Web search basics

Content

- History
- Web Size
- Spam
- Link Analysis

History

Brief (non-technical) history

Hypertext

In the 1990's:

(1) Server communicates with the client via a protocol (http) that is lightweight and simple, asynchronous, simple markup language (HTML)

(2) The client (browser) ignores what it does not understand

Brief (non-technical) history

Making web content discoverable

(1) Full-text index (Altavista, Excite, Infoseek)

(2) Taxonomies populated in categories such as Yahoo!

  (1) Manual (difficult to scale)

  (2) Need s to now what sub-trees to seek

Brief (non-technical) history

First challenge: scale

Then: quality and relevance of query results

Brief (non-technical) history

- 1998+: Link-based ranking pioneered by Google
  - Blew away all early engines save Inktomi
  - Great user experience in search of a business model
  - Meanwhile Goto/Overture's annual revenues were nearing $1 billion

Brief (non-technical) history

- Early keyword-based engines
  - Altavista, Excite, Infoseek, Inktomi, ca. 1995-1997
- Sponsored search ranking: Goto.com (morphed into Overture.com → Yahoo!)
  - Your search ranking depended on how much you paid
  - Auction for keywords: ***casino*** was expensive!

Advertising

- Graphical banner advertisements on web pages at popular websites (news and entertainment sites, such as MSN, CNN, etc)
- Purpose: Branding
- Cost Per Mil (CPB) model: cost of having its banner advertisement displayed 1000 times (also called impression)
- Cost per Click (CPC) model
- Purpose: Make a purchase -> transaction oriented

Advertising

Goto (later Overture)

Not a search engine

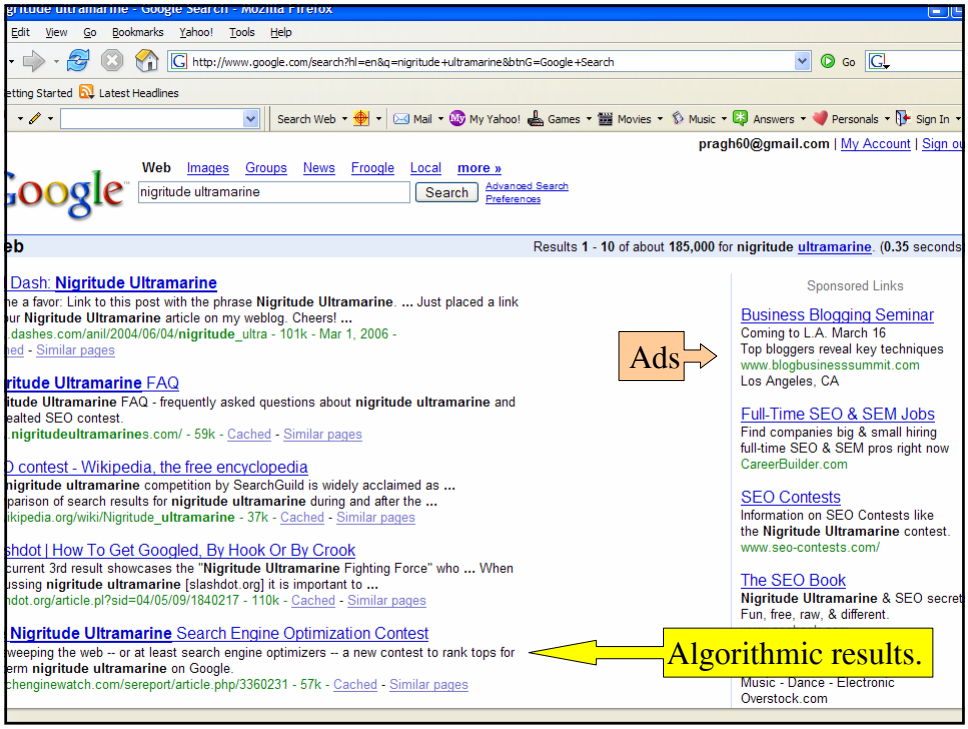
- For every query term *q*, it accepted bids for companies who wanted their web page shown on the query *q*
- As results, Goto returned the pages of all advertisers who bid for *q*
- When the user clicked, the advertiser would pay

Sponsored Search or Search advertising

Advertising

Combine:

- Pure search engines (aka *algorithmic search results*)
- Sponsored search engines (displayed separately and distinctively to the reight of the algorithmic results)



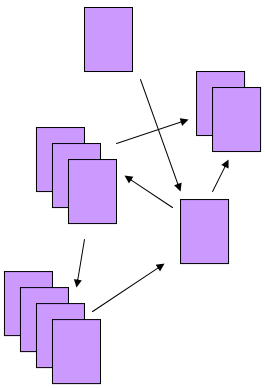
Advertising

Paid inclusion:

Pay to have one's web page included in the search engine's index

Effect on ranking or not

The Web document collection



The Web

- No design/co-ordination
- Distributed content creation, linking, democratization of publishing
- Content includes truth, lies, obsolete information, contradictions …
- Unstructured (text, html, …), semi-structured (XML, annotated photos), structured (Databases)…
- Scale much larger than previous text collections
- Growth – slowed down from initial "volume doubling every few months" but still expanding
- Content can be *dynamically generated*

- Static web page: its content does not vary from one request to that page to the next

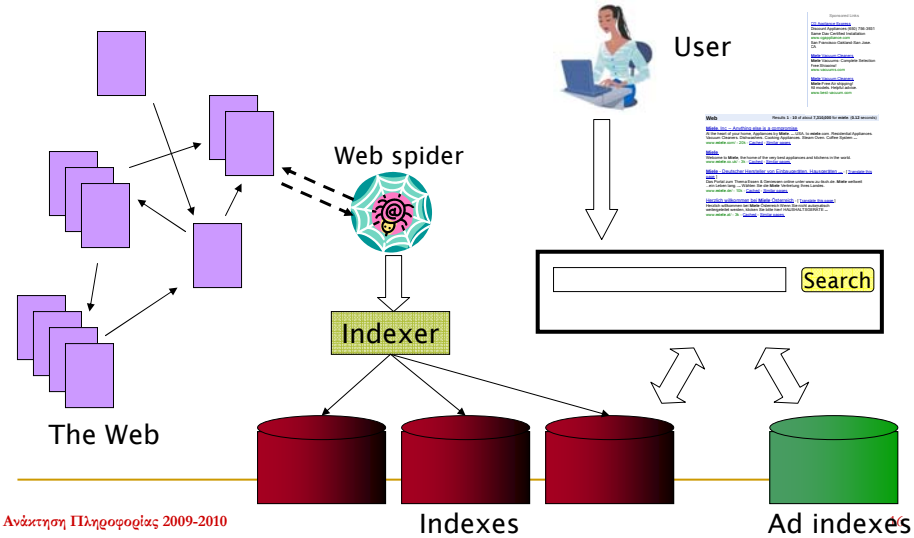
- Dynamic web page:

