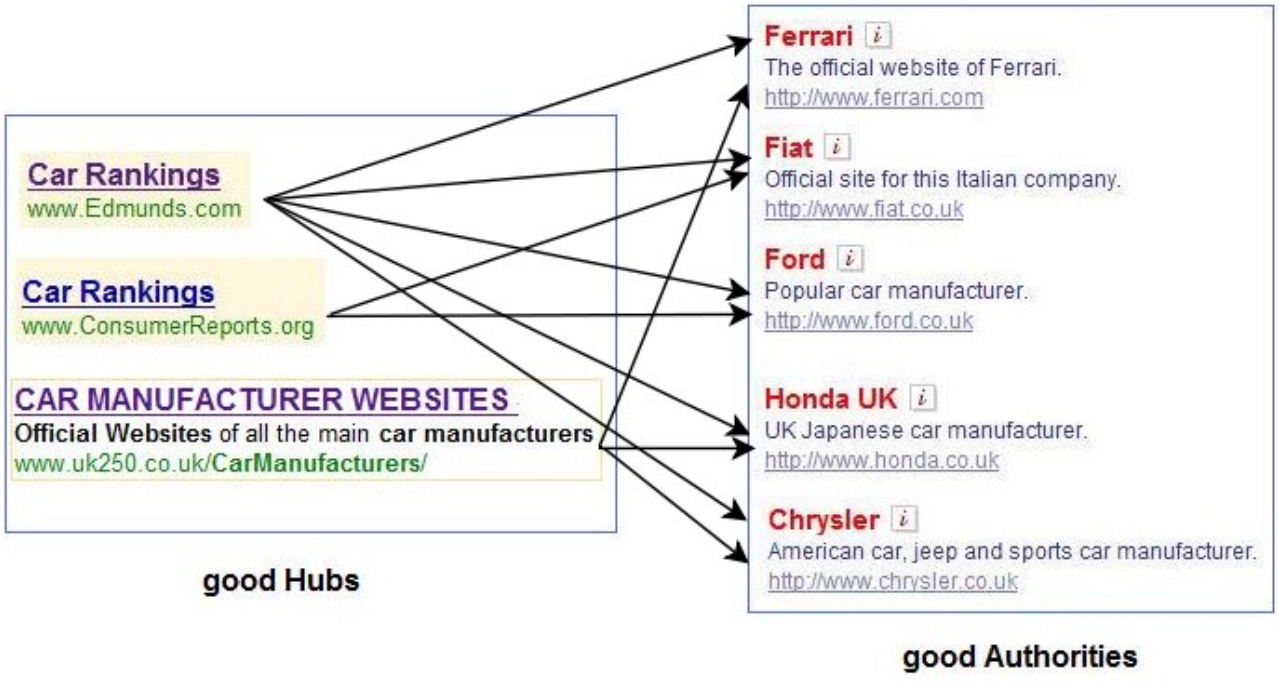
Lecture #4: HITS Algorithm - Hubs and Authorities on the Internet

In the same time that PageRank was being developed, Jon Kleinberg a professor in the Department of Computer Science at Cornell came up with his own solution to the Web Search problem. He developed an algorithm that made use of the link structure of the web in order to discover and rank pages relevant for a particular topic. **HITS** ***(hyperlink-induced topic search)*** is now part of the **Ask** search engine (www.Ask.com).

One of the interesting points that he brought up was that the human perspective on how a search process should go is more complex than just compare a list of query words against a list of documents and return the matches. Suppose we want to buy a car and type in a general query phrase like "the best automobile makers in the last 4 years", perhaps with the intention to get back a list of top car brands and their official web sites. When you ask this question to your friends, you expect them to be able to understand that automobile means car, vehicle, and that automobile is a general concept that includes vans, trucks, and other type of cars. When you ask this question to a computer that is running a text based ranking algorithm, things might be very different. That computer will count all occurrences of the given words in a given set of documents, but will not do intelligent rephrasing for you. The list of top pages we get back, while algorithmically correct, might be very different than what expected. One problem is that most official web sites are not enough self descriptive. They might not advertise themselves the way general public perceives them. Top companies like Hunday, Toyota, might not even use the terms "automobile makers" on their web sites. They might use the term "car manufacturer" instead, or just describe their products and their business.

