Chapter 21
Link analysis
Content

- Anchor text
- Link analysis for ranking
  - Pagerank and variants
  - HITS
Assumption 1: a hyperlink is a quality signal
  • A hyperlink between pages denotes author perceived relevance

Assumption 2: The anchor text describes the target page
  • we use anchor text somewhat loosely here
  • extended anchor text, window of text surrounding anchor text
  • You can find cheap cars <a href= ...>here</a>
### [document text only] vs. [document text + anchor text]

- Searching on [document text + anchor text] is often more effective than searching on [document text only].
- **Example: Query** *IBM*
  - Matches IBM’s copyright page
  - Matches many spam pages
  - Matches IBM wikipedia article
  - May not match IBM home page! (if IBM home page is mostly graphical)
- Searching on anchor text is better for the query *IBM*.
- Represent each page by all the anchor text pointing to it.
- In this representation, the page with the most occurrences of *IBM* is [www.ibm.com](http://www.ibm.com).
Anchor text containing *IBM* pointing to www.ibm.com

www.nytimes.com: “IBM acquires Webify”

www.slashdot.org: “New IBM optical chip”

www.stanford.edu: “IBM faculty award recipients”

www.ibm.com
Indexing anchor text

- Thus: anchor text is often a better description of a page’s content than the page itself.
- Anchor text can be weighted more highly than document text (based on Assumptions 1 & 2).
- When indexing a document $D$, include anchor text from links pointing to $D$.

Armonk, NY-based computer giant IBM announced today

Joe’s computer hardware links
Sun
HP
IBM

www.ibm.com

Big Blue today announced record profits for the quarter
Google bombs

- Indexing anchor text can have unexpected side effects: Google bombs.
  - what else does not have side effects?
- A Google bomb is a search with “bad” results due to maliciously manipulated anchor text
- Google introduced a new weighting function in January 2007 that fixed many Google bombs
Google bomb example

Biography of President George W. Bush
Biography of the president from the official White House web site.
www.whitehouse.gov/president/gwbbio.html - 29k - Cached - Similar pages
Past Presidents - Kids Only - Current News - President
More results from www.whitehouse.gov »

Welcome to MichaelMoore.com!
Official site of the gadfly of corporations, creator of the film Roger and Me and the television show The Awful Truth. Includes mailing list, message board, ...
www.michaelmoore.com/ - 35k - Sep 1, 2005 - Cached - Similar pages

BBC NEWS | Americas | 'Miserable failure' links to Bush
Web users manipulate a popular search engine so an unflattering description leads to the president's page.
news.bbc.co.uk/2/hi/americas/3298443.stm - 31k - Cached - Similar pages

Google's (and Inktomi's) Miserable Failure
A search for miserable failure on Google brings up the official George W. Bush biography from the US White House web site. Dismissed by Google as not a ...
searchenginewatch.com/sereport/article.php/3296101 - 45k - Sep 1, 2005 - Cached - Similar pages
Origins of PageRank: Citation analysis (1)

- Citation analysis: analysis of citations in the scientific literature
- Example citation: “Miller (2001) has shown that physical activity alters the metabolism of estrogens.”
- “Miller (2001)” is a hyperlink linking two scientific articles.
- One application of these “hyperlinks” in the scientific literature:
  - Measure the similarity of two articles by the overlap of other articles citing them.
  - This is called cocitation similarity.
- Cocitation similarity on the web?
Origins of PageRank: Citation analysis (2)

- Citation frequency can be used to measure the impact of an article.
  - Each article gets one vote.
  - Not a very accurate measure.
- Better measure: weighted citation frequency / citation rank
  - An article’s vote is weighted according to its citation impact.
  - Circular? No: can be formalized in a well-defined way.
  - This is basically PageRank.
  - PageRank was invented in the context of citation analysis by Pinsker and Narin in the 1960s.
Query-independent ordering

- First generation link-based ranking for web search
  - using link counts as simple measures of popularity.
  - simple link popularity: number of in-links

  - First, retrieve all pages meeting the text query (say *venture capital*).
  - Then, order these by the simple link popularity

- Easy to spam. Why?
Basic for Pagerank: random walk

- Imagine a web surfer doing a random walk on the web page:
  - start at a random page
  - at each step, go out of the current page along one of the links on that page, equiprobably
- In the steady state each page has a long-term visit rate - use this as the page’s score
- So, pagerank = steady state probability = long-term visit rate
Not quite enough

- The web is full of dead-ends
  - random walk can get stuck in dead-ends
  - makes no sense to talk about long-term visit rates
Teleporting

- Teleport operation: surfer jumps from a node to any other node in the web graph, chosen uniformly at random from all web pages.

