Google Maths

Philip Knight

March 13, 2009

Outline

- Brief history of search engines
- Google and Googleplex
- Web as a graph
- Link Analysis
- Search 3.0, 4.0, 5.0, ...

History

- Archie (1990), Veronica and Jughead (1993)
- WWW Wanderer, ALIWEB(1993)
- WWW Worm, JumpStation, RBSE (1993)
- WebCrawler (1994)
- Yahoo, Lycos (1994), AltaVista (1995)
- Google (1998)

Google and Googleplex

- Founded by Larry Page and Sergei Brin.
- "Scraped together" **$1000000** in 1998 to start up.

Google and Googleplex

- Founded by Larry Page and Sergei Brin.
- "Scraped together" **$1000000** in 1998 to start up.
- First office a garage (9/98).
- By 6/99 had **$25000000** in funding.

Google and Googleplex

- Founded by Larry Page and Sergei Brin.
- "Scraped together" **$1000000** in 1998 to start up.
- First office a garage (9/98).
- By 6/99 had **$25000000** in funding.
- Launched search engine 9/99, "world's biggest" 9 months later.

Google and Googleplex

- Founded by Larry Page and Sergei Brin.
- "Scraped together" $1000000 in 1998 to start up.
- First office a garage (9/98).
- By 6/99 had $25000000 in funding.
- Launched search engine 9/99, "world's biggest" 9 months later.
- 14142135 shares floated in 2004 for $2718281828.
- Current market value is $100,000,000,000.

Fundamental Problems

A successful search engine requires

- A well-maintained index.
- Effective presentation of results.
- A good business model.
- Properly ordered results.

Web As A Graph

- Think of web as a collection of vertices and edges.
- Vertices (V) are web pages.
- Edges (E) are links.
- Web is a directed graph $G(V, E)$: edge goes from v_i to v_j if page i links to page j.
- $|V| \approx 20,000,000,000$, $|E| \approx 200,000,000,000$.

Is there any structure?

- Define in-degree, $I(v)$, as number of vertices connecting to vertex v.
- $Pr(I(v) = i) \approx C/i^x$ (some $x > 1$).
- Web appears to have properties of a small world network
- Most pages separated by < 20 links.
- This doesn't tell the whole story.

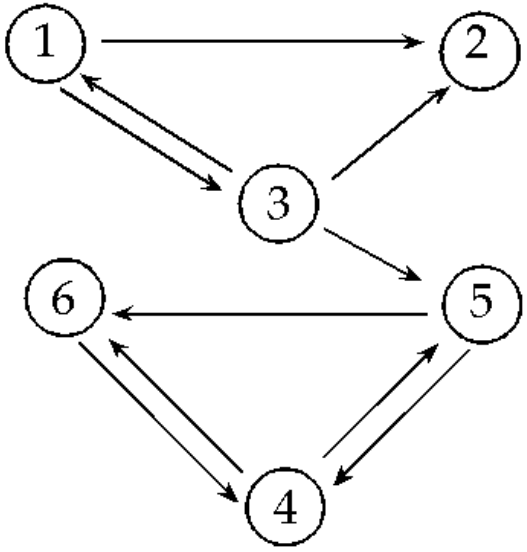
More Structure

- Graph has several distinct components.
- Strongly connected centre.
- In pages.
- Out pages.
- Tendrils.
- Disconnected networks.
- Spam/loops/link farms.

What Can This Tell Us?

- Designing crawl strategies on the web.
- Understanding the sociology of content creation on the web.
- Analysing the behavior of web algorithms that make use of link information (e.g., PageRank).
- Predicting the evolution of web structures.
- Predicting the emergence of important new phenomena in the web graph.
- Developing algorithms for discovering them.

Link Analysis



$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

- What can links tell us about a web page's potential relevance?

PageRank

- Given any particular query, we'd expect many matches.
- How do we find the most useful?
- PageRank orders web pages according to their impact.

PageRank

- Given any particular query, we'd expect many matches.
- How do we find the most useful?
- PageRank orders web pages according to their impact.
- A web page has high PageRank if it is linked to by pages with high PageRank.

Connectivity Matrix

- To measure PageRank we need a connectivity matrix of web, C.
- Label pages $p_1, p_2, \ldots, p_{20000000000}$.
- $c_{ij} = 1$ if p_i is linked to by p_j (otherwise 0).

