Information Retrieval

Lecture 4: Search engines and linkage algorithms

Computer Science Tripos Part II

UNIVERSITY OF CAMBRIDGE

Simone Teufel

Natural Language and Information Processing (NLIP) Group

sht25@cl.cam.ac.uk

Today

- Fixed document collections → World Wide Web:
  What are the differences?
- Linkage-based algorithms
  - PageRank (Brin and Page, 1998)
  - HITS (Kleinberg, 1998)

Differences closed-world/web: data on web is...

- Large-volume
  - Estimates of 80 billion pages for 2006 (1600 TB) (1TB = 1024 GB = 2^{40}B)
  - Google indexed 8 billion pages in 2004; coverage 15-20% of web
  - Size of the web is doubling every half a year (Lawrence and Giles, "Searching the world wide web", Science, 1998)
- Redundant (copied or dynamically generated)
- Unstructured/differently structured documents
- Heterogenous (length, quality, language, contents)
- Volatile/dynamic
  - 1 M new pages per day; average page changes every 2-3 weeks
  - 2-9% of indexed pages are invalid
- Hyperlinked

Differences closed-world/web: search algorithms

- Different syntactic features in query languages
  - Ranked with proximity, phrase units, order relevant, with or without stemming
- Different indexing ("web-crawling")
  - Heuristic enterprise; not all pages are indexed (est. 15-20% (2005); 28-55% (1999) of web covered)
- Different heuristics used (in addition to standard IR measures)
  - Proximity and location of search terms (Google)
  - Length of URL (AltaVista)
  - Anchor text pointing to a page (Google)
  - Quality estimates based on link structure

Web Crawling

- At search time, browsers do not access full text
- Index is built off-line; crawlers/spiders find web pages
  - Start with popular URLs and recursively follow links
  - Send new/updated pages to server for indexing
  - Search strategy: breadth-first, depth-first, backlink count, estimated popularity
- Parallel crawling
  - Avoid visiting the same page more than once
  - Partition the web and explore each partition exhaustively
- Agreement `robots.txt`: directories off-limits for crawlers
- In 1998, Google processed 4 M pages/day (50 pages, 500 links per second); fastest crawlers today: 10 M pages/day
- In 1998, AltaVista used 20 processors with 130G RAM and 500 GB disk each for indexing.

Link structure as a quality measure

- Links contain valuable information: latent human judgement
- Idea: derive quality measure by counting links
- Cf. citation index in science: papers which are cited more are considered to be of higher quality
- Similarity to scientific citation network
  - Receiving a “backlink” is like being cited (practical caveat: on the web, there is no certainty about the number of backlinks)

Simple backlink counting

Suggestion: of all pages containing the search string, return the pages with the most backlinks

- Generalisation problem
  - Many pages are not sufficiently self-descriptive
  - Example: the term "car manufacturer" does not occur anywhere on Honda homepage
  - No endogenous information (ie. information found in the page itself, rather than elsewhere) will help
- Page quality not considered at all, only raw backlink number
  - Overall popular page (Yahoo, Amazon) would be wrongly considered an expert on every string it contains
  - A page pointed to by an important page is also important (even if it has only that one single backlink)
  - Possible to manipulate this measure

Additional problem: manipulatability

- Web links are not quite like scientific citations
  - Large variation in web pages: quality, purpose, number of links, length (scientific articles are more homogeneous)
    - ∗ No publishing/production costs associated with web sites
    - ∗ No quality check (cf. peer review in scientific articles)
    - ∗ No cost associated with links (cf. length restrictions in scientific articles)
  - Therefore, linking is gratuitous (replicable), whereas citing is not
  - Any quality evaluation strategy which counts replicable features of web pages is prone to manipulation
- Therefore, raw counting will work less well than it does in scientific area
- Must be more clever when using link structure: PageRank, HITS

PageRank (Brin and Page, 1998)