Typically generated by an application server in response to a query to a database

Character "?" in its URL

Example: airport's flight status, etc

Web search basics



User

User Needs

- Need [Brod02, RL04] `Low hemoglobin`
  1. **Informational** – want to learn about something (~40% / 65%)

     Not a single web page, assimilate information from many sites

     `United Airlines`
  2. **Navigational** – want to go to that page (~25% / 15%)

     Best, precision at 1

User Needs

3. Transactional – want to do something (web-mediated) (~35% / 20%)

- Access a service
- Downloads
- Shop

`Seattle weather`
`Mars surface images`
`Canon S410`

Listing sites with interfaces for such services

- **Gray areas**
  - Find a good hub
  - Exploratory search "see what's there"

`Car rental Brasil`

User Needs

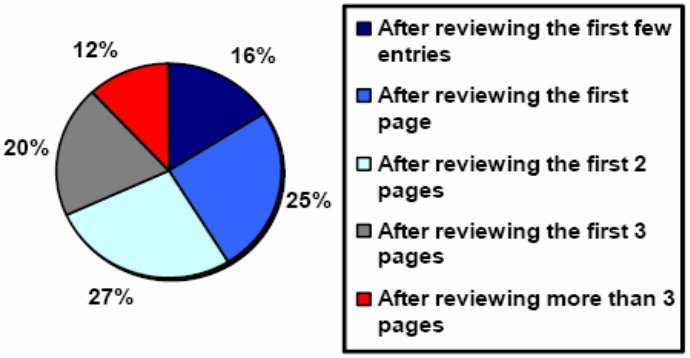
Type of query influences both the algorithmic search results and the query for sponsored search results

Other characteristics:

- Average number of keywords between 2 and 3
- Syntax operators are seldom used

How far do people look for results?

"When you perform a search on a search engine and don't find what you are looking for, at what point do you typically either revise your search, or move on to another search engine? (Select one)"



(Source: iprospect.com WhitePaper_2006_SearchEngineUserBehavior.pdf)

Users' empirical evaluation of results

- Quality of pages varies widely
  - Relevance is not enough
  - Other desirable qualities (non IR!!)
    - *Content*: Trustworthy, diverse, non-duplicated, well maintained
    - *Web readability*: display correctly & fast
    - *No annoyances*: pop-ups, etc
- Precision vs. recall
  - On the web, recall seldom matters
- What matters
  - Precision at 1? Precision above the fold?
  - Comprehensiveness – must be able to deal with obscure queries
    - Recall matters when the number of matches is very small
- User perceptions may be unscientific, but are significant over a large aggregate

Users' empirical evaluation of engines

- Relevance and validity of results
- UI – Simple, no clutter, error tolerant
- Trust – Results are objective
- Coverage of topics for polysemic queries
- Pre/Post process tools provided
  - Mitigate user errors (auto spell check, search assist,…)
  - Explicit: Search within results, more like this, refine ...
  - Anticipative: related searches
- Deal with idiosyncrasies
  - Web specific vocabulary
    - Impact on stemming, spell-check, etc
  - Web addresses typed in the search box
  - …

Spam

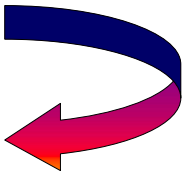
The trouble with sponsored search …

- It costs money. What's the alternative?
- *Search Engine Optimization:*
  - "Tuning" your web page to rank highly in the algorithmic search results for select keywords
  - Alternative to paying for placement, thus, intrinsically a marketing function
- Performed by companies, webmasters and consultants ("Search engine optimizers") for their clients
- Some perfectly legitimate, some very shady

Simplest forms

- First generation engines relied heavily on *tf/idf*
  - The top-ranked pages for the query **maui resort** were the ones containing the most **maui**'s and **resort**'s
- SEOs responded with dense repetitions of chosen terms
  - e.g., **maui resort maui resort maui resort**
  - Often, the repetitions would be in the same color as the background of the web page
    - Repeated terms got indexed by crawlers
    - But not visible to humans on browsers

Pure word density cannot be trusted as an IR signal



Variants of keyword stuffing

- Misleading meta-tags, excessive repetition
- Hidden text with colors, style sheet tricks, etc.

Meta-Tags =
"… London hotels, hotel, holiday inn, hilton, discount, booking, reservation, sex, mp3, britney spears, viagra, …"

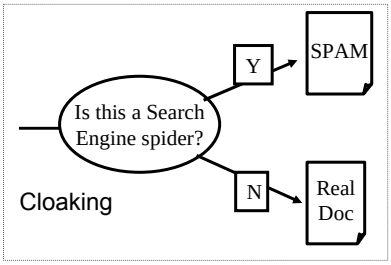
Search engine optimization (Spam)

- Motives
  - Commercial, political, religious, lobbies
  - Promotion funded by advertising budget
- Operators
  - Contractors (Search Engine Optimizers) for lobbies, companies
  - Web masters
  - Hosting services
- Forums
  - E.g., Web master world (www.webmasterworld.com)
    - Search engine specific tricks
    - Discussions about academic papers ☺

Cloaking

- Serve fake content to search engine spider
- DNS cloaking: Switch IP address. Impersonate

Get indexed under misleading keywords



More spam techniques

- Doorway pages
  - Pages optimized for a single keyword that re-direct to the real target page
- Link spamming
  - Mutual admiration societies, hidden links, awards – more on these later
  - *Domain flooding:* numerous domains that point or re-direct to a target page
- Robots
  - Fake query stream – rank checking programs
    - "Curve-fit" ranking programs of search engines
  - Millions of submissions via Add-Url

Click spam

The war against spam

- Quality signals - Prefer authoritative pages based on:
  - Votes from authors (linkage signals)
  - Votes from users (usage signals)
- Policing of URL submissions
  - Anti robot test
- Limits on meta-keywords
- Robust link analysis
  - Ignore statistically implausible linkage (or text)
  - Use link analysis to detect spammers (guilt by association)
- Spam recognition by machine learning
  - Training set based on known spam
- Family friendly filters
  - Linguistic analysis, general classification techniques, etc.
  - For images: flesh tone detectors, source text analysis, etc.
- Editorial intervention
  - Blacklists
  - Top queries audited
  - Complaints addressed
  - Suspect pattern detection