Connectivity Matrix

- To measure PageRank we need a connectivity matrix of web, C.
- Label pages $p_1, p_2, \ldots, p_{20000000000}$.
- $c_{ij} = 1$ if p_i is linked to by p_j (otherwise 0).
- Scale C so all column sums are 1.

Connectivity Matrix

- To measure PageRank we need a connectivity matrix of web, C.
- Label pages $p_1, p_2, \ldots, p_{20000000000}$.
- $c_{ij} = 1$ if p_i is linked to by p_j (otherwise 0).
- Scale C so all column sums are 1.
- Must deal with pages with no outlinks

Random surf

- Measure PageRank by looking at a random surf.
- To move from a page, we choose one of the links at random.

Random surf

- Measure PageRank by looking at a random surf.
- To move from a page, we choose one of the links at random.
- Theory of Markov chains tells us* we'll reach a steady-state.
- PageRank is this steady-state.

Random surf

- Measure PageRank by looking at a random surf.
- To move from a page, we choose one of the links at random.
- Theory of Markov chains tells us* we'll reach a steady-state.
- PageRank is this steady-state.
- * Subject to terms and conditions.

Fudge Factor

- Our surf can get bogged down.
- We may get stuck in a part of the web with no way out.

Fudge Factor

- Our surf can get bogged down.
- We may get stuck in a part of the web with no way out.
- Build teleportation into the model.

Fudge Factor

- Our surf can get bogged down.
- We may get stuck in a part of the web with no way out.
- Build teleportation into the model.
- With probability p we choose a link.
- With probability $1 - p$ we choose a page completely at random.

Fudge Factor

- Our surf can get bogged down.
- We may get stuck in a part of the web with no way out.
- Build teleportation into the model.
- With probability p we choose a link.
- With probability $1 - p$ we choose a page completely at random.
- $p \approx .85$.
- Markov matrix is $A = pC + \frac{1-p}{n}E$, $e_{ij} = 1$.

Computing PageRank

- Solving systems with 20000000000 variables is challenging!
- Conventional methods require around $|V|^3$ computations.

Computing PageRank

- Solving systems with 20000000000 variables is challenging!
- Conventional methods require around $|V|^3$ computations.
- Efficient **iterative** techniques require $2k|E|$.
- k small, hopefully!

Computing Pagerank

- Want to find x such that
  $Ax = (pC + \frac{1-p}{n} ee^T)x = x$.
- Can be rewritten as

$$(I - pC)x = \frac{1-p}{n} ee^T x = \frac{1-p}{n} e.$$

- Iterative algorithms: Krylov subspace methods.
- E.g, power method: $x_{k+1} = Ax_k$.
- For x_0, use previously computed PageRank.
- Typically, need around 80 iterations.
- Computed monthly.

Influence of p

	$p = 0.85$	0.9	0.95	0.99
United States	1	1	1	1
Race (US Census)	2	2	4	20
UK	3	3	2	2
France	4	4	5	7
2005	5	5	11	10
Cat: politics	13	7	6	5
Cat:wikiportals	18	8	3	3

PageRank Today

- Victim of its own success: spammers target PageRank, Google bombs.
- Optimal outlink strategy: point to top page that points to you.
- Google uses several factors in its current ranking.
- Google claim PageRank is at its core.

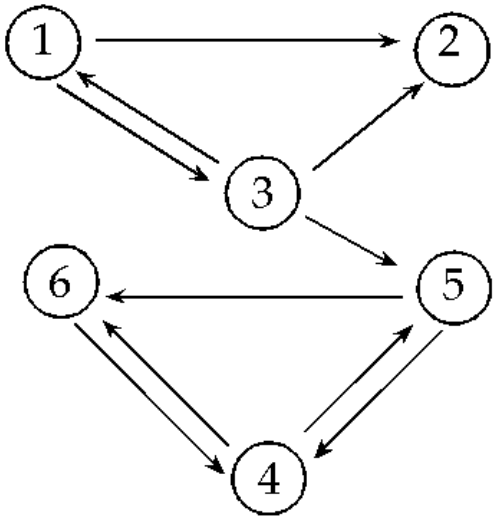
HITS

- Introduced by Kleinberg in 1999.
- Pages can be ranked according to their authority and "hubness".
- Good hubs point to good authorities and vice versa.
- Given a matrix of links, L, $a_k = Lh_{k-1}$, $h_k = L^T a_k$.
- Need to find dominant eigenvectors of LL^T and $L^T L$.