- Used in two ways:
  - At a dead end, jump to a random web page.
  - At any non-dead end, with teleportation probability $0 < \alpha < 1$ (say, $\alpha = 0.1$), jump to a random web page; with remaining probability $1 - \alpha$ (0.9), go out on a random link.

- Now cannot get stuck locally.

- There is a long-term rate at which any page is visited:
  - Not obvious, explain later.
  - How do we compute this visit rate?
Markov chains

- A Markov chain consists of $n$ states, plus an $n \times n$ transition probability matrix $P$.
- At each step, we are in exactly one of the states.
- For $1 \leq i, j \leq n$, the matrix entry $P_{ij}$ tells us the probability of $j$ being the next state, given we are currently in state $i$.

- Clearly, for each $i$, $\sum_{j=1}^{n} P_{ij} = 1$.
- Markov chains are abstractions of random walk
  - State = page
Exercise

Represent the teleporting random walk as a Markov chain, for the following case, using transition probability matrix

\[ \alpha = 0.3: \]

\[
\begin{pmatrix}
0.1 & 0.45 & 0.45 \\
0.45 & 0.45 & 0.1 \\
0.45 & 0.45 & 0.1 \\
\end{pmatrix}
\]
Ergodic Markov chains

- A Markov chain is **ergodic** iff it’s irreducible and aperiodic
  - Irreducibility: roughly, there is a path from any state to any other
  - Aperiodicity: roughly, the nodes cannot be partitioned such that the random walker visits the partitions sequentially
- A non-ergodic Markov chain
Ergodic Markov chains

- Theorem: For any ergodic Markov chain, there is a unique long-term visit rate for each state.
  - Steady-state probability distribution.

- Over a long time-period, we visit each state in proportion to this rate.

- It doesn’t matter where we start.
Formalization of visit: probability vector

- A probability (row) vector $\mathbf{x} = (x_1, \ldots, x_n)$ tells us where the walk is at any point.
- e.g., (000...1...000) means we’re in state $i$

More generally, the vector $\mathbf{x} = (x_1, \ldots, x_n)$ means the walk is in state $i$ with probability $x_i$

$$\sum_{i=1}^{n} x_i = 1$$
If the probability vector is \( \mathbf{x} = (x_1, \ldots, x_n) \) at this step, what is it at the next step?

Recall that row \( i \) of the transition probability matrix \( \mathbf{P} \) tells us where we go next from state \( i \).

So from \( \mathbf{x} \), our next state is distributed as \( \mathbf{xP} \).
The steady state is simply a vector of probabilities \( \mathbf{a} = (a_1, \ldots, a_n) \):

- \( a_i \) is the probability that we are in state \( i \)
- \( a_i \) is the long-term visit rate (or pagerank) of state (page) \( i \)
- so we can think of pagerank as a long vector, one entry for each page
How do we compute this vector?

- Let $a = (a_1, \ldots, a_n)$ denote the row vector of steady-state probabilities.
- If our current position is described by $a$, then the next step is distributed as $aP$.
- But $a$ is the steady state, so $a = aP$.
- Solving this matrix equation gives us $a$.
  - so $a$ is the (left) eigenvector for $P$.
  - corresponds to the principal eigenvector of $P$ with the largest eigenvalue.
  - transition probability matrices always have largest eigenvalue 1.
One way of computing

- Recall, regardless of where we start, we eventually reach the steady state $a$.
- Start with any distribution (say $x=(10\ldots0)$).
- After one step, we’re at $xP$.
- After two steps at $xP^2$, then $xP^3$ and so on.
- “Eventually” means for “large” $k$, $xP^k = a$.
- Algorithm: multiply $x$ by increasing powers of $P$ until the product looks stable.
- This is called the power method.
Power method: Example

- Two-node example: \( \bar{x} = (0.5, 0.5) \), \( P = \begin{pmatrix} 0.25 & 0.75 \\ 0.25 & 0.75 \end{pmatrix} \)
- \( \bar{x}P = (0.25, 0.75) \)
- \( \bar{x}P^2 = (0.25, 0.75) \)
- Convergence in one iteration!
Pagerank summary

- **Preprocessing:**
  - Given graph of links, build transition probability matrix $P$
  - From it compute $a$
  - The entry $a_i$ is a number between 0 and 1: the pagerank of page $i$.

- **Query processing:**
  - Retrieve pages meeting query
  - Rank them by their pagerank
  - Order is query-\textit{independent}

- **In practice, pagerank alone wouldn’t work**

- Google paper: http://infolab.stanford.edu/~backrub/google.html
In practice

Consider the query “video service”
- Yahoo! has very high pagerank, and contains both words
- With simple pagerank alone, Yahoo! Would be top-ranked
- Clearly not desirable

In practice, composite score is used in ranking
- Pagerank, cosine similarity, term proximity etc.
- May apply machine-learned scoring
- Many other clever heuristics are used
How important is PageRank?

- Frequent claim: PageRank is the most important component of web ranking.
- The reality:
  - There are several components that are at least as important: e.g., anchor text, phrases, proximity, tiered indexes . . .
  - Rumor has it that PageRank in its original form (as presented here) has a negligible impact on ranking!
  - However, variants of a page’s PageRank are still an essential part of ranking.
  - Addressing link spam is difficult and crucial.
Pagerank: Issues and Variants

- How realistic is the random surfer model?
  - What if we modeled the back button?
  - Surfer behavior sharply skewed towards short paths
  - Search engines, bookmarks & directories make jumps non-random.

- Biased Surfer Models
  - Weight edge traversal probabilities based on match with topic/query (non-uniform edge selection)
  - Bias jumps to pages on topic (e.g., based on personal bookmarks & categories of interest)

- Non-uniform teleportation allows topic-specific pagerank and personalized pagerank
Conceptually, we use a random surfer who teleports, with say 10% probability, using the following rule:

- Selects a category (say, one of the 16 top level ODP categories) based on a query & user-specific distribution over the categories
- Teleport to a page uniformly at random within the chosen category
A person is reputable if s/he receives many references from reputable people.

How to compute reputation for people?
Hyperlink-Induced Topic Search (HITS)

- In response to a query, instead of an ordered list of pages each meeting the query, find two sets of inter-related pages:
  - *Hub pages* are good lists of links to pages answering the information need
    - e.g., “Bob’s list of cancer-related links
  - *Authority pages* are direct answers to the information need
    - occur recurrently on good hubs for the subject
- Most approaches to search do not make the distinction between the two sets
Hubs and Authorities

- Thus, a good hub page for a topic *points* to many authoritative pages for that topic
- A good authority page for a topic is *pointed* to by many good hubs for that topic
- Circular definition - will turn this into an iterative computation
Examples of hubs and authorities

Long distance telephone companies

Hubs

Alice

Bob

AT&T

Sprint

MCI

Authorities
High-level scheme

- Do a regular web search first
- Call the search results the root set
- Add in any page that either
  - points to a page in the root set, or
  - is pointed to by a page in the root set
- Call this the base set
- From these, identify a small set of top hub and authority pages
- Iterative algorithm
Assembling the base set

- Root set typically 200-1000 nodes
- **Base set may have up to 5000 nodes**
- How do you find the base set nodes?
  - Follow out-links by parsing root set pages
  - Get in-links from a *connectivity server*, get pages
    - This assumes our inverted index supports searches for links, in addition to terms
Distilling hubs and authorities

- Compute, for each page $x$ in the base set, a **hub score** $h(x)$ and an **authority score** $a(x)$.
- Initialize: for all $x$, $h(x) \leftarrow 1$; $a(x) \leftarrow 1$;
- Iteratively update all $h(x)$, $a(x)$;
- **After convergence**
  - output pages with highest $h()$ scores as top hubs
  - output pages with highest $a()$ scores as top authorities
  - so we output two ranked lists
Iterative update

- Iterate these two steps until convergence

for all $x$:

$$h(x) \leftarrow \sum_{x \mapsto y} a(y)$$

for all $x$:

$$a(x) \leftarrow \sum_{y \mapsto x} h(y)$$
Scaling

- To prevent the $h()$ and $a()$ values from getting too big, can scale down after each iteration.
- Scaling factor doesn’t really matter:
  - we only care about the relative values of the scores.
How many iterations?

- Relative values of scores will converge after a few iterations.
- In fact, suitably scaled, $h()$ and $a()$ scores settle into a steady state!
  - proof of this comes later.

- In practice, ~5 iterations get you close to stability.
Japan Elementary Schools

Hubs
- schools
- LINK Page-13
- “ú–«šw Z
- a‰o„ ¬šw Zfz [f fy fW
- 100 Schools Home Pages (English)
- K-12 from Japan 10/...rnet and Education)
- http://www...iglobe.ne.jp/~IKESAN
- ,l,fj ¬šw Z,U”N,P ‘g•Œê
- ÔŠ—’ —§ ÔŠ—”Œ ¬šw Z
- Koulutus ja oppilaitokset
- TOYODA HOMEPAGE
- Education
- Cay's Homepage(Japanese)
- ¬y”i ¬šw ZƗfz [f fy [fW
- UNIVERSITY
- %‰J—³ ¬šw Z DRAGON97-TOP
- Â‰u¬šw Z,T”N,P ‘gfz [f fy [fW
- µé¾Å© ¥å¥ï ¥å¥ï ¥å¥ï ¥å¥ï

Authorities
- The American School in Japan
- The Link Page
- %‰ è s—§”ä“c ¬šw Zfz [f fy fW
- Kids' Space
- „Â è s—§”Â è ¼” ¬šw Z
- ‹{ év3”ç ’ašw• ‘@ ¬šw Z
- KEIMEI GAKUEN Home Page (Japanese)
- Shiranuma Home Page
- fuzoku-es.fukui-u.ac.jp
- welcome to Miasa E&J school
- _“p iŒ§ E‰i•l s—§’ ‹ i ¼ ¬šw Z,Ïfy
- http://www...p/~m_maru/index.html
- fukui haruyama-es HomePage
- Torisu primary school
- goo
- Yakumo Elementary,Hokkaido,Japan
- FUZOKU Home Page
- Kamishibun Elementary School...
Things to note

- Pulled together good pages regardless of language of page content
- Use *only* link analysis after base set assembled
  - iterative scoring is query-independent
- Iterative computation after text index retrieval - significant overhead
The PageRank and HITS make two different design choices concerning (i) the eigenproblem formalization (ii) the set of pages to apply the formalization to. These two are orthogonal.

- We could also apply HITS to the entire web and PageRank to a small base set.
- On the web, a good hub almost always is also a good authority.
- The actual difference between PageRank ranking and HITS ranking is therefore not as large as one might expect.
Researchers publish/present papers in conferences. A conference is reputable if it hosts many reputable researchers to publish/present their papers. A researcher is reputable if s/he publishes/presents many papers in reputable conferences.

How to compute reputation for conferences? How to compute reputation for researchers?
Proof of convergence

- $n \times n$ adjacency matrix $A$:
  - each of the $n$ pages in the base set has a row and column in the matrix.
  - Entry $A_{ij} = 1$ if page $i$ links to page $j$, else $= 0$. 
Hub/authority vectors

- View the hub scores \( h() \) and the authority scores \( a() \) as vectors with \( n \) components.
- Recall the iterative updates

\[
\begin{align*}
    h(x) & \leftarrow \sum_{y \leftarrow x} a(y) \\
    a(x) & \leftarrow \sum_{y \rightarrow x} h(y)
\end{align*}
\]
Rewrite in matrix form

- $h = Aa$.
- $a = A^t h$.

Recall $A^t$ is the transpose of $A$.

Substituting, $h = AA^t h$ and $a = A^t A a$.

Thus, $h$ is an eigenvector of $AA^t$ and $a$ is an eigenvector of $A^t A$.

Further, our algorithm is a particular, known algorithm for computing eigenvectors: the *power iteration* method.

Guaranteed to converge.