More on spam

- Web search engines have policies on SEO practices they tolerate/block
  - http://help.yahoo.com/help/us/ysearch/index.html
  - http://www.google.com/intl/en/webmasters/
- Adversarial IR: the unending (technical) battle between SEO's and web search engines
- Research http://airweb.cse.lehigh.edu/

Web as a graph

Web Graph

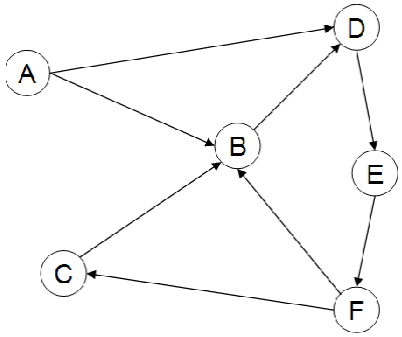
As a directed graph

Nodes: static HTL pages

Edge: hyperlinks between pages

Anchor text (href attribute of <a>)

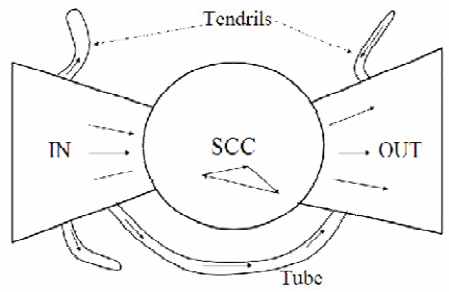
Web Graph



- Not strongly connected
- In-links, in-degree: 8 – 15
- Out-links, out-degree
- Power law distribution of in-degree: web pages with in-degree i is proportional to $1/i^{\alpha}$, $\alpha = 2.1$

Web Graph

Bowtie structure



SCC: Strongly connected component

IN, **OUT** roughly equal in size, SCC somewhat larger

Most web pages in one of the three sets

Tubes (small sets outside SCC that lead directly form IN to OUT)

Tendrils (lead nowhere from IN, or from nowhere to OUT)

Size of the web

19.5

What is the size of the web ?

- Issues
  - The web is really infinite
    - Dynamic content, e.g., calendar
    - Soft 404: www.yahoo.com/<anything> is a valid page
  - Static web contains syntactic duplication, mostly due to mirroring (~30%)
  - Some servers are seldom connected
- Who cares?
  - Media, and consequently the user
  - Engine design
  - Engine crawl policy. Impact on recall.

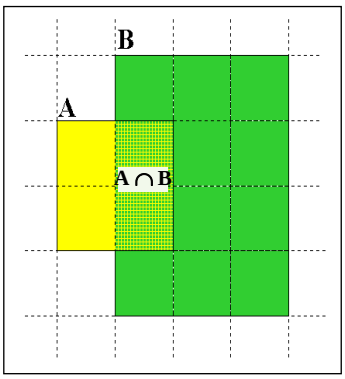
What can we attempt to measure?

- The relative sizes of search engines
  - The notion of a page being indexed is still *reasonably* well defined.
  - Already there are problems
    - Document extension: e.g. engines index pages not yet crawled, by indexing anchortext.
    - Document restriction: All engines restrict what is indexed (first *n* words, only relevant words, etc.)
- The coverage of a search engine relative to another particular crawling process.

New definition?

- (IQ is whatever the IQ tests measure.)
  - The statically indexable web is whatever search engines index.
- Different engines have different preferences
  - max url depth, max count/host, anti-spam rules, priority rules, etc.
- Different engines index different things under the same URL:
  - frames, meta-keywords, document restrictions, document extensions, ...

Relative Size from Overlap given two engines A and B



Sample URLs randomly from A

Check if contained in B and vice versa

$A \cap B = (1/2) * \text{Size } A$

$A \cap B = (1/6) * \text{Size } B$

$(1/2)*\text{Size } A = (1/6)*\text{Size } B$

$\therefore \text{ Size } A / \text{ Size } B = (1/6)/(1/2) = 1/3$

Each test involves: (i) Sampling (ii) Checking

Sampling URLs

Assumption: A and B independent and uniform random subsets of the Web

Have or have not access to the search engine

How to achieve a sample …

Sampling URLs

- Ideal strategy: Generate a random URL and check for containment in each index.
- Problem: Random URLs are hard to find! Enough to generate a random URL contained in a given Engine.
- Approach 1: Generate a random URL contained in a given engine
  - Suffices for the estimation of relative size
- Approach 2: Random walks / IP addresses
  - In theory: might give us a true estimate of the size of the web (as opposed to just relative sizes of indexes)

Statistical methods

- Approach 1
  - Random queries
  - Random searches
- Approach 2
  - Random IP addresses
  - Random walks

Random URLs from random queries

- Generate random query: how?
  - picking random terms from say Webster's dictionary
    - Not all terms occur equally often (not the same as chosen documents uniformly at random from a search engine)
    - Many terms not in the dictionary
    - Thus, a sample web dictionary: **Lexicon:** 400,000+ words from a web crawl
  - **Conjunctive Queries:** w_1 and w_2

    e.g., vocalists AND rsi

Random URLs from random queries

- Get 100 result URLs from engine A
- Choose a random URL *p* as the candidate to check for presence in engine B
- This distribution induces a probability weight W(*p*) for each page.
- Conjecture: $W(SE_A) / W(SE_B) \sim |SE_A| / |SE_B|$

How to test for the presence of p (document D) in B?

Query Based Checking

- *Strong Query* to check whether an engine *B* has a document *D*:
  - Download *D*. Get list of words.
  - Use 6-8 low frequency words as AND query to *B*
  - Check if *D* is present in result set.
- Problems:
  - Near duplicates
  - Frames
  - Redirects
  - Engine time-outs
  - Is 8-word query good enough?

Advantages & disadvantages

- Statistically sound under the induced weight.
- Biases induced by random query
  - Query Bias: Favors content-rich pages in the language(s) of the lexicon
  - Ranking Bias: *Solution:* Use conjunctive queries & fetch all
  - Checking Bias: Duplicates, impoverished pages omitted
  - Document or query restriction bias: engine might not deal properly with 8 words conjunctive query
  - Malicious Bias: Sabotage by engine
  - Operational Problems: Time-outs, failures, engine inconsistencies, index modification.

