Chapter 5: Link Analysis for Authority Scoring

5.3 Comparison and Extensions

Literature contains plethora of variations on Page-Rank and HITS

Key points are:

- mutual reinforcement between hubs and authorities
- re-scale edge weights (normalization)

Unified notation (for link graph with n nodes):

L	- $n \times n$ link matrix, $L_{ij} = 1$ if there is an edge (i,j), 0 else
din	- $n \times 1$ vector with din_i = indegree(i), $Din_{n \times n}$ = diag(din)
dout	- $n \times 1$ vector with $dout_i$ = outdegree(i), $Dout_{n \times n}$ = diag(dout)
x	- $n \times 1$ authority vector
y	- $n \times 1$ hub vector
Iop	- operation applied to incoming links
Oop	- operation applied to outgoing links

HITS and PageRank in Unified Framework

HITS: $x = Iop(y)$, $y = Oop(x)$ with $Iop(y) = L^T y$, $Oop(x) = Lx$

PageRank: $x = Iop(x)$ with $Iop(x) = P^T x$ with $P^T = L^T \, Dout^{-1}$

or $P^T = \alpha L^T \, Dout^{-1} + (1-\alpha) \, (1/n) \, e \, e^T$

SALSA (PageRank-style computation with mutual reinforcement):

$x = Iop(y)$ with $Iop(y) = P^T y$ with $P^T = L^T \, Dout^{-1}$

$y = Oop(x)$ with $Oop(x) = Q x$ with $Q = L \, Din^{-1}$

and other models of link analysis can be cast into this framework, too

A Familiy of Link Analysis Methods

General scheme: $\text{Iop}(\cdot) = \text{Din}^{-p}\, L^T\, \text{Dout}^{-q}\, (\cdot)$ and $\text{Oop}(\cdot) = \text{Iop}^T\, (\cdot)$

