Web Search: Ranking Web Pages

CPS 296.1

Topics in Database Systems

How do Web search engines work?

- Arasu et al, "Searching the Web." *ACM Trans. on Internet Technology*, 2001
- Crawling
  - Download Web pages
- Indexing
  - Index downloaded pages to facilitate searching and future crawls
- Searching

search engine | Search

Characteristics of Web search

- Huge amounts of text to search through
- Pages are linked
- Pages differ greatly in quality
- A single search may return many pages
  - A user will not look at all result pages
  - Result pages need to be ranked
  - Complete result set may be unnecessary

Ranking result pages

- Based on content
  - Number of occurrences of the search terms
  - Similarity to the query text
- Based on link structure
  - Backlink count
  - PageRank
  - Hub and authority scores (HITS)
- And more…

Textual similarity

- Vocabulary: $[w_1, \ldots, w_n]$
- IDF (Inverse Document Frequency): $[f_1, \ldots, f_n]$
  - $f_i = 1 / $ the number of times w_i appears on the Web
- Significance of words on page p: $[p_1 f_1, \ldots, p_n f_n]$
  - p_i is the number of times w_i appears on p
- Textual similarity between two pages p and q is defined to be $[p_1 f_1, \ldots, p_n f_n] \cdot [q_1 f_1, \ldots, q_n f_n] = p_1 q_1 f_1^2 + \ldots + p_n q_n f_n^2$
  - q could be the query text
- There are other IDF definitions in the IR literature

Why weight significance by IDF?

- "the" occurs frequently on the Web, so its occurrence on a particular page should be considered less significant
- "engine" occurs infrequently on the Web, so its occurrence on a particular page should be considered more significant
- Without IDF weighting, the similarity measure would be dominated by the so-called stop words

Problems with content-based ranking

- Many pages containing search terms may be of poor quality or irrelevant
  - Example: a page with just a line “search engine”
- Many high-quality or relevant pages do not even contain the search terms
  - Example: Google homepage
- Page containing more occurrences of the search terms are ranked higher; spamming is easy
  - Example: a page with line “search engine” repeated many times

Backlink

- A backlink of a page *p* is a link that points to *p*
- A page with more backlinks is ranked higher
- Intuition: Each backlink is a “vote” for the page’s importance
- Based on local link structure; still easy to spam
  - Create lots of pages that point to a particular page

PageRank and HITS

- Page et al., “The PageRank Citation Ranking: Brining Order to the Web.” 1998
- Kleinberg, “Authoritative Sources in a Hyperlinked Environment.” *Journal of the ACM*, 1999
- Main idea: Pages pointed by high-ranking pages are ranked higher
- Definition is recursive by design
- Based on global link structure; hard to spam

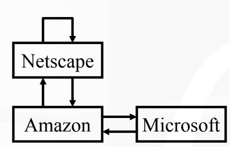
Naïve PageRank

- $N(p)$: number of outgoing links from page p
- $B(p)$: set of pages that point to p
- $\text{PageRank}(p) = \sum_{q \in B(p)} (\text{PageRank}(q) / N(q))$
- Intuition
  - Each page q evenly distributes its importance to all pages that q points to
  - Each page p gets a boost of its importance from each page that points to p

Definition in linear algebra

- Create a stochastic matrix M for the link structure
  - Each page i corresponds to row i and column i
  - If page j has n outgoing links, then the $M(i, j) = 1/n$ if page j points to page i, or 0 otherwise
- $$\begin{bmatrix} \text{PageRank}(p_1) \\ \text{PageRank}(p_2) \\ \ldots \\ \text{PageRank}(p_m) \end{bmatrix} = M \begin{bmatrix} \text{PageRank}(p_1) \\ \text{PageRank}(p_2) \\ \ldots \\ \text{PageRank}(p_m) \end{bmatrix}$$
- Solve by setting all PageRank’s to 1 initially, and then applying M repeatedly until the values converge