Equilibration

- Let C be matrix of graph of web links.
- Find diagonal matrices D_1 and D_2 so that $P = D_1 C D_2$ is doubly stochastic.
- Clearly, stationary distribution of P tells us nothing.
- Let $r = \mathbf{diag}(D_1)$, $c = \mathbf{diag}(D_2)$.
- Claim: authoritativeness of node i is inversely proportional to c_i, "hubness" is inversely proportional to r_i.
- A teleport is necessary.

Comparison of Algorithms



- PageRank ($p = .9$)
- HITS
- SK ($p = 1/60$).

- PageRanking: $4, 6, 5, 2, 3, 1$.
- HITS: Authorities $5, 2, 6, 1, 4, 3$. Hubs $3, 4, 1, 5, 6, 2$.
- Equilibration: A $4, 6, 5, 2, 3, 1$. H $3, 1, 4, 5, 6, 2$.

Example: Wikipedia Ratings

United States	2000	Political parties
Race (US Census)	Marriage	Environment topics
United Kingdom	US	State leaders
France	2003, 2004, 2005	Airlines
2005,2004,2000	UK/England	2 letter combinations
Canada	Canada	Masts
England	Japan	Mathematicians
Cat. by country	Australia	Peerage of the UK
2003	2001, 2002	Record labels
Cat.:Culture	Germany	Biblical names

Ranking factors

Keyword in title/inlinks/body	4.9/4.4/3.7
PageRank of site/page/inlinks	4.4/4/3.6
"Hubness"	3.5
Age of site/page/inlinks	4.1/3.4/3.2
Keyword semantics	3.4
Freshness	
Number of inlinks	
Link from authority	

Improving search

- Can we search the “invisible“ web?
- Can results be personalised/localised?
- Can we deal with synonyms and homonyms?
- How do we order information that’s very new or generated dynamically?

Improving search

- Can we search the "invisible" web?
- Can results be personalised/localised?
- Can we deal with synonyms and homonyms?
- How do we order information that's very new or generated dynamically?

The Goal: Perfect Search

- Ask any question and get **your** perfect answer.

Text Searching

- Web crawls generate vast quantities of data.
- Need to do some data mining.
- Given a search string, how do we find matching pages?
- How do we cope with vagaries of language?

Latent Semantic Indexing

- In literal matching, we look for exact matches of search strings.
- In latent semantic indexing, we attempt to match contextually.
- For example...“maths” and “mathematics”.
- First prepare data: index, eliminate, stem (maths = mathematics = math).
- Then match queries and contextualise.

Some Linear Algebra

- Suppose our search string contains m terms $s_1, s_2, \ldots, s_m$.
- Represent each web page as a vector $w^{(i)} \in \mathbb{R}^m$.
- $w_j^{(i)} = 1$ if s_j appears in web page.
- Otherwise $w_j^{(i)} = 0$.
- Gives an $m \times n$ matrix W and an n vector s.
- SVD: $W = USV^T$, $U^TU = I$, $V^TV = I$, $S = \mathbf{diag}(\sigma_1, \sigma_2, \ldots, \sigma_m)$.

Stopping

a a's able about above according accordingly across actually after afterwards again against ain't all allow allows almost alone along already also although always am among amongst an and another any anybody anyhow anyone anything anyway anyways anywhere apart appear appreciate appropriate are aren't around as aside ask asking associated at available away awfully b be became because become becomes becoming been before beforehand behind being believe below beside besides best better between beyond both brief but by c c'mon c's came can can't cannot cant cause causes certain certainly...

Query matching

- Find columns of W so that $\|w^{(i)} - s\| < tol$.
- Least squares problem.
- Solve with SVD.

Contextualisation

- Matrix W is huge.
- Want to get hold of its "essence".
- Approximate with something a whole lot smaller.
- Use the SVD.
- $W_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$, $k \ll m$.
- Can give better results than full W.

Example

baby, child, guide, health, home, infant, proofing, safety, toddler

- Infant and toddler first aid
- Babies and children's room (for your home)
- Child safety at home
- Your baby's health and safety: from infant to toddler
- Baby proofing basics
- Your guide to easy rust proofing
- Beanie babies collector's guide

Example

- Searching abstracts of papers for relevant research.
- With $\boldsymbol{W} \in \mathbb{R}^{7519 \times 200}$ we're wrong 5% of the time.
- $\boldsymbol{W}_5$ reduces errors to 2%.