Information Retrieval

Lecture 4: Search engines and linkage algorithms

Computer Science Tripos Part II
Lent Term 2004

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)

Data on the web is

- Large-volume
  - Estimates of 10–20 billion pages for 2003 (300 TB) (1TB = 1024 GB = 2^{43}B)
  - Size of the web is doubling every half a year (Lawrence and Giles, “Searching the world wide web”, Science, 1998)
- Redundant
- 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. 28-55% of web covered)
- Different heuristics used (in addition to standard IR measures)
  - Heuristics:
    - ∗ Proximity of search terms (Google)
    - ∗ Length of URL (AltaVista)
    - ∗ Anchor text pointing to a page (Google)
  - Quality estimates based on link structure

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

Possible search heuristics: term frequency

Suggestion 1: of all pages containing the search string, return the pages with highest term frequency

- Generalisation problem
  - Many pages are not sufficiently self-descriptive; super types are rarely explicitly given
  - Example: Honda homepage assumes you know Honda is a car manufacturer; the term “car manufacturer” does not occur anywhere on this page
  - No endogenous information (ie. information found in the page itself, rather than elsewhere) will help here
- Quality of pages is not considered at all

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

- Generalisation problem (cf. above)
- Too much importance to raw number of backlinks
  - Overall popular page (Yahoo, Amazon) would be considered an authority on every string it contains
- Intuition about importance of links is ignored
  - A page pointed to by an important page is by definition also an important page, even if it has only that one single backlink
- Possible to manipulate this measure

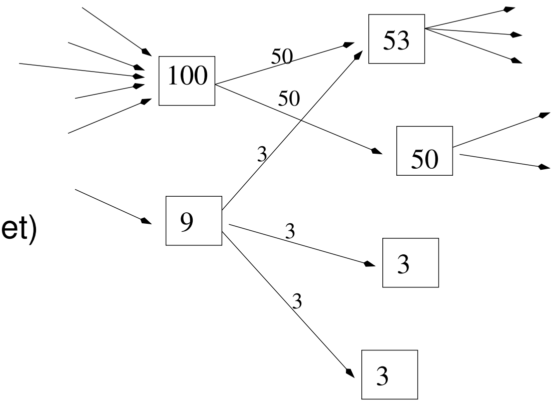
- Web links are not quite like scientific citations
  - Large variation in web pages: quality, purpose, number of links, length (whereas scientific articles are more homogeneous)
    - ∗ No quality check (cf. peer review in scientific articles)
    - ∗ No cost associated with links (cf. length restrictions in scientific articles)
    - ∗ No publishing/production costs associated with web sites
  - 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 $1 - q$
- The number of visits to each page is the PageRank of that page

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

Simplified PageRank (q=1.0):



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

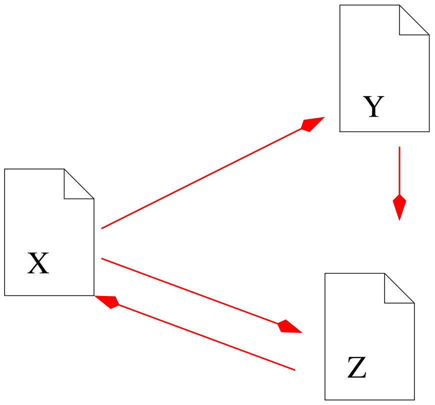
Matrix notation of PageRank

$$\vec{r} = c(A\vec{r} + \vec{e})$$

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

$\vec{r}$	PageRank vector (over all web pages), the desired result
A	normalised link matrix of the web:

$$A_{vu} = \begin{cases} \frac{1}{N_v} & if \exists u \rightarrow v \\ 0 & otherwise \end{cases}$$



From

$$\text{To}\quad \begin{array}{c|ccc|} & X & Y & Z \\ X & 0 & 0 & 1 \\ Y & .5 & 0 & 0 \\ Z & .5 & 1 & 0 \end{array} = A$$

Let’s calculate PageRank for this “mini-web”: eigenvector calculation!