Naïve PageRank example



$$\begin{bmatrix} n \\ m \\ a \end{bmatrix} = \begin{bmatrix} 0.5 & 0 & 0.5 \\ 0 & 0 & 0.5 \\ 0.5 & 1 & 0 \end{bmatrix} \begin{bmatrix} n \\ m \\ a \end{bmatrix}$$

$$\begin{bmatrix} n \\ m \\ a \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0.5 \\ 1.5 \end{bmatrix}, \begin{bmatrix} 1.25 \\ 0.75 \\ 1 \end{bmatrix}, \begin{bmatrix} 1.125 \\ 0.5 \\ 1.375 \end{bmatrix}, \begin{bmatrix} 1.25 \\ 0.6875 \\ 1.0625 \end{bmatrix}, \ldots, \begin{bmatrix} 1.2 \\ 0.6 \\ 1.2 \end{bmatrix}$$

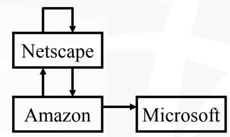
- Note that the sum of all PageRank’s is the total number of pages

Random surfer model

- A random surfer
  - Starts with a random page
  - Randomly selects a link on the page to visit next
  - Never uses the "back" button
- PageRank(p) measures the probability that a random surfer visits page p

Dead ends

- Pages with no outgoing links
- Called a rank leak because eventually all importance will "leak" out of the Web

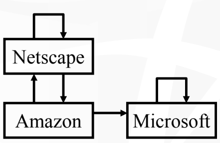


$$\begin{bmatrix} n \\ m \\ a \end{bmatrix} = \begin{bmatrix} 0.5 & 0 & 0.5 \\ 0 & 0 & 0.5 \\ 0.5 & 0 & 0 \end{bmatrix} \begin{bmatrix} n \\ m \\ a \end{bmatrix}$$

$$\begin{bmatrix} n \\ m \\ a \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0.5 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 0.75 \\ 0.25 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 0.625 \\ 0.25 \\ 0.375 \end{bmatrix}, \begin{bmatrix} 0.5 \\ 0.1875 \\ 0.3125 \end{bmatrix}, \ldots, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Spider traps

- A group of pages with no links out of the group
- Called a rank sink because this group will eventually accumulate all importance of the Web



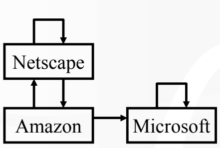
$$\begin{bmatrix} n \\ m \\ a \end{bmatrix} = \begin{bmatrix} 0.5 & 0 & 0.5 \\ 0 & 1 & 0.5 \\ 0.5 & 0 & 0 \end{bmatrix} \begin{bmatrix} n \\ m \\ a \end{bmatrix}$$

$$\begin{bmatrix} n \\ m \\ a \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1.5 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 0.75 \\ 1.75 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 0.625 \\ 2 \\ 0.375 \end{bmatrix}, \begin{bmatrix} 0.5 \\ 2.1875 \\ 0.3125 \end{bmatrix}, \ldots, \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix}$$

Practical PageRank

- d: decay factor
- PageRank(p) =
  $d \cdot \sum_{q \in B(p)} (\text{PageRank}(q) / N(q)) + (1 - d)$
- Intuition in the random surfer model
  - A surfer occasionally gets bored and jump to a random page on the Web instead of following a random link
- Another intuition
  - Make the graph fully connected

Practical PageRank example



$$\begin{bmatrix} n \\ m \\ a \end{bmatrix} = 0.8 \begin{bmatrix} 0.5 & 0 & 0.5 \\ 0 & 1 & 0.5 \\ 0.5 & 0 & 0 \end{bmatrix} \begin{bmatrix} n \\ m \\ a \end{bmatrix} + \begin{bmatrix} 0.2 \\ 0.2 \\ 0.2 \end{bmatrix}$$

$$\begin{bmatrix} n \\ m \\ a \end{bmatrix} = \ldots, \begin{bmatrix} 7/11 \\ 21/11 \\ 5/11 \end{bmatrix}$$

