Lecture Notes: Social Networks: Models, Algorithms, and Applications
Lecture 28: Apr 26, 2012
Scribes: Mauricio Monsalve and Yamini Mule

1 How big is the Web

How big is the Web? In the past, this question could was ”easily answered” by counting the number of files in the servers; in 1997, Bharat and Broder estimated the size of the Web to be around 200 million. But the Web has grown considerably over the years and many web pages are generated dynamically today. For example, many pages are the result of sending a form or using [GET] parameters on links. So, how should we count these dynamically generated paged? One solution is to let the *indexable Web* be the Web we are talking about. The indexable Web is the Web reachable and stored by the web crawlers of the search engines.

In 2005, Gulli and Signorini (U of Iowa) used the notion of *indexable Web* to estimate the size of the Web [1]. They compared the number of indexed pages reported by 4 search engines, Google, Yahoo!, Ask.com and MSN, and estimated how many pages were crawled by more than one search engine. The result was that the Web had a size of approximately 11.5 billion pages.

Their methodology consisted in the following. Let A, B, etc, be search engines. Let $s(A)$ be the number of pages indexed by search engine A. Now, let $\Pr(A\&B|A)$ define the probability a web page is in indexed by search engines A and B, given that we know it is in A. Then,

$$\Pr(A\&B|A) \approx \frac{s(A\&B)}{s(A)}.$$

If we divide the same probabilities given different conditionals, we obtain

$$\frac{\Pr(A\&B|A)}{\Pr(A\&B|B)} \approx \frac{s(B)}{s(A)}.$$

Therefore, it is possible to relate the sizes of all the search engines through these conditional probabilities, and estime their overlap at the same time. Their results are shown in Fig. 1.

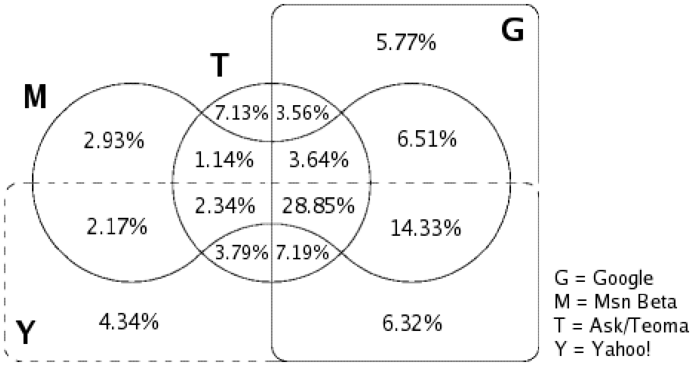

Figure 1: The size and overlap between search engine databases. From [1].

These probabilities $\Pr(A \cap B|A)$ were estimated thorough sampling. To get samples, they used pages from `dmoz.org`, a human curated directory of web pages. This ensured that the indexed pages had real meaning. Gulli and Signorini retrieved around 4 million pages and 2 million terms. They sorted these terms by ocurrence count. Then, they divided the terms in bags of 20 words, and proceeded to obtain the probabilities thorough searches. Then, the sizes s were obtained thorough linear regression.

2 Ranking and searching pages

Around 1997, Yahoo! was famous for its *ontology*: they kept an extensive human curated directory/ontology of web pages. Users went to this ontology to search for what they were looking for. But search engines appeared, taking the attention away from ontologies. In search engines, a few keywords satisfied to retrieve some *meaningful links*. But some time was needed for search results to improve.

Research for improving search engines focused in developing *ranking functions*, so that search results included the *best* results firsts. One of the main ideas behind these developments was to return the *most authoritative pages*. This line of work devoted to defining what being *authoritative* means and how to identify these pages.

2.1 Solution 1: Altavista, HotBot, Lycos

One of the first solutions, adopted by Altavista, HotBot and Lycos, consisted in using the *indegree* of web pages as a measure of their authority. Note that the indegree of a web page is the number of links pointing to that page. For example, in the graph of Fig. 2.1, the rightmost node is the most authoritative one, with indegree 3, while all the other nodes have indegree 1.