- L. Page et al: “The PageRank Citation Ranking: Bringing order to the web”, Tech Report, Stanford Univ., 1998
- S. Brin, L. Page: “The anatomy of a large-scale Hypertextual Web Search Engine”, WWW7/Computer Networks 30(1-7):107-117, 1998
- Goal: estimate overall relative importance of web pages
- Simulation of a random surfer
  - Given a random page, follows links for a while (randomly), with probability q — assumption: never go back on already traversed links
  - Gets bored after a while and jumps to the next random page, with probability $1 - q$
  - Surfs infinitely long
- PageRank is the number of visits to each page

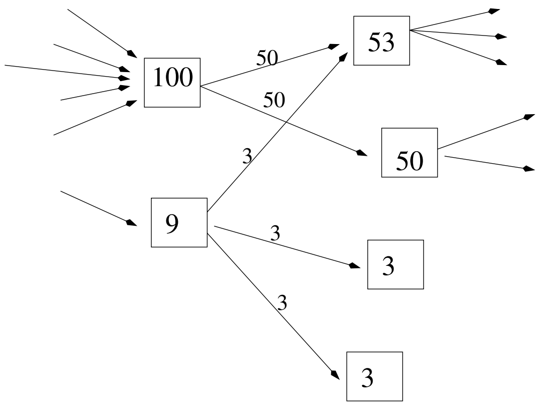
PageRank formula (simple case)

$$R(u) = (1-q) + q \sum_{v \in B_u} \frac{R(v)}{N_v}$$

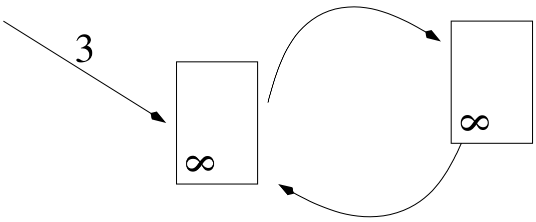
u	a web page
F_u	set of pages u points to (“Forward” set)
B_u	set of pages that point to u
$N_u = \lvert F_u \rvert$	number of pages u points to
q	probability of staying locally on page

This formula assumes that no PageRank gets lost in any iteration. In order for this to be the case, each page must have at least one outgoing link.

Simplified PageRank (q=1.0):

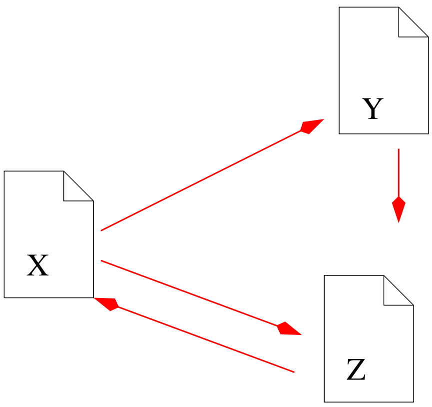


Rank sinks and rank sources



- The amount of pagerank in the web should be equal to N (so that the average page rank on the web is 1)
- Rank must stay constant in each step, but rank sinks lose infinitely much rank
- Rank also gets lost in each step for pages without onward links
- Solution: rank source $\vec{e}$ counteracts rank sinks
- $\vec{e}$ is the vector of the probability of random jumps of random surfer to a random page

An example: PageRank computation



$$R(u) = (1-q) + q \sum_{v \in B_u} \frac{R(v)}{N_v}$$

This assumes that all R(v)s are from the previous iteration.

Pagerank for the “mini-web” (q=.85)

Iteration	PR(X)	PR(Y)	PR(Z)	$\sum$(PR(i))	Iteration	PR(X)	PR(Y)	PR(Z)	$\sum$(PR(i))
1	1.00000	1.000000	1.00000	3.00000	1	0.00000	0.00000	0.00000	0.00000
2	1.00000	0.575000	1.06375	2.63875	2	0.15000	0.21375	0.39543	0.75918
3	1.05418	0.598029	1.10635	2.75857	3	0.48612	0.35660	1.50243	1.50243
4	1.09040	0.613420	1.13482	2.83865	4	0.71075	0.45203	0.83633	1.99915
5	1.11460	0.623706	1.15385	2.89216	5	0.86088	0.51587	0.95436	2.33112
6	1.13077	0.630581	1.16657	2.92793	6	0.96121	0.55853	1.03325	2.55298
7	1.14158	0.635175	1.17507	2.95183	7	1.02826	0.58701	1.08597	2.70125
8	1.14881	0.638245	1.18075	2.96781	8	1.07307	0.60605	1.12120	2.80034
9	1.15363	0.640292	1.18454	2.97846	9	1.10302	0.61878	1.14475	2.86656
	...	...	...	...		...	...	...	...
82	1.16336	0.64443	1.19219	2.99999	86	1.16336	0.64443	1.19219	2.99999
83	1.16336	0.64443	1.19219	3.00000	87	1.16336	0.64443	1.19219	3.00000