- Note that the sum of all PageRank's is still the total number of pages

HITS

- Ranking is computed over a focused subgraph of the Web that depends on the query
- Two scores instead of one
  - Authorities: pages that offer quality information about a topic, e.g., www.google.com
  - Hubs: pages that do not provide information directly but instead tell you where to find it, e.g., www.searchenginewatch.com

Identifying the focused subgraph

- Find R, the root set of t pages containing the query terms
- Start with base set $S = R$
- For each page p in R, add to S
  - All pages that p points to
    - Intuition: treat p as a hub, and add authorities
  - Up to d pages that point to p
    - Intuition: treat p as an authority, and add hubs; d is crucial in the case where an extremely popular page appears in R
- Focused subgraph is the graph induced by the base set, without the links that are used purely for navigation

Hub and authority scores

- $B(p)$: set of pages that point to p
- $F(p)$: set of pages that p points to
- Hub score vector: $[h_1, \ldots, h_n]$
- Authority score vector: $[a_1, \ldots, a_n]$
- $a_i = \sum_{j \in B(i)} h_j$, and $h_i = \sum_{j \in F(i)} a_j$

- Intuition
  - The score of a hub h is measured by the total score of the authorities that h points to
  - The score of an authority a is measured by the total score of the hubs that point to a

Computing hub and authority scores

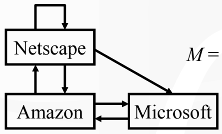
- Initialize $\vec{a}$ and $\vec{h}$ to arbitrary values
- Repeat until convergence:
  - $a_i = \sum_{j \in B(i)} h_j$
  - $h_i = \sum_{j \in F(i)} a_j$
  - Normalize so that $\sum_i a_i^2 = 1$ and $\sum_i h_i^2 = 1$

Definition in linear algebra

- Create a matrix M for the link structure
  - Each page i corresponds to row i and column i
  - $M(i, j) = 1$ if page i points to page j, or 0 otherwise
- $\vec{a} = \lambda\, M^T \vec{h}$, and $\vec{h} = \mu\, M\, \vec{a}$
- Therefore, $\vec{a} = \lambda\, \mu\, M^T M\, \vec{a}$, and $\vec{h} = \lambda\, \mu\, M M^T \vec{h}$

- Note the similarity to PageRank
- And the bug in Arasu et al.

HITS example



$$M = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} a_n \\ a_m \\ a_a \end{bmatrix} = \begin{bmatrix} 2 & 2 & 1 \\ 2 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} a_n \\ a_m \\ a_a \end{bmatrix}$$

$$\begin{bmatrix} h_n \\ h_m \\ h_a \end{bmatrix} = \begin{bmatrix} 3 & 1 & 2 \\ 1 & 1 & 0 \\ 2 & 0 & 2 \end{bmatrix} \begin{bmatrix} h_n \\ h_m \\ h_a \end{bmatrix}$$

$$\begin{bmatrix} a_n \\ a_m \\ a_a \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 5 \\ 5 \\ 4 \end{bmatrix}, \begin{bmatrix} 24 \\ 24 \\ 18 \end{bmatrix}, \begin{bmatrix} 114 \\ 114 \\ 84 \end{bmatrix}, \ldots \qquad \begin{bmatrix} h_n \\ h_m \\ h_a \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 6 \\ 2 \\ 4 \end{bmatrix}, \begin{bmatrix} 28 \\ 8 \\ 20 \end{bmatrix}, \begin{bmatrix} 132 \\ 36 \\ 96 \end{bmatrix}, \ldots$$

- We did not normalize here, but the ratios of scores still converge

Some other ranking ideas

- Include in the content of a page the texts surrounding links to this page
- Use HTML tags, e.g., title and emphasized words are considered more significant
- Use personal preferences, e.g., prefer bookmarked pages
- Prefer pages that are accessed more often
- Prefer sites that pay us more (Overture)