Random searches

- Choose random searches extracted from a local log [Lawrence & Giles 97] or build "random searches" [Notess]
  - Use only queries with small results sets.
  - Count normalized URLs in result sets.
  - Use ratio statistics

Advantages & disadvantages

- Advantage
  - Might be a better reflection of the human perception of coverage
- Issues
  - Samples are correlated with source of log
  - Duplicates
  - Technical statistical problems (must have non-zero results, ratio average not statistically sound)

Random searches

- 575 & 1050 queries from the NEC RI employee logs
- 6 Engines in 1998, 11 in 1999
- Implementation:
  - Restricted to queries with < 600 results in total
  - Counted URLs from each engine after verifying query match
  - Computed size ratio & overlap for individual queries
  - Estimated index size ratio & overlap by averaging over all queries

Queries from Lawrence and Giles study

- *adaptive access control*
- *neighborhood preservation topographic*
- *hamiltonian structures*
- *right linear grammar*
- *pulse width modulation neural*
- *unbalanced prior probabilities*
- *ranked assignment method*
- *internet explorer favourites importing*
- *karvel thornber*
- *zili liu*
- *softmax activation function*
- *bose multidimensional system theory*
- *gamma mlp*
- *dvi2pdf*
- *john oliensis*
- *rieke spikes exploring neural*
- *video watermarking*
- *counterpropagation network*
- *fat shattering dimension*
- *abelson amorphous computing*

Random IP addresses

- Generate random IP addresses
- Find a web server at the given address
  - If there's one
- Collect all pages from server
  - From this, choose a page at random

Random IP addresses

- HTTP requests to random IP addresses
  - Ignored: empty or authorization required or excluded
  - [Lawr99] Estimated 2.8 million IP addresses running crawlable web servers (16 million total) from observing 2500 servers.
  - OCLC using IP sampling found 8.7 M hosts in 2001
    - Netcraft [Netc02] accessed 37.2 million hosts in July 2002
- [Lawr99] exhaustively crawled 2500 servers and extrapolated
  - Estimated size of the web to be 800 million
  - Estimated use of metadata descriptors:
    - Meta tags (keywords, description) in 34% of home pages, Dublin core metadata in 0.3%

Advantages & disadvantages

- Advantages
  - Clean statistics
  - Independent of crawling strategies
- Disadvantages
  - Doesn't deal with duplication
  - Many hosts might share one IP, or not accept requests
  - No guarantee all pages are linked to root page.
    - Eg: employee pages
  - Power law for # pages/hosts generates bias towards sites with few pages.
    - But bias can be accurately quantified IF underlying distribution understood
  - Potentially influenced by spamming (multiple IP's for same server to avoid IP block)

Random walks

- View the Web as a directed graph
- Build a random walk on this graph
  - Includes various "jump" rules back to visited sites
    - Does not get stuck in spider traps!
    - Can follow all links!
  - Converges to a stationary distribution
    - Must assume graph is finite and independent of the walk.
    - Conditions are not satisfied (cookie crumbs, flooding)
    - Time to convergence not really known
  - Sample from stationary distribution of walk
  - Use the "strong query" method to check coverage by SE

Advantages & disadvantages

- Advantages
  - "Statistically clean" method at least in theory!
  - Could work even for infinite web (assuming convergence) under certain metrics.
- Disadvantages
  - List of seeds is a problem.
  - Practical approximation might not be valid.
  - Non-uniform distribution
    - Subject to link spamming

Conclusions

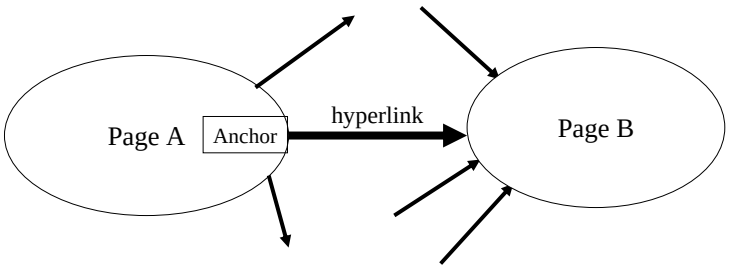
- No sampling solution is perfect.
- Lots of new ideas ...
- ....but the problem is getting harder
- Quantitative studies are fascinating and a good research problem

Link Analysis

Content

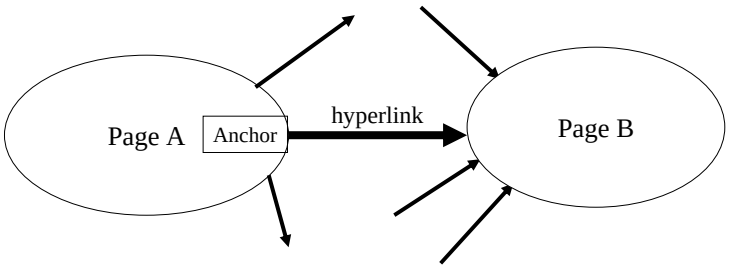
- Introduction
- PageRank
- HITS

The Web as a Directed Graph



Assumption 1: The anchor of the hyperlink describes the target page (textual context)

The Web as a Directed Graph

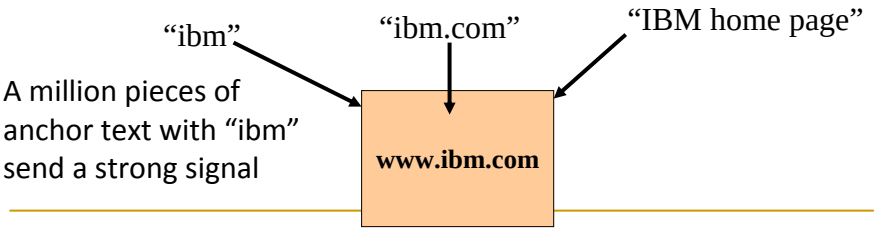


Assumption 2: A hyperlink between pages denotes author perceived relevance (quality signal)
An endorsement of page B by the creator of A

Anchor Text

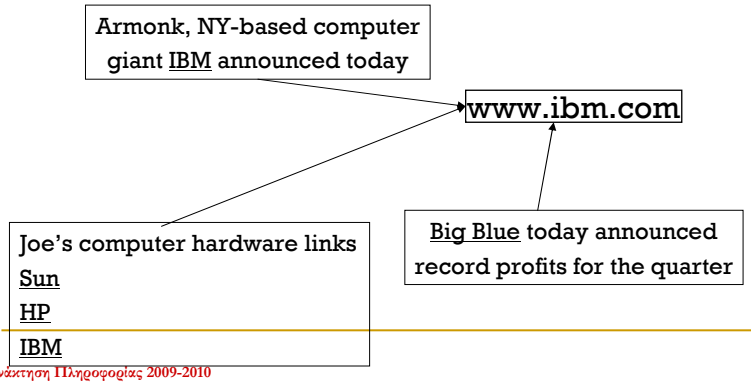
WWW Worm - McBryan [Mcbr94]

- For ***ibm*** how to distinguish between:
  - IBM's home page (mostly graphical)
  - IBM's copyright page (high term freq. for 'ibm')
  - Rival's spam page (arbitrarily high term freq.)