Matrix notation of PageRank

$$\vec{r} = c(qA\vec{r} + (1-q)m\vec{1})$$

such that c is maximised and $||\vec{r}||_1 = 1$. ($||\vec{r}||_1$ is the L_1 norm of $\vec{r}$).

$$\vec{r} = c(qA + \frac{1-q}{N}\mathbf{1})\vec{r}$$

A normalised link matrix of the web:
$$A_{uv} = \begin{cases} \frac{1}{N_v} & if\ \exists v \rightarrow u \\ 0 & otherwise \end{cases}$$

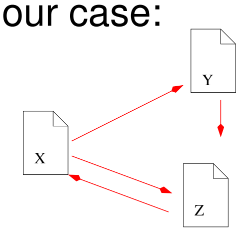
$\vec{r}$ PageRank vector (over all web pages), the desired result.
$\vec{1}$ a column vector consisting only of ones
$\mathbf{1}$ a matrix filled with all ones
m average pagerank per page (e.g., 1).

We know from linear algebra that $\vec{r} := A\vec{r}$; normalise ($\vec{r}$); $\vec{r} := A\vec{r}$ ... will make $\vec{r}$ converge to the dominant eigenvector of A (independently of $\vec{r}$’s initial value), with eigenvalue c.

Pagerank, matrix algorithm

1. Initialise $\vec{r}$, A
2. Loop:
   - $\vec{r} = c(qA + \frac{1-q}{N}\mathbf{1})\vec{r}$
   - Stop criterion: $||\vec{r_{i+1}} - \vec{r_i}|| < N\epsilon$ ($||\vec{r_{i+1}} - \vec{r_i}||$ is page-wise "movement" in PageRank between two iterations)
   - This will result in a Page rank vector $\vec{r}$ whose average PageRank per page is 1: $||\vec{r}_{i+1}||_1 = N$

In our case:



A= $\begin{vmatrix} 0 & 0 & 1 \\ .5 & 0 & 0 \\ .5 & 1 & 0 \end{vmatrix}$

$$\vec{r_0} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}; \; q = .85; \; B = qA + \frac{1-q}{N}\mathbf{1}$$

B = $\begin{vmatrix} .050 & .050 & .900 \\ .475 & .050 & .050 \\ .475 & .900 & .050 \end{vmatrix}$

Now iterate { $\vec{r_n}$ = B$\vec{r_{n-1}}$; normalise $\vec{r_n}$}

Iterative matrix-based PageRank computation

B = $\begin{vmatrix} .050 & .050 & .900 \\ .475 & .050 & .050 \\ .475 & .900 & .050 \end{vmatrix}$

Iterate $\vec{r_n} = B\vec{r_{n-1}}$:

$$\vec{r_0} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}; \; \vec{r_1} = \begin{bmatrix} 1.0000 \\ 0.5750 \\ 1.4250 \end{bmatrix}; \; \vec{r_2} = \begin{bmatrix} 1.3613 \\ 0.5750 \\ 1.0637 \end{bmatrix}; \; \vec{r_3} = \begin{bmatrix} 1.0542 \\ 0.7285 \\ 1.2173 \end{bmatrix}; \; \vec{r_4} = \begin{bmatrix} 1.1847 \\ 0.5980 \\ 1.2173 \end{bmatrix}; \; \vec{r_5} = \begin{bmatrix} 1.1847 \\ 0.6535 \\ 1.1618 \end{bmatrix};$$

