Search Engines

Technology and Algorithms for Efficient Searching of Information on the Web

Lecture’s Outline

- The Web and its Search Engines
- Heuristics-based Ranking
  - Page rank (Google)
    - for discovering the most "important" web pages
- HITS: hubs and authorities (Clever project)
  - more detailed evaluation of pages’ importance

The Web in 2001: Some Facts

- More than 3 billion pages; several terabytes
- Highly dynamic
  - More than 1 million new pages every day!
  - Over 600 GB of pages change per month
  - Average page changes in a few weeks
- Largest crawlers
  - Refresh less than 18% in a few weeks
  - Cover less than 50% ever
- Average page has 7-10 links
  - Links form content-based communities

Chaos on the Web

Internet lacks organization and structure:
- pages written in any language, dialect or style;
- different cultures, interests and motivation;
- mixes truth, falsehood, wisdom, propaganda…

Challenge:
- Quickly extract from this digital morass, high-quality, relevant, up-to-date pages in response to specific information needs
- No precise mathematical measure of "best" results

Search Products and Services (in 2000)

- Verity
- Fulcrum
- PLS
- Oracle text extender
- DB2 text extender
- Infoseek Intranet
- SMART (academic)
- Glimpse (academic)
- Inktomi (HotBot)
- Alta Vista
- Raging Search
- Google
- Dmoz.org
- Yahoo!
- Infoseek Internet
- Lycos
- Excite

Relevance Ranking

- Recall = coverage
  - What fraction of relevant documents were reported
- Precision = accuracy
  - What fraction of reported documents were relevant
- Trade-off
  - 'Query' generalizes to 'topic'

* heuristics-based
* humanly-selected
Heuristics-based Ranking

Naïve attempt used by many search engines.

Heuristics employed:
- number of times a page contains the query term
- favor instances where the term appears early
- give weight to word appearing in a special place or form; e.g., in a title or in bold.

All heuristics fail miserably due to:
- spamming, or
- polysemy and synonymy of natural language words

Hyperlink Graph Analysis

- Hypermedia is a social network
  - Telephoned, advised, co-authored, paid
- Social network theory (cf. Wasserman & Faust)
  - Extensive research applying graph notions
  - Centrality and prestige
  - Co-citation (relevance judgment)
- Applications
  - Web search: HITS, Google, CLEVER
  - Classification and topic distillation

Hypertext Models for Classification

- $c =$ class, $t =$ text, $n =$ number of neighbors
- Text-only model: $Pr(t | c)$
- Using neighbors’ text to judge my topic: $Pr(t, n(A) | c)$
- Better model: $Pr(t, c(N) | c)$
- Non-linear relaxation

Exploiting the Web’s Hyperlink Structure

Underlying assumption: view each link as an implicit endorsement of the location it points to

Page Rank

Intuition: Recursive Definition of “importance”.
A page is important if important pages link to it.

Method: Create a stochastic matrix of the Web
- each page corresponds to a matrix’s row and column
- if page $j$ has $n$ successors, then the $ij$-th entry is
  - $1/n$ if page $i$ is one of these $n$ successors of page $j$
  - 0 otherwise

Page Rank: Intuition

- Initially, each page has one unit of importance.
- At each round, each page shares whatever importance it has with its successors, and receives new importance from its predecessors.
- Eventually, the importance reaches a limit, which happens to be its component of the principal eigenvector of this matrix.

Importance = probability that a random Web surfer, starting from a random Web page, and following random links will be at the page in question after a long series of links.
Page Rank Example: The Web in 1689

Equation:
\[
\begin{bmatrix}
N \\
M \\
A
\end{bmatrix} = \begin{bmatrix}
1/2 & 0 & 1/2 \\
0 & 0 & 1/2 \\
1/2 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
N \\
M \\
A
\end{bmatrix}
\]

Solution by relaxation:
\[
N = 1 \quad 1 \quad 5/4 \quad 9/8 \quad 5/4 \quad \ldots \quad 6/5
\]
\[
M = 1 \quad 1/2 \quad 3/4 \quad 1/2 \quad 11/16 \quad \ldots \quad 3/5
\]
\[
A = 1 \quad 3/2 \quad 1 \quad 11/8 \quad 17/16 \quad \ldots \quad 6/5
\]

Problems with Real Web Graphs

Dead ends: a page that has no successors has nowhere to send its importance.
- Eventually, all importance will "leak out of" the Web.

Spider traps: a group of one or more pages that have no links outside the group.
- Eventually, these pages will accumulate all the importance of the Web.

Microsoft tries to duck monopoly charges...

Equation:
\[
\begin{bmatrix}
N \\
M \\
A
\end{bmatrix} = \begin{bmatrix}
1/2 & 0 & 1/2 \\
0 & 1 & 1/2 \\
1/2 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
N \\
M \\
A
\end{bmatrix}
\]

Solution by relaxation:
\[
N = 1 \quad 1 \quad 3/4 \quad 5/8 \quad 1/2 \quad \ldots \quad 0
\]
\[
M = 1 \quad 1/2 \quad 1/4 \quad 1/4 \quad 3/16 \quad \ldots \quad 0
\]
\[
A = 1 \quad 1/2 \quad 1/2 \quad 1/2 \quad 5/16 \quad \ldots \quad 0
\]

