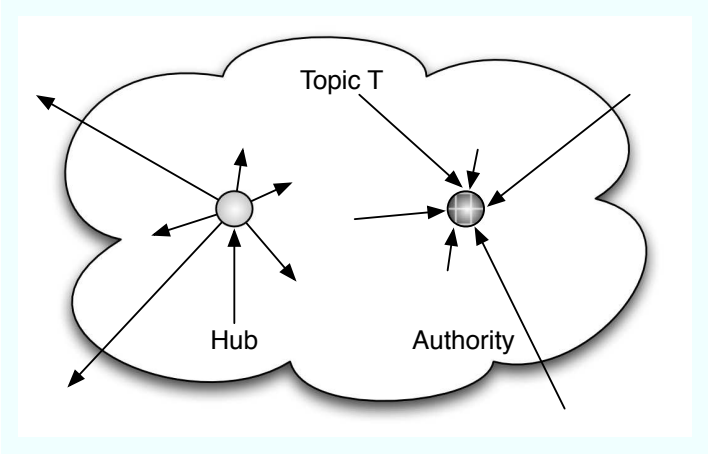

Figure 17.2. Hubs and Authorities: on the left, a page (vertex) with many edges (links) pointing to other pages (vertex) of the same topic T i.e. an hub. On the right a page (vertex) with many edges (links) pointing from other pages (vertex) of the same topic T i.e. an authority.

The challenging part is of course to find the collection of hubs and authorities. To accomplish point 1., Kleinberg conducts a text based search: to find some (200) results with a web search engine (HotBot, Altavista) and then, broading the set of URL^s by adding one-hop neighbor downstream (the edge representing a son in the large graph), and one-hop neighbor upstream (the father). Two observations: the first one is that in order to gather information of the down and upstream neighborhood we need the knowledge of the graph.

The second observation is the reason why Kleinberg uses only one hop neighborhood information: whenever the node found is an authority, the set of k ($k > 1$) hop neighborhood can be too vast; that is why the identification algorithm has as input only with one hop information.

Just looking at the degree of any node in the subgraph S of G associated with the topic T, the system is able to cluster hubs and authorities.

Kleinberg formalizes an authorities and an hub weight as:

Definition 5. Given a graph $G = G(V, E)$ and a topic T of G (subgraph S), an authority of weight p for T, is the quantity $\{x^{<p>}\}$

Definition 6. Given a graph $G = G(V, E)$ and a topic T of G (subgraph S), an hub of weight p for T, is the quantity $\{y^{<p>}\}$

The algorithm identifies hubs and authorities updating k times the weights with two operations, denoted O meaning Output degree of a vertex V of G, and I meaning Input degree.

$$x^{<p>} \leftarrow \sum_{p:(p,q)\in E} y^{<q>} \tag{17.1}$$

[1]The algorithm runs with good results if $k = 20$. After 20 steps of the algorithm experimentally has been found that no changes of the first top 50 (up to 200) are observed (equilibrium values have been reached).

and

$$y^{<p>} \leftarrow \sum_{p:(p,q)\in E} x^{<q>} \tag{17.2}$$

Moreover, a maintaining invariance conditions (normalizing the coefficient to 1 after each step) have to be satisfied, i.e. $\sum_{p\in S_\sigma}(x^{<p>})^2 = 1$ and $\sum_{p\in S_\sigma}(y^{<p>})^2 = 1$ where σ is a query string. As we can see form the formulas above, the hubs power are given by the authorities and vice-versa.

17.2.1 Pseudocode of the Iterative Algorithm

Let us see the pseudocode of the algorithm to identify hubs and authorities

Iterate(G, k)
G is a graph collection of n linked pages, k number of rounds
  z is an unitary vector of $\mathbb{R}^n$ ($x_0 = z$ and $y_0 = z_i$)
  for i = 1,. . .,k do
    apply "I" operation to $x_{i-1}, y_{i-1} \rightarrow x_i^{'}$
    apply "O" operation to $x_{i-1}, y_{i-1} \rightarrow y_i^{'}$
    Normalize $x_i^{'} \rightarrow x_i$
    Normalize $y_i^{'} \rightarrow y_i$
end for
return (x_k, y_k)

17.2.2 Sketch of the proof of convergence

Given M n x n symmetric and A adjacency matrix (matrix of connectivity and so symmetric and n x n as well) and the vectors x_i of authorities and y_i of hubs, the proof of convergence is based on the follow logic steps:

- Hubs and Authorities are updated as follow: $x_i \leftarrow A^t y_{i-1}$ and $y_i \leftarrow A x_{i-1}$.
- Expanding until there are no more iterations, we have:
  $x_k \leftarrow (A^t A)^{k-1} A^t x_0$ and
  $y_k \leftarrow (A^t A)^k y_0$

  notice that in class the updates where written in a slightly different notation from that one in the paper: but the matrix M in your notes is suppose to be A:
  $x_i \leftarrow (A^t A) x_{i-2}$
  $x_{2k} \leftarrow (A^t A)^k x_0$
  and the same for y.
- From linear algebra is known that if x_0 is a vector not orthogonal to the principal eigenvector ω of M, then the vector x_0 in the direction of $M^k x_0$ converge to ω and so the sequence $\{x_k\} \rightarrow$x$_0$ for k $\rightarrow \infty$.

Bibliography

[1] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. In *Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications networking*, pages 309–320, Amsterdam, The Netherlands, The Netherlands, 2000. North-Holland Publishing Co.

[2] J.Kleinberg. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 46, 1999.

[3] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The Pagerank Citation Ranking: Bringing Order to the Web. Technical report, Stanford Digital Library Technologies Project, 1998.