$$\vec{r_6} = \begin{bmatrix} 1.1375 \\ 0.6535 \\ 1.2090 \end{bmatrix}; \; \vec{r_7} = \begin{bmatrix} 1.1776 \\ 0.6335 \\ 1.1889 \end{bmatrix}; \; \vec{r_8} = \begin{bmatrix} 1.1606 \\ 0.6505 \\ 1.1889 \end{bmatrix}; \; \vec{r_9} = \begin{bmatrix} 1.1606 \\ 0.6432 \\ 1.1962 \end{bmatrix}; \; \vec{r_{10}} = \begin{bmatrix} 1.1667 \\ 0.6432 \\ 1.1900 \end{bmatrix} \ldots$$

PageRank computation (practicalities)

- Space
  - Example: 75 M unique links on 25 M pages
  - Then: memory for PageRank 300MB
  - Link structures is compact (8B/link compressed)
- Time
  - Each iteration takes 6 minutes (for the 75 M links)
  - Whole process: 5 hours
  - Convergence after 52 iter. (322M links), 48 iter. (161M links)
  - Scaling factor linear in log n
- Pages without children removed during iteration
- Raw data can be obtained during web crawl; cost of computing PageRank is insignificant compared to the cost of building a full index

PageRank versus usage data

- Difference between linking behaviour (public) and actual usage data (web page access numbers from NLANR)
  - PageRank uses only public information; thus fewer privacy implications than usage data (pages that are accessed but not linked to)
  - PageRank produces a finer resolution compared to small usage sample
  - But: not all web users create links
- Propagation simulates word-of-mouth effects in complex network (ahead of time):
  - PageRank can change fast (one link on Yahoo)
    - Good pages often have only a few important backlinks (at first)
    - Those pages would not be found by simply back-link counting
  - Net traffic can change fast (one mention on the radio)

Summary PageRank

- Model of collaborative trust; users want information from “trusted” sources
- PageRank is immune to manipulation: it must convince an important site, or many unimportant ones, to point to it
  - Spamming PageRank costs real money – a good property for a search algorithm
  - Google’s business model: never sell PageRank (only advertising space)
- PageRank is a good predictor of optimal crawling order

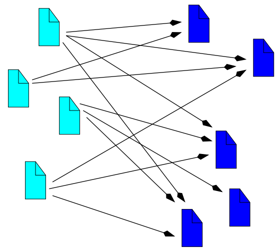
Top 15 PageRanks in July 1996

Download Netscape Software	11589.00
http://www.w3.org	10717.70
Welcome to Netscape	8673.51
Point: It’s what you’re searching for	7930.92
Web-Counter home page	7254.97
THe Blue Ribbon Campaign for Online Free Speech	7010.39
CERN Welcome	6562.49
Yahoo!	6561.80
Welcome to Netscape	6203.47
Wusage 4.1: A Usage Statistics System for Web Servers	5963.27
The World Wide Web consortium (W3C)	5672.21
Lycos, Inc. Home Page	4683.31
Starting Point	4501.98
Welcome to Magellan!	3866.62
Oracle Corporation	3587.63

Benefits for search with PageRank are greatest for underspecified queries

Hypertext Induced Topic Search (HITS)

- J. Kleinberg, “Authoritative sources in a hyperlinked environment”, ACM-SIAM 1998
- Goal: find authorities on a certain topic (relevance, popularity)
- Idea: There are hubs and authorities on the web, which exhibit a mutually reinforcing relationship
- Hubs: Recommendation pages with links to high-quality pages (authorities), e.g. compilations of favourite bookmarks, “useful links”
- Authorities: Pages that are recognised by others (particularly by hubs!) as experts on a certain topic

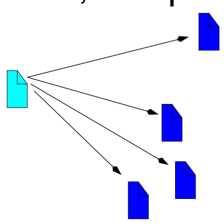


- Authorities are different from universally popular pages (high backlink count), which are not particular experts on that topic