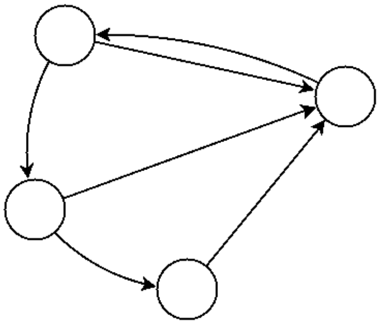

Figure 2: A graph of indegrees of 1, 1, 1, and 3.

An obvious problem with this approach is that it is easily played. As some web page/site owners wanted to have their pages visited more often, they *artificially* increased the number of incoming links, Thus, their pages got more views by appearing in the first places of their related searches. Soon, companies for *search engine optimization* appeared, selling the service of improving web pages' ranks. (The current situation of these companies is different. In the presence of Google, they try hard to reverse engineer how this search engine works. Also, Google has become a player in that industry too, allowing pages to appear on top of search results if they pay for that and by periodically changing how their search works, making cheating (competition) difficult. However, this is a tricky business. Search result quality worsens as more pages are allowed to appear on top.)

2.2 Solution 2: PageRank

PageRank was introduced by Sergey Brin and Lawrence Page [2], and is the original algorithm used by Google to search the Web. But as mentioned before, Google has changed over time and its current algorithm is kept secret.

PageRank differs from the previous approach as it considers the pointer, not only the pointee. The basic idea behind this approach is that *a page has high rank if the sum of the ranks of the pointing pages is high.* PageRank works as follows:

```
PageRank($G$)
    $rank_v$:=$1/|V(G)|$, for all $v \in V(G)$
    Until "convergence":
            $rank'_v$:=$\sum_{\{u,v\}\in E(G)} rank_u/deg(u)$, for all $v \in V(G)$
            $rank_v$:=$rank'_v$, for all $v \in V(G)$
```

The solution *converges* when $rank'_v \approx rank_v$.

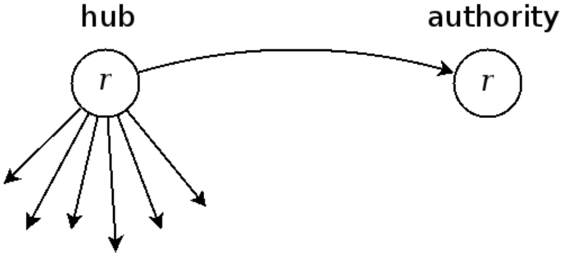

Figure 3: Hubs spread their rank to other web pages.

One of the properties of the above algorithm is that it keeps the solution summing 1. But it has a big drawback: mutually supportive nodes (Fig. 2.2) tend to absorb all the rank. Some fixes have been introduces to solve this problem, including:

1. *Random Restart.* The PageRank algorithm can be seen as a random walk procedure, where the contribution of node u to v is the probability of hitting v starting from u at some time t, for $t \gg 0$ (a steady probability). But in the presence of mutually supportive nodes, the random walk would get trapped in the cycle(s). Therefore, one solution is that, at each time step, a random walk can return to the origin node u with some probability.

2. *Tax method.* Back to the original computation, after the new rank is computed ($rank'$), it is taxed and these taxes contribute equally to the ranks of all nodes. This solution still converges.

   Let us study this method through linear algebra. Let us start with classic PageRank. Let A be the adjacency matrix of the Web graph. Define matrix N as

$$N_{ij} = \frac{A_{ij}}{deg(i)},$$

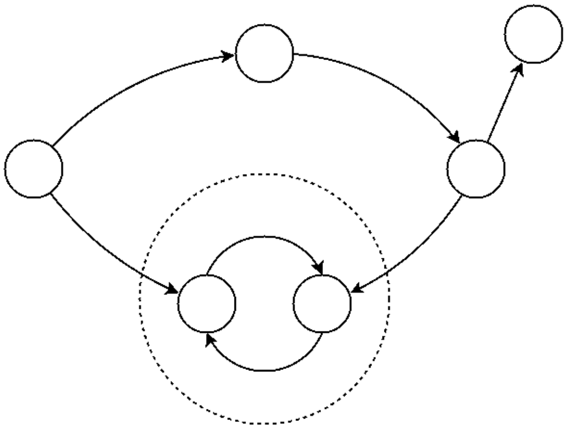

Figure 4: A graph with a rank sink.