What is to be done in this case? It would be of course great if computers could have a dictionary or ontology, such that for any query, they could figure out sinonimes, equivalent meanings of phrases. This might improve the quality of search, nevertheless, in the end, we would still have a text based ranking system for the web pages. We would still be left with the initial problem of sorting the huge number of pages that are relevant to the different meanings of the query phrase. We can easily convince ourselves that this is the case. Just remember one of our first examples, about a page that repeats the phrase "automobile makers = cars manufacturers = vehicle designers" a billion times. This web page would be the first one displayed by the query engine. Nevertheless, this page contains practically no usable information.

The conclusion is that even if trying to find pages that contain the query words should be the starting point, a different ranking system is needed in order to find those pages that are ***authoritative*** for a given query. Page i is called an ***authority*** for the query "automobile makers" if it contains valuable information on the subject. Official web sites of car manufacturers, such as www.bmw.com, HyundaiUSA.com, www.mercedes-benz.com would be authorities for this search. Commercial web sites selling cars might be authorities on the subject as well. These are the ones truly relevant to the given query. These are the ones that the user expects back from the query engine. However, there is a second category of pages relevant to the process of finding the authoritative pages, called ***hubs***. Their role is to advertise the authoritative pages. They contain useful links towards the authoritative pages. In other words, hubs point the search engine in the "right direction". In real life, when you buy a car, you are more inclined to purchase it from a certain dealer that your friend recommends. Following the analogy, the authority in this case would be the car dealer, and the hub would be your friend. You trust your friend, therefore you trust what your friend recommends. In the world wide web, hubs for our query about automobiles might be pages that contain rankings of the cars, blogs where people discuss about the cars that they purchased, and so on.



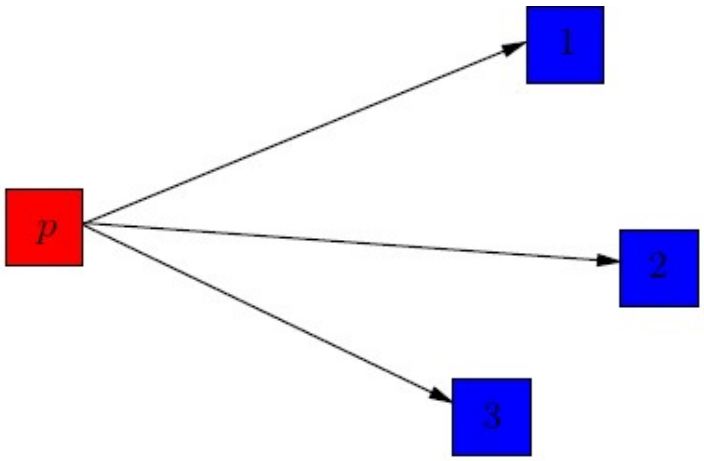
Query: Top automobile makers

Jon Kleinberg's algorithm called **HITS** identifies good authorities and hubs for a topic by assigning two numbers to a page: an authority and a hub weight. These weights are defined recursively. A higher authority weight occurs if the page is pointed to by pages with high hub weights. A higher hub weight occurs if the page points to many pages with high authority weights.