Idealised PageRank computation

$A\vec{r} = \lambda\vec{r}$

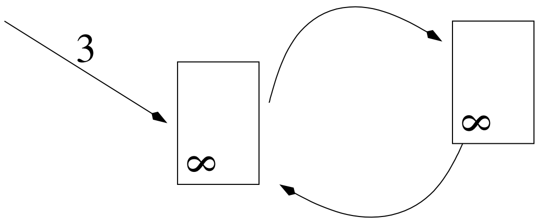
$$A = \begin{vmatrix} 0 & 0 & 1 \\ .5 & 0 & 0 \\ .5 & 1 & 0 \end{vmatrix}$$

$\vec{r}$ is the dominant eigenvector of A
λ is the eigenvalue (normalisation factor c)

$$\vec{r_0} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \vec{r_{1,n}} = \begin{bmatrix} .333 \\ .167 \\ .5 \end{bmatrix} \vec{r_{2,n}} = \begin{bmatrix} .5 \\ .167 \\ .333 \end{bmatrix} \vec{r_{3,n}} = \begin{bmatrix} .333 \\ .25 \\ .383 \end{bmatrix} \vec{r_{4,n}} = \begin{bmatrix} .399 \\ .200 \\ .399 \end{bmatrix} \quad \rightarrow \quad \vec{r} = \begin{bmatrix} .4 \\ .2 \\ .4 \end{bmatrix}$$

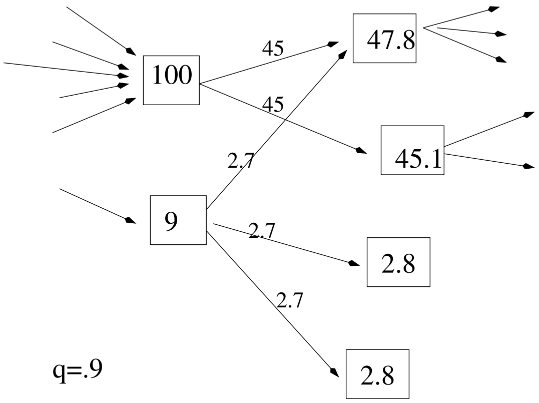
$$\vec{r_1} = \begin{bmatrix} 1 \\ .5 \\ 1.5 \end{bmatrix} \vec{r_2} = \begin{bmatrix} .5 \\ .167 \\ .333 \end{bmatrix} \vec{r_3} = \begin{bmatrix} .333 \\ .25 \\ .383 \end{bmatrix} \vec{r_4} = \begin{bmatrix} .383 \\ .192 \\ .383 \end{bmatrix}$$

λ_1=.333; λ_2=1
λ_3=1; λ_4=1.045



- 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 (therefore, $c < 1$)
- Solution: rank source $\vec{e}$ counteracts rank sinks
- $\vec{e}$ is the vector of the probability $1 - q$ for each page: the probability of random jumps of random surfer to a random page
- In practice: let $\vec{e}$ be a uniform vector, e.g with $1 - q$=.15

Actual PageRank calculation



```
PageRank computation:

   $\vec{r_0}$ := $S$
loop while $\delta > \epsilon$:
$\vec{r}_{i+1}$ := $A\vec{r_i}$
     $d$ := $||\vec{r_i}||_1 - ||\vec{r}_{i+1}||_1$
$\vec{r}_{i+1}$ := $\vec{r}_{i+1} + de$
     $\delta$ := $||\vec{r}_{i+1} - \vec{r_i}||_1$
```

- $\vec{r} = c(A + \vec{e} \times 1)\ \vec{r}$ (1 is the vector consisting of all ones)
- Then, $\vec{r}$ is an eigenvector of $(A + \vec{e} \times 1)$
- d is the normalisation factor

- Space
  - Example: 75 M unique links on 25 M pages
  - Then: memory for PageRank 300MB
- Time
  - Each iteration takes 6 minutes (for the 75 M links)
  - Whole process: 5 hours
  - Convergence after 52 iterations (322M links), 48 iterations (161M links)
  - Scaling factor linear in log n
- Pages without children removed during iteration
- Cost of computing PageRank is insignificant compared to the cost of building a full index
- PageRank is a good predictor of optimal crawling order

Why PageRank works

- Pages have different inherent importance
  - Yahoo’s home page is not the same as my home page
  - Better maintained, more useful, and its links are more important
  - Advertising on Yahoo is expensive
- Users want information from “trusted” sources
  - Collaborative trust
- Propagation simulates word-of-mouth effects in complex network (ahead of time)
  - Good pages often have only a few important backlinks (at first)
  - Those pages would not be found by simply back-link counting
- 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

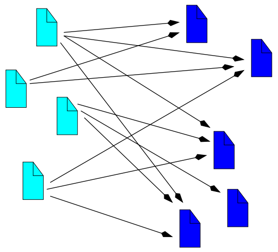
Download Netscape Software	11589.00
http://www.w3.org	10717.70
Welcome to Netscape	8673.51
Point: It's what you're searching for	7930.92
Web-Counter home page	7254.97
THe Blue Ribbon Campaign for Online Free Speech	7010.39
CERN Welcome	6562.49
Yahoo!	6561.80
Welcome to Netscape	6203.47
Wusage 4.1: A 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

PageRank versus usage data

- There is a difference between linking behaviour and actual usage data (web page access numbers from NLANR)
  - There are pages that people access a lot but don't want to point to in their web pages
  - PageRank has fewer privacy implications, as it uses only public information
- Link structures is compact (8B/link compressed) and raw data can be obtained during web crawl
- Finer resolution compared to small usage sample
- But not all web users create links
- PageRank can change fast (one link on Yahoo); Net traffic can change fast (one mention on the radio)

- 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

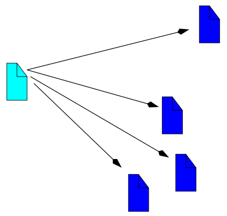


- Hubs: Recommendation pages with links to high-quality pages (authorities), e.g. compilations of favourite bookmarks, “useful links”
- Authorities: Pages that are recognised by others (particularly by hubs!) as experts on a certain topic
- Authorities are different from universally popular pages (high back-link count), which are not particular experts on that topic

HITS

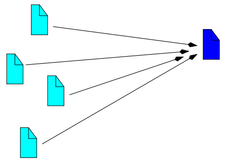
- Each page has two non-negative weights: an authority weight a and a hub weight h
- At each iteration, update the weights:
  - If a page points at many good authorities, it is probably a good hub:

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



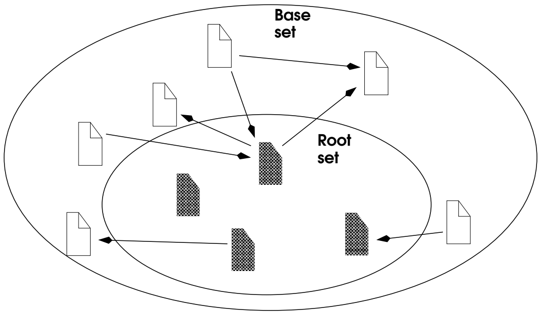
  - If a page is pointed to by many good hubs, it is probably a good authority:

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



- Normalise weights after each iteration

- 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 $D = \{D_1 \ldots D_n\}$ 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 $c = 1 \ldots k$

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

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

- Updates:

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

- After the first iteration:

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

- After the second iteration:

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

- Convergence to
  - $\vec{a} \leftarrow$ dominant eigenvector$(A^T A)$
  - $\vec{h} \leftarrow$ dominant eigenvector(AA^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
- Other differences between closed-world assumption (IR) and world wide web: data, indexing, query constructs, search heuristics