Google Solution to Dead Ends and Spider Traps

Instead of applying the matrix directly, "tax" each page with some fraction of its current importance, and distribute the taxed importance equally among all pages.
\[
\begin{bmatrix}
N \\
M \\
A
\end{bmatrix} = 0.8 \begin{bmatrix}
1/2 & 0 & 1/2 \\
0 & 1 & 1/2 \\
1/2 & 1/2 & 0
\end{bmatrix}
\begin{bmatrix}
N \\
M \\
A
\end{bmatrix} + 0.2 \begin{bmatrix}
N \\
M \\
A
\end{bmatrix}
\]

The solution to this equation is now
\[
N = 7/11; \quad M = 21/11; \quad A = 5/11
\]

Google Anti-Spam Devices

"Spamming": an attempt by many Web sites to appear to be about a subject that will attract surfers, without truly being about the subject

- Google, unlike other engines tends to believe what others say about a homepage in an incoming anchor text, making it harder for a homepage to appear to be about something it is not.
- The use of page rank to measure importance, rather than the more naive "number of links into a page", also protects against spamming. E.g., page rank recognizes as unimportant 1000 pages that mutually link to one another.
Google Facts (from end 2001)

- Indexes 3 billion Web pages
  - If printed, they would result in a stack of paper 200 km high
  - If a person reads a page per minute (and does nothing else), (s)he would need 6000 years to read them all
- 200 million search queries a day
  - Approx. 80 billion searches a year!
- Most searches take less than half second
- Support for 35 non-English languages
- Searchable index contains 3 trillion items
  - Updated every 28 days

Google Architecture (approx.)

1. The web server sends the query to the index servers.
   The content inside the index servers is similar to the index in the back of a book - it tells which pages contain the words that match the query.
2. The query travels to the doc servers, which actually retrieve the stored documents. Snippets are generated to describe each search result.
3. The search results are returned to the user typically in less than a second.

Hubs and Authorities

Defined in a mutually recursive way:

- a hub links to many (valuable) authorities;
- an authority is linked to by many (good) hubs.

Authorities turn out to be pages that offer the best information about a topic;
Hubs are pages that do not provide any information, but specify a collection of links on where to find the information.

Hubs and Authorities

Use a matrix formulation similar to that of Page rank, but without the stochastic restriction.
- Rows and columns correspond to Web pages:
  - \(A[i,j] = 1\), if page \(i\) links to page \(j\);
  - \(A[i,j] = 0\), otherwise
The transpose of \(A\) looks like the matrix for Page rank, but it has a 1 where the Page-rank matrix has fraction
Repeated application of the matrix leads to divergence. However, we can introduce scaling factors and keep the computed values of "authority" and "hubbiness" for each page within finite bounds.

Computing Hubbiness and Authority of Pages

Let \(a\) and \(h\) be vectors
- \(i\)-th component corresponds to the degrees of authority and hubbiness of the \(i\)-th page.
Let \(\lambda\) and \(\mu\) be suitable scaling factors.
Then:
- the hubbiness of each page is the sum of authorities it links to, scaled by \(\lambda\)
  \[ h = \lambda A a \]
- the authority of each page is the sum of the hubbiness of all pages that link to it, scaled by \(\mu\)
  \[ a = \mu A^T h \]

By simple substitution, two equations that relate vectors \(a\) and \(h\) only to themselves:

\[ a = \lambda \mu A A a \]
\[ h = \lambda A a \]

Thus, we can compute \(a\) and \(h\) by relaxation, giving us the principal eigenvectors of the matrices \(AA^T\) and \(A^TA\), respectively.
Netscape acknowledges Microsoft’s existence

Matrices:

\[ A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \]

\[ A^T = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \]

\[ AA = \begin{pmatrix} 3 & 1 & 2 \\ 1 & 1 & 0 \\ 2 & 0 & 2 \end{pmatrix} \]

\[ A^T A = \begin{pmatrix} 2 & 2 & 1 \\ 2 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} \]

Hubs and Authorities Example

Assuming \( \lambda = \mu = 1 \):

solution by relaxation:

\[ a(N) = 1 \quad 5 \quad 24 \quad 114 \quad ... \quad 1.3 \quad a(A) \]

\[ a(M) = 1 \quad 5 \quad 24 \quad 114 \quad ... \quad 1.3 \quad a(A) \]

\[ a(A) = 1 \quad 4 \quad 18 \quad 84 \quad ... \quad a(A) \]

\[ h(N) = 1 \quad 6 \quad 28 \quad 132 \]

\[ h(M) = 1 \quad 2 \quad 8 \quad 36 \]

\[ h(A) = 1 \quad 4 \quad 20 \quad 96 \]

Authorities and Hubs Pragmatics

- The system is jump-started by obtaining a set of root pages from a standard text index such as AltaVista.
- The iterative process settles very rapidly.
  - A root set of 3,000 pages requires just 5 rounds of calculations!
  - The results are independent of the initial estimates
- Algorithm naturally separates Web sites into clusters
  - e.g., a search for "abortion" partitions the Web into a pro-life and a pro-choice community

Ranking by popularity in Search Engines

- In-degree \( \approx \) prestige
- Not all votes are worth the same
- Prestige of a page is the sum of prestige of citing pages:
  \[ p = Ap \]
- Pre-compute query independent prestige score
- Google model
- High prestige \( \approx \) good authority
- High reflected prestige \( \approx \) good hub
- Bipartite iteration
  - \( a = Ah \)
  - \( h = A^T a \)
- HITS/Clever model

Philosophical Remarks

- The Web of today is dramatically different from what it was five years ago.
- Predicting the next five years seems futile.
  - Will even the basic act of indexing soon become infeasible?
  - If so, will our notion of searching the Web undergo fundamental changes?
- The Web’s relentless growth will continue to generate computational challenges for wading through the ever increasing volume of on-line information.