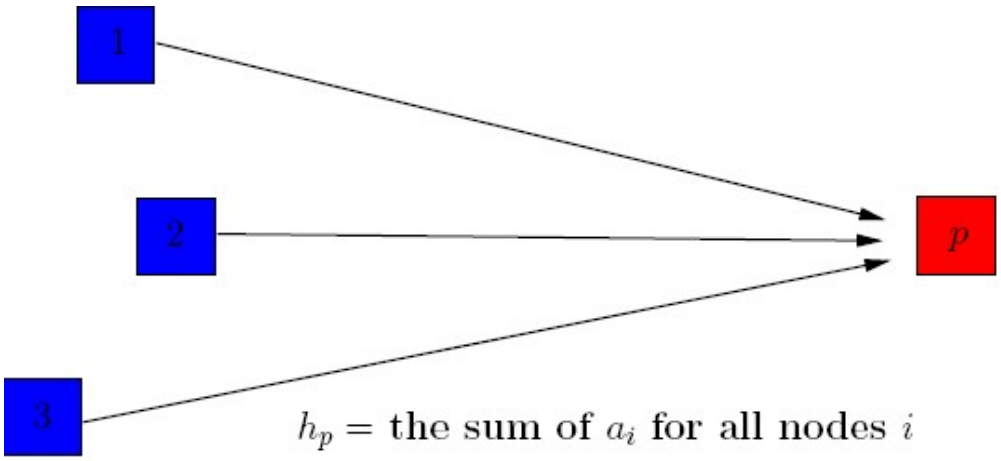
In order to get a set rich in both hubs and authorities for a query Q, we first collect the top 200 documents that contain the highest number of occurrences of the search phrase Q. These, as pointed out before may not be of tremendous practical relevance, but one has to start somewhere. Kleinberg points out that the pages from this set called root (R_Q) are essentially very heterogeneous and in general contain only a few (if any) links to each other. So the web subgraph determined by these nodes is almost totally disconnected; in particular, we can not enforce Page Rank techniques on R_Q.

Authorities for the query Q are not extremely likely to be in the root set R_Q. However, they are likely to be pointed out by at least one page in R_Q. So it makes sense to extend the subgraph R_Q by including all edges coming from or pointing to nodes from R_Q. We denote by S_Q the resulting subgraph and call it the *seed* of our search. Notice that S_Q we have constructed is a reasonably small graph (it is certainly much smaller then the 30 billion nodes web graph!). It is also likely to contain a lot of authoritative sources for Q . The question that remains is how to recognize and rate them? Heuristically, authorities on the same topic should have a lot of common pages from S_Q pointing to them. Using our previous terminology, there should be a great overlap in the set of hubs that point to them.

From here on, we translate everything into mathematical language. We associate to each page i two numbers: an authority weight a_i, and a hub weight h_i. We consider pages with a higher a_i number as being better authorities, and pages with a higher h_i number as being better hubs. Given the weights $\{a_i\}$ and $\{h_i\}$ of all the nodes in S_Q, we dynamically update the weights as follows:



a_p = the sum of h_i for all nodes i pointing to p



h_p = the sum of a_i for all nodes i pointed to by p

A good hub increases the authority weight of the pages it points. A good authority increases the hub weight of the pages that point to it. The idea is then to apply the two operations above alternatively until equilibrium values for the hub and authority weights are reached.
Let A be the adjacency matrix of the graph S_Q and denote the authority weight vector by v and the hub weight vector by u, where

$$v = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \quad \text{and} \quad u = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix}$$

Let us notice that the two update operations described in the pictures translate to:

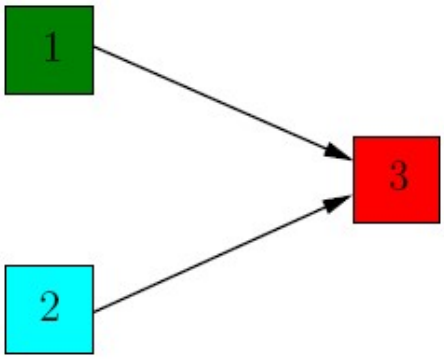
$$\begin{cases} v = A^t \cdot u \\ u = A \cdot v \end{cases}.$$

If we consider that the initial weights of the nodes are

$u_0 = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$ and $v_0 = A^t \cdot \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$ then, after *k* steps we get the system:

$$\begin{cases} v_k = (A^t \cdot A) \cdot v_{k-1} \\ u_k = (A \cdot A^t) \cdot u_{k-1} \end{cases}.$$

Example: Let us consider a very simple graph:



The adjacency matrix of the graph is $A = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$, with transpose

$A^t = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$. Assume the initial hub weight vector is: $u = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$.