Indexing anchor text

- When indexing a document *D*, include anchor text from links pointing to *D*.



Indexing anchor text

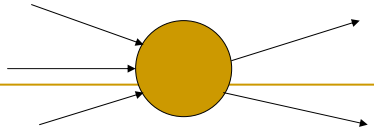
- Can sometimes have unexpected side effects - *e.g.*, ***evil empire***.
- Can score anchor text with weight depending on the authority of the anchor page's website
  - E.g., if we were to assume that content from cnn.com or yahoo.com is authoritative, then trust the anchor text from them

Anchor Text

- Gap between the terms in a web page ad how users would describe this web page
- Index also the window surrounding anchor text
- Anchor text terms weighted based on frequency (similar to idf – “Click” “Here”)

Query-independent ordering

- First generation: using link counts as simple measures of popularity.
- Two basic suggestions:
  - Undirected popularity:
    - Each page gets a score = the number of in-links plus the number of out-links (3+2=5).
  - Directed popularity:
    - Score of a page = number of its in-links (3).



Query processing

- First retrieve all pages meeting the text query (say ***venture capital***).
- Order these by their link popularity (either variant on the previous page).

Spamming simple popularity

- How do you spam each of the following heuristics so your page gets a high score?
- Each page gets a static score = the number of in-links plus the number of out-links.
- Static score of a page = number of its in-links.

Citation Analysis

- Citation frequency
- Co-citation coupling frequency
  - Cocitations with a given author measures "impact"
  - Cocitation analysis
- Bibliographic coupling frequency
  - Articles that co-cite the same articles are related
- Citation indexing
  - Who is author cited by? (Garfield 1972)
- Pagerank preview: Pinsker and Narin '60s

PageRank

PageRank scoring

Not all links to a page are equal

Links from "important" pages (i.e. pages with many links) count more

Assign to every node (page) in the web graph a numerical score between 0 and 1-> PageRank

Given a query: compute a composite score for each web page that combines relevance with PageRank

Ορισμός PageRank

Παράδειγμα

Έστω ότι υπάρχει μια γενική ποσότητα PR που μοιράζεται στις σελίδες του συστήματος.
Έστω 4 σελίδες: A, B, C και D.
Αρχική προσεγγιστική τιμή για καθεμία: PR = 0.25

- Έστω B, C, και D έχουν link μόνο στο A,
τότε όλα το PageRank PR( ) τους θα μαζευόταν στο A

$$PR(A) = PR(B) + PR(C) + PR(D).$$

- Έστω τώρα ότι η B έχει link στη C, και η D έχει links και στο B και στο C
Η τιμή του PR μιας σελίδας μοιράζεται ανάμεσα στις εξωτερικές ακμές της
Άρα η ψήφος της B έχει αξία για την A 0.125 και 0.125 για την C.
Αντίστοιχα, μόνο το 1/3 του PageRank του D μετρά για PageRank του A (περίπου 0.083).

$$PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1} + \frac{PR(D)}{3}.$$

Ορισμός PageRank

Γενικός ορισμός του PageRank για μια σελίδα Α:

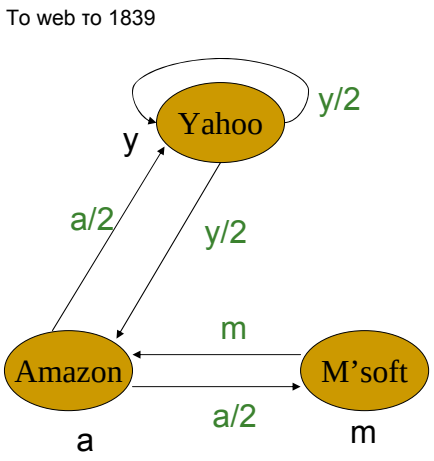
Έστω ότι η Α έχει τις σελίδες Τ1, ...,Τn που δείχνουν σε αυτήν (δηλαδή, αναφορές)

Έστω C(T) ο αριθμός των εξωτερικών ακμών μιας σελίδας Τ

$$PR(A) = PR(T1)/C(T1) + ... + PR(Tn)/C(Tn)$$

Υπολογισμός PageRank

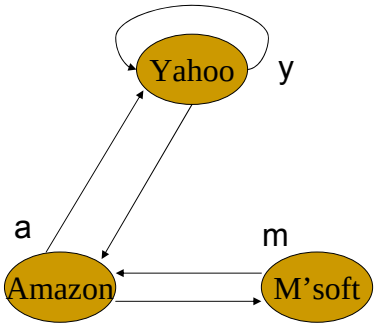
Απλό μοντέλο «ροής» -"flow" model



PageRank: a, y, m

$y = y/2 + a/2$
$a = y/2 + m$
$m = a/2$

Διατύπωση με την μορφή πίνακα



$y = y/2 + a/2$

$a = y/2 + m$

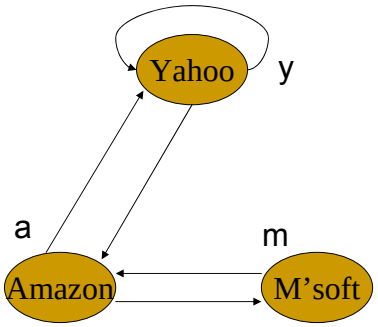
$m = a/2$

Adjacency Matrix

	y	a	m
y	1/2	1/2	0
a	1/2	0	1
m	0	1/2	0

Άθροισμα 1 (οι ψήφοι στο y)

Διατύπωση με την μορφή πίνακα (παράδειγμα)



$y = y/2 + a/2$

$a = y/2 + m$

$m = a/2$

r (rank vector)

r [y, a, m]

A = Adjacency Matrix

r = A r

$$\begin{bmatrix} y \\ a \\ m \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} y \\ a \\ m \end{bmatrix}$$

Ιδιοδιανύσματα (eigenvectors)

- Οι εξισώσεις ροής μπορούν να γραφούν

$$\mathbf{r} = \mathbf{M}\,\mathbf{r}$$

- Δηλαδή, ο rank vector είναι ένα ιδιοδιάνυσμα (eigenvector) του στοχαστικού πίνακα γειτνίασης του web
  - Συγκεκριμένα είναι το βασικό ιδιοδιάνυσμα (αυτό που αντιστοιχεί στην ιδιοτιμή λ = 1)