i.e. the normalized version of A. Let $rank = (r_1, r_2, ..., r_n)$ be the current rank vector. Then, after one iteration, the new rank of i is

$$r_i' = N_{1i}r_1 + N_{2i}r_2 + ...N_{ni}r_n,$$

so the whole new rank vector is

$$rank' = N^\top rank.$$

Now, in the taxed case, we tax a proportion s to all nodes. We can specify this through a new normalized matrix $\tilde{N}$ such that

$$\tilde{N}_{ij} = sN_{ij} + \frac{1-s}{n}.$$

Using $\tilde{N}$ instead of N does not guarantee $\sum_{i \in V(G)} r_i = 1$. Thus, due to the necessity to renormalize, the taxed version of PageRank is slower than the original.

2.3 Solution 3: HITS

HITS is a precursor of PageRank. HITS stands for Hyperlink-Induced Topic Search, and was introduced by Jon Kleinberg [3]. It is said that a similar algorithm was used by Teoma (which was later acquired by Ask.com).

HITS starts "locally": the algorithm starts with a small set of nodes called the *root*. Every node has two sides: it is both a hub and an authority. In practice, it may not be a hub (if it does not have links) or it may not be an authority (if it is not linked-to and it is in the root). Some basic ideas behind the quality of a hub and an authority are that:

1. A good authority has high indegree,
2. a good hub has high outdegree,
3. it matters for authority to be pointed by good hubs, and
4. it matters for hubs to point good authorities.

The Web graph is then described as a bipartite graph G of hubs and authorities (see Fig. 2.3), where a web page can be both. And pages have two scores: a-score (for authority) and h-score (for hubs). These scores are normalized to sum 1 independently.

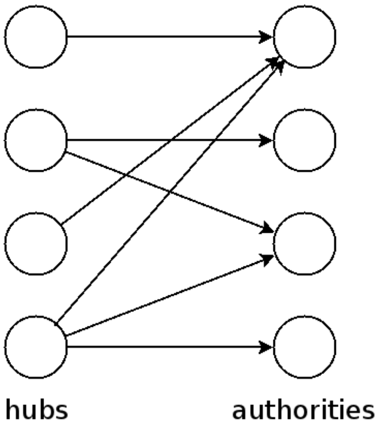

Figure 5: The Web as a bipartite graph of hubs and authorities.

The scores are computed by the following updating operations:

1. $a_p{:=}\sum_{(q,p)\in E(G)} h_q$,
2. $h_p{:=}\sum_{(p,q)\in E(G)} a_q$,
3. renormalization.

Regardless of the initial values of a-score and h-score, as long as their are positive, the method converges.

Let us study this method through linear algebra. Let M be the adjacency matrix of the bipartite graph G and let vectors $\vec{a}_0, \vec{h}_0$ consist of non negative values (but not only zeroes!). Then, the new a-scores are

$$\vec{a}_1 = M^\top \vec{h}_0,$$

and the new h-scores are

$$\vec{h}_1 = M\vec{a}_1 = MM^\top \vec{h}_0.$$

Ignoring the renormalization step, we can repeat and obtain

$$\vec{a}_2 = M^\top \vec{h}_1 = M^\top MM^\top \vec{h}_0,$$

and

$$\vec{h}_2 = M\vec{a}_2 = MM^\top MM^\top \vec{h}_0.$$

Following by simple induction, we get that

$$\vec{a}_k = (M^\top M)^{k-1} M^\top \vec{h}_0,$$

and

$$\vec{h}_k = (MM^\top)^k \vec{h}_0.$$

(It can be shown that these vectors converge in direction. This can be done through the properties of matrix $MM^\top$, which is symmetric semidefinite positive.)

References

[1] Antonio Gulli, Alessio Signorini (2005). The indexable Web is more than 11.5 billion pages. WWW Conference 2005.

[2] Sergey Brin, Lawrence Page (1998). The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems 30: 107117

[3] Jon Kleinberg (1999). Authoritative sources in a hyperlinked environment. Journal of the ACM 46 (5): 604632