We compute the authority weight vector by:

$$v = A^t \cdot u = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

Then, the updated hub weight is:

$$u = A \cdot v = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix}.$$

This already corresponds to our intuition that node 3 is the most authoritative, since it is the only one with incoming edges, and that nodes 1 and 2 are equally important hubs. If we repeat the process further, we will only obtain scalar multiples of the vectors v and u computed at step 1. So the relative weights of the nodes remain the same.

For more complicated examples of graphs, we would expect the convergence to be problematic, and the equilibrium solutions (if there are any) to be more difficult to find.

Theorem: Under the assumptions that AA^t and A^tA are primitive matrices, the following statements hold:

1. If $v_1, \ldots, v_k, \ldots$ is the sequence of authority weights we have computed, then $\boldsymbol{V_1}, \ldots, \boldsymbol{V_k}, \ldots$ converges to the unique probabilistic vector corresponding to the dominant eigenvalue of the matrix A^tA. With a slight abuse of notation, we denoted in here by $\boldsymbol{V_k}$ the vector v_k normalized so that the sum of its entries is 1.
2. Likewise, if $u_1, \ldots, u_k, \ldots$ are the hub weights that we have iteratively computed, then $\boldsymbol{U_1}, \ldots, \boldsymbol{U_k}, \ldots$ converges to the unique probabilistic vector corresponding to the dominant eigenvalue of the matrix AA^t. We use the same notation, that $\boldsymbol{U_k} = (1/c)u_k$, where c is the scalar equal to the sum of the entries of the vector u_k.

So the authority weight vector is the probabilistic eigenvector corresponding to the largest eigenvalue of A^tA, while the hub weights of the nodes are given by the probabilistic eigenvector of the largest eigenvalue of AA^t.

In the background we rely on the following mathematical theorems:

Theorems:

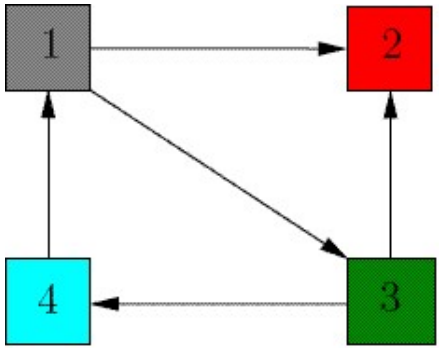
1. The matrices AA^t and A^tA are real and symmetric, so they have only real eigenvalues.
2. **Perron Frobenius**. If M is a primitive matrix, then:
   i. The largest eigenvalue λ of M is positive and of multiplicity 1.
   ii. Every other eigenvalue of M is in modulus strictly less than λ
   iii. The largest eigenvalue λ has a corresponding eigenvector with all entries positive.
3. Let M be a non-negative symmetric and primitive matrix and v be the largest eigenvector of M, with the sum of its entries equal to 1. Let z be the column vector with all entries non-negative, then, if we normalize the vectors $z, Mz, \ldots, M^kz$, then the sequence converges to v.

We use the notion of "convergence" in here in a loose sense. We say that a sequence of vectors z_k converges to a vector v in the intuitive sense that as k gets big, the entries in the column vector z_k are very close to the corresponding entries of the column vector v. Without going into the technical details of the proof, the power method works because we have only one largest eigenvalue that dominates the behavior.

HITS algorithm is in the same spirit as **PageRank**. They both make use of the link structure of the Weg graph in order to decide the relevance of the pages. The difference is that unlike the **PageRank** algorithm, **HITS** only operates on a small subgraph (the seed S_Q) from the web graph. This subgraph is query dependent; whenever we search with a different query phrase, the seed changes as well. **HITS** ranks the seed nodes according to their authority and hub weights. The highest ranking pages are displayed to the user by the query engine.

Problem 1: Prove that for any square matrix A, the matrices A^tA and AA^t are symmetric.

Problem 2: Compute the hub and authority weights for the following graph:



Hint: Compute the adjacency matrix A and show that the eigenvalues of AA^t are 0,1, and 3. The normalized eigenvector for the largest eigenvalue $\lambda = 3$ is the hub weight vector. What can you say about the authority weights of the 4 nodes? Does this correspond to your intuition?

Back Table of Contents Next