Power Iteration method – Επαναληπτική Μέθοδο

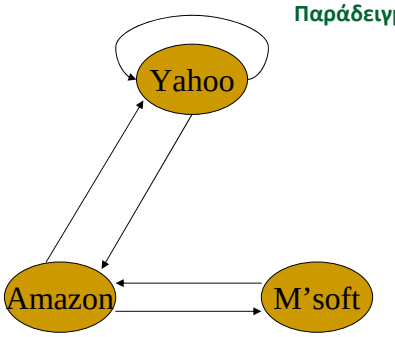
Ένα απλό επαναληπτικό σχήμα (relaxation)
Έστω N web σελίδες

Αρχικοποίηση: $\mathbf{r}^0 = [1/N,\ldots,1/N]^T$
Επανάληψη: $\mathbf{r}^{k+1} = \mathbf{M}\mathbf{r}^k$
Τερματισμός όταν $|\mathbf{r}^{k+1} - \mathbf{r}^k|_1 < \varepsilon$

$|\mathbf{x}|_1 = \sum_{1 \le i \le N} |x_i|$ είναι L_1 norm

Μπορεί να χρησιμοποιηθούν και άλλες μετρικές, πχ Ευκλείδεια

Παράδειγμα



	y	a	m
y	1/2	1/2	0
a	1/2	0	1
m	0	1/2	0

y		1/3	1/3	5/12	3/8		2/5
a	=	1/3	1/2	1/3	11/24	. . .	2/5
m		1/3	1/6	1/4	1/6		1/5

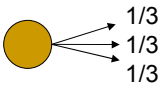
Συγκλίνει; Μοναδική Λύση;

PageRank scoring

Intuitively, the probability a random surfer would visit the node

PageRank scoring

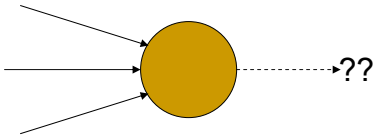
- Imagine a browser doing a random walk on web pages:
  - Start at a random page
  - At each step, go out of the current page along one of the links on that page, equiprobably



- "In the steady state" each page has a long-term visit rate - use this as the page's score.

Not quite enough

- The web is full of dead-ends (no out-links).
  - Random walk can get stuck in dead-ends.
  - Makes no sense to talk about long-term visit rates.



Teleporting

- At a dead end, jump to a *random web page*.
- At any non-dead end, with probability α, say α = 10%, jump to a random web page.
  - With remaining probability (90%), go out on a random link.
  - 10% (α) - a parameter.

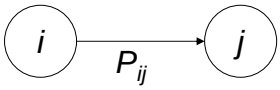
If N total number of web pages: teleport with 1/N

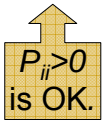
Result of teleporting

- Now cannot get stuck locally.
- There is a long-term rate at which any page is visited
- How do we compute this visit rate?

Markov chains

- A Markov chain consists of *n* states, plus an *n*×*n* transition probability matrix **P**.
- At each step, we are in exactly one of the states.
- For $1 \le i,j \le n$, the matrix entry P_{ij} tells us the probability of *j* being the next state, given we are currently in state *i* (transition probability, Markov property, depends only on i)

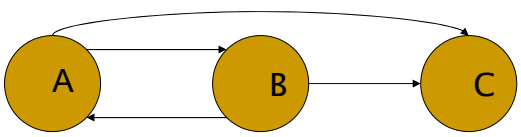




Markov chains

- Clearly, for all i, $\sum_{j=1}^{n} P_{ij} = 1.$
- Markov chains are abstractions of random walks.

example



Random Surfer and Markov chains

State -> web page

Transition probability -> probability moving from one page to another

Adjacency matrix A of the web:

A_{ij} = 1 if ∃ link from i to j, 0 otherwise

Random Surfer and Markov chains

Adjacency matrix A of the web -> Probability matrix P

- Divide each 1 in A by the number of 1's in its row
- Multiple the resulting matrix by (1 – α)

Add α/N to every entry of the resulting matrix

Random Surfer and Markov chains

Example

Three nodes, 1, 2 and 3

1-> 2, 3->2, 2->3 and α = 0.5

Random Surfer and Markov chains

The probability of a surfer's position at any time by a vector x

At t = 0, if at state t, (1 at the corresponding state, all others 0)

At t = 1, x P

At t = 2 (xP) P and so on

Does it converges?

PageRank of each node u = steady-state visit frequency

Probability vectors

- 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.
  1 i n

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

$$\sum_{i=1}^{n} x_i = 1.$$

Ergodic Markov chains

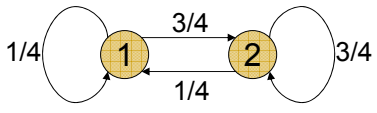
- A Markov chain is ergodic if
  - you have a path from any state to any other
  - For any start state, after a finite transient time T_0, the probability of being in any state at a fixed time $T>T_0$ is nonzero.

Ergodic Markov chains

- 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.

Steady state example

- The steady state looks like a vector of probabilities **a** = $(a_1, \ldots a_n)$:
  - a_i is the probability that we are in state *i*.



For this example, $a_1=1/4$ and $a_2=3/4$.

How do we compute this vector?

- Let **a** = $(a_1, \ldots a_n)$ denote the row vector of steady-state probabilities.
- If we 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 larges eigenvalue 1.

One way of computing **a**

- Recall, regardless of where we start, we eventually reach the steady state **a**.
- Start with any distribution (say **x**=(*10…0*)).
- After one step, we're at **xP**;
- after two steps at $\mathbf{xP}^2$, then $\mathbf{xP}^3$ and so on.
- "Eventually" means for "large" *k*, $\mathbf{xP}^k = \mathbf{a}$.
- Algorithm: multiply **x** by increasing powers of **P** until the product looks stable.

Pagerank summary

- Preprocessing:
  - Given graph of links, build 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-*independent*.

The reality

- Pagerank is used in google, but so are many other clever heuristics.

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)

PageRank

Topic-Specific PageRank

Topic Specific Pagerank

Idea:

Teleport to a random page non-uniformly

How?

Topic Specific 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
- Sounds hard to implement: can't compute PageRank at query time!