HITS

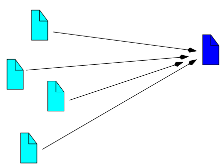
- Each page has two non-negative weights: an authority weight a and a hub weight h
- At each iteration, update the weights:
  - If a page points at many good authorities, it is probably a good hub:

$$h_p = \sum_{q:<p,q>\in A} a_q$$



  - If a page is pointed to by many good hubs, it is probably a good authority:

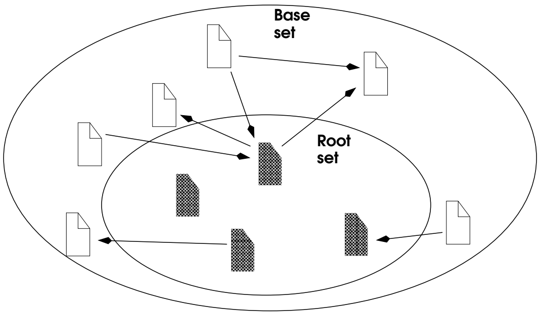
$$a_p = \sum_{q:<q,p>\in A} h_q$$



- Normalise weights after each iteration

HITS: Initialisation

- Start with the root set: set of web pages containing the query terms
- Create the base set: root set plus all pages pointing to the root set (cut-off if too many), and being pointed to by the root set
- The base set typically contains 1000-5000 documents



HITS: Algorithm

Given:

- a set $D = \{D_1 \dots D_n\}$ of documents (base set)
- A, the linking matrix: edge $< i, j > \in A$ iff D_i points to D_j
- k, the number of desired iterations

Initialise: $\vec{a}$ = {1, 1,. . . , 1}; $\vec{h}$ = {1, 1, . . . , 1}
Iterate: for $c = 1 \dots k$

- for $i = 1 \dots n : a_p = \Sigma_{q:<q,p>\in A} h_q$
- for $i = 1 \dots n : h_p = \Sigma_{q:<p,q>\in A} a_q$

Normalise $\vec{a}$ and $\vec{h}$: $\Sigma_{i \in D_i} a_i = \Sigma_{i \in D_i} h_i = 1$

HITS: Convergence

- Updates:

$$\vec{a} = A^T \vec{h} \qquad \vec{h} = A\vec{a}$$

- After the first iteration:

$$\vec{a}_1 = A^T A \vec{a}_0 = (A^T A)\vec{a}_0 \qquad \vec{h}_1 = A A^T \vec{h}_0 = (A A^T)\vec{h}_0$$

- After the second iteration:

$$\vec{a}_2 = (A^T A)^2 \vec{a}_0 \qquad \vec{h}_2 = (A A^T)^2 \vec{h}_0$$

- Convergence to
  - $\vec{a} \leftarrow$ dominant eigenvector$(A^T A)$
  - $\vec{h} \leftarrow$ dominant eigenvector$(A A^T)$

HITS: Example results

Authorities on "java"

0.328	`http://www.gamelan.com`	Gamelan
0.251	`http://java.sun.com`	JavaSoft home page
0.190	`http://www.digitalfocus.com/digital`	The Java Developer: How do I

Authorities on "censorship"

0.376	`http://www.eff.org`	EFF – The Electronic Frontier Fountation
0.344	`http://www.eff.org/blueribbon.html`	The Blue Ribbon Campaign for Online Free Speech
0.238	`http://www.cdt.org`	The Center for Democracy and Technology
0.235	`http://www.vtw.org`	Voters Telecommunication Watch
0.218	`http://www.aclu.org`	ACLU: American Civil Liberties Union

Authorities on "search engine"

0.346	`http://www.yahoo.com`	Yahoo
0.291	`http://www.excite.com`	Excite
0.239	`http://www.mckinley.com`	Welcome to Magellan
0.231	`http://www.lycos.com`	Lycos Home Page
0.231	`http://www.altavista.digital.com`	AltaVista: Main Page

- Both HITS and PageRank infer quality/“expert-ness” from link structure of the web
- Link structure contains latent human judgement
- Use different models of type of web pages
- Iterative algorithms
- Use of these weights for search (in different ways)
- Other differences between closed-world assumption (IR) and world wide web: data, indexing, query constructs, search heuristics