Topic Specific Pagerank

- Offline:Compute pagerank for *individual* categories
  - Query independent as before
  - Each page has multiple pagerank scores – one for each ODP category, with teleportation only to that category
- Online: Distribution of weights over categories computed by query context classification
  - Generate a dynamic pagerank score for each page - weighted sum of category-specific pageranks

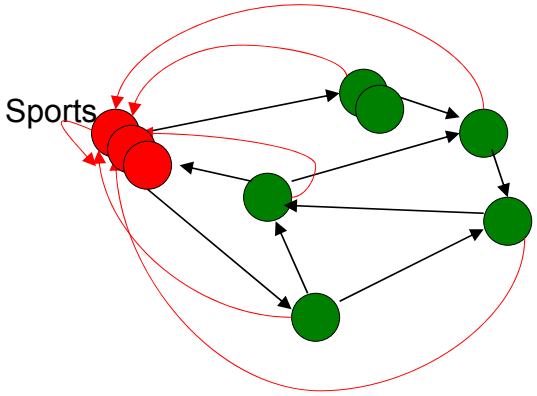
Influencing PageRank ("Personalization")

- Input:
  - Web graph *W*
  - influence vector **v**

    v : (page → degree of influence)
- Output:
  - Rank vector **r**: (page → page importance wrt **v**)
- **r** = PR(*W* , **v**)

Non-uniform Teleportation

Teleport with 10% probability to a Sports page



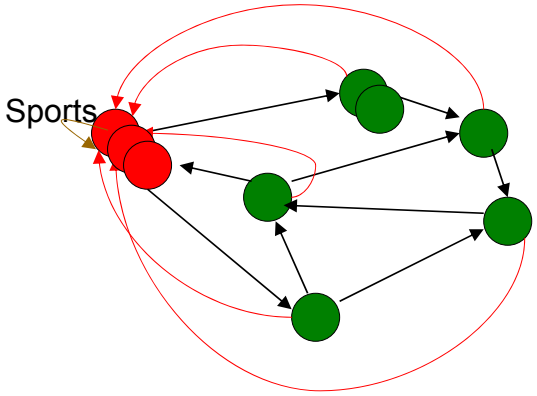
Interpretation of Composite Score

- For a set of personalization vectors $\{\mathbf{v}_j\}$

$$\sum_j [w_j \cdot PR(W, \mathbf{v}_j)] = PR(W, \sum_j [w_j \cdot \mathbf{v}_j])$$

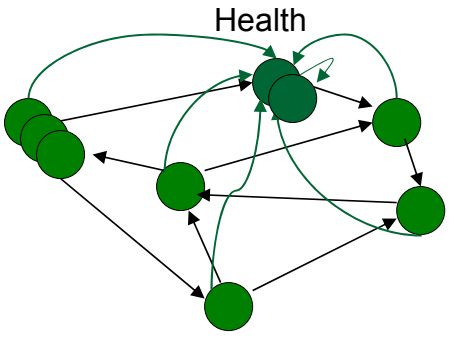
- Weighted sum of rank vectors itself forms a valid rank vector, because PR() is linear wrt $\mathbf{v}_j$

Interpretation



10% Sports teleportation

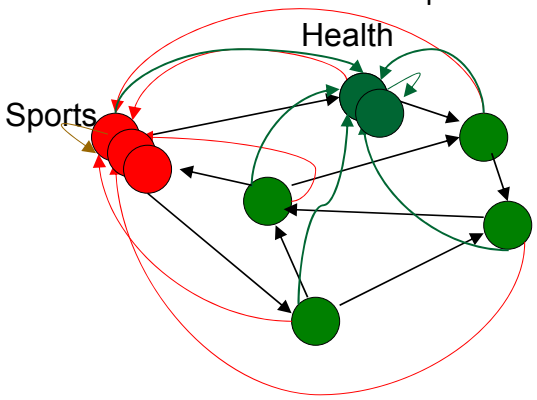
Interpretation



10% Health teleportation

Interpretation

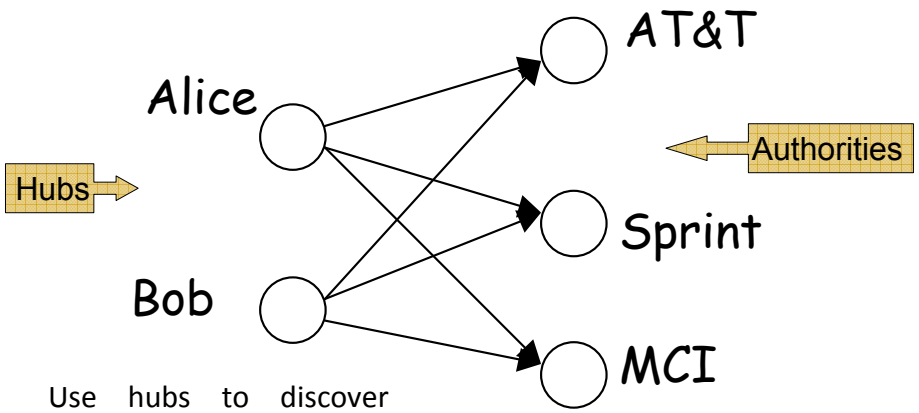
pr = (0.9 PR_{sports} + 0.1 PR_{health}) gives you:
9% sports teleportation, 1% health teleportation



HITS

The hope

Query: Long distance telephone companies



Use hubs to discover authorities

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:

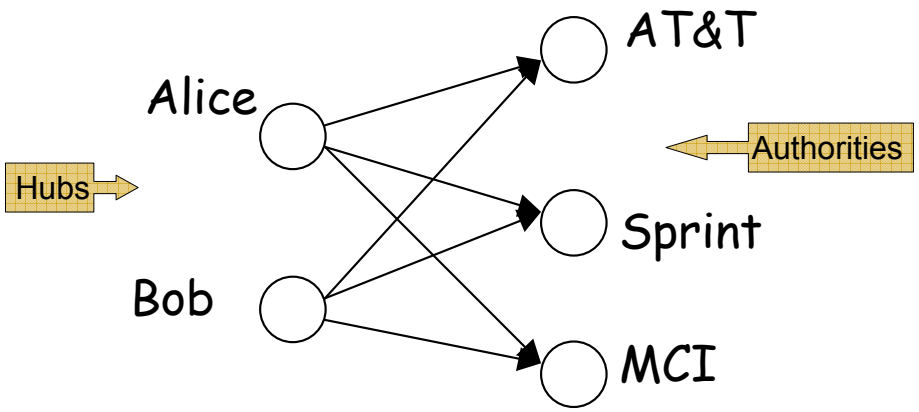
- *Hub pages* are good lists of links on a subject.
  - e.g., "Bob's list of cancer-related links."
- *Authority pages* occur recurrently on good hubs for the subject.

Each page has two scores for each query: a hub score and an authority score

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.

The hope



Query: Long distance telephone companies

Hyperlink-Induced Topic Search (HITS)

- Best suited for “broad topic” queries rather than for page-finding queries.
- Gets at a broader slice of common *opinion*.

High-level scheme

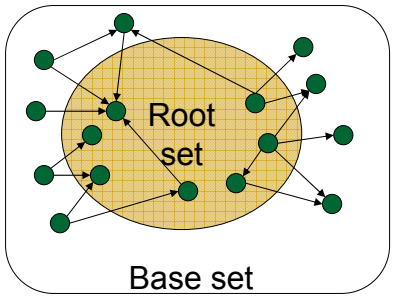
- Extract from the web a base set of pages that *could* be good hubs or authorities.
- From these, identify a small set of top hub and authority pages;
  - → iterative algorithm.

Base set

Query specific

- Given text query (say ***browser***), use a text index to get all pages containing ***browser.***
  - Call this the root set of pages.
- 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.

Visualization



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 (and out-links) from a *connectivity server.*
  - (Actually, suffices to text-index strings of the form ***href="URL"*** to get in-links to *URL*.)

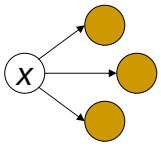
Distilling hubs and authorities

- 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)←1; a(x) ←1;*
- Iteratively update all *h(x), a(x)*;
- After iterations
  - output pages with highest *h()* scores as top hubs
  - highest *a()* scores as top authorities.

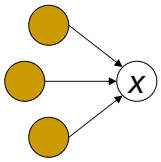
Iterative update

- Repeat the following updates, for all *x*:

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



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



Αναπαράσταση με πίνακες

Έστω το βασικό σύνολο σελίδων {1, 2, ..., n}

Πίνακας Γειτνίασης (adjacency matrix) B: n x n

B[i, j] = 1 αν η σελίδα i περιέχει σύνδεσμο που δείχνει στη σελίδα j

Έστω h = <h_1, h_2, ..., h_n> το διάνυσμα συντελεστών κομβικών ρόλων

και α = <α_1, α_2, ..., α_n> το διάνυσμα συντελεστών αυθεντικότητας

(αντίστοιχο του r vector)

Αναπαράσταση με πίνακες

Οι κανόνες ενημέρωσης

Αρχικά

$h = B\ a$ $\qquad a = B^T h$

1η επανάληψη

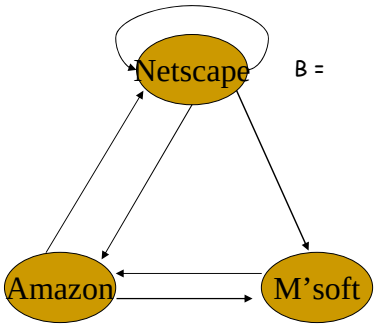
$h = B\ B^T h = (B\ B^T)\ h$ $\qquad a = B^T B\ a = (B^T B)\ a$

2η επανάληψη

$h = (B\ B^T)^2\ h$ $\qquad a = (B^T B)^2\ a$

Σύγκλιση στα ιδιοδιανύσματα του BB^T και B^TB αν κανονικοποιηθούν αρχικά οι συντελεστές

Αναπαράσταση με πίνακες



B =

n	m	a
1	1	1
0	0	1
1	1	0

B^T =

n	m	a
1	0	1
1	0	1
1	1	0

$B\ B^T$ =

n	m	a
3	1	2
1	1	0
2	0	2

$$\begin{pmatrix} 3 & 1 & 2 \\ 1 & 1 & 0 \\ 2 & 0 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 6 \\ 2 \\ 4 \end{pmatrix} \ldots$$

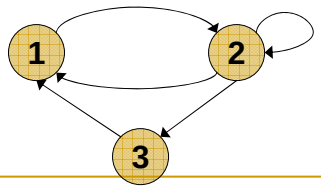
$h = BB^T h$

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.

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.



	1	2	3
1	0	1	0
2	1	1	1
3	1	0	0

Hub/authority vectors

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

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

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

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}Aa**.

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.

How many iterations?

- Claim: relative values of scores will converge after a few iterations:
  - in fact, suitably scaled, *h()* and *a()* scores settle into a steady state!
- We only require the relative orders of the *h()* and *a()* scores - not their absolute values.
- In practice, ~5 iterations get you close to stability.

Japan Elementary Schools

Hubs

- schools
- LINK Page-13
- “ú–{,ÌŠw□Z
- □a‰„□¬Šw□Zƒz□[ƒ□ƒy□[ƒW
- 100 Schools Home Pages (English)
- K-12 from Japan 10/...rnet and Education)
- http://www...iglobe.ne.jp/~IKESAN
- ,I,f,j□¬Šw□Z,U”N,P‘g•¨Œê
- □ÒŠ—’¬—§□ÒŠ—“Œ□¬Šw□Z
- Koulutus ja oppilaitokset
- TOYODA HOMEPAGE
- Education
- Cay's Homepage(Japanese)
- –y“ì□¬Šw□Z,Ìƒz□[ƒ□ƒy□[ƒW
- UNIVERSITY
- ‰J—³□¬Šw□Z DRAGON97-TOP
- □Â‰ª□¬Šw□Z,T”N,P‘gƒz□[ƒ□ƒy□[ƒW
- ¶µ°é¼ÂÁ© ¥á¥Ë¥â¡¼ ¥á¥Ë¥â¡¼

Authorities

- The American School in Japan
- The Link Page
- ‰ª□è□s—§ˆä“c□¬Šw□Zƒz□[ƒ□ƒy□[ƒW
- Kids' Space
- ˆÀ□é□s—§ˆÀ□é□¼•”□¬Šw□Z
- ‹{□é‹³ˆç‘âŠw•□‘®□¬Šw□Z
- KEIMEI GAKUEN Home Page (Japanese)
- Shiranuma Home Page
- fuzoku-es.fukui-u.ac.jp
- welcome to Miasa E&J school
- □_“Þ□ìŒ§□E‰¡•I□s—§‘†□ì□¼□¬Šw□Z,Ìƒy
- 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.

Issues

- Topic Drift
  - Off-topic pages can cause off-topic “authorities” to be returned
    - E.g., the neighborhood graph can be about a “super topic”
- Mutually Reinforcing Affiliates
  - Affiliated pages/sites can boost each others’ scores
    - Linkage between affiliated pages is not a useful signal

Resources

- IIR Chap 21
- http://www2004.org/proceedings/docs/1p309.pdf
- http://www2004.org/proceedings/docs/1p595.pdf
- http://www2003.org/cdrom/papers/refereed/p270/kamvar-270-xhtml/index.html
- http://www2003.org/cdrom/papers/refereed/p641/xhtml/p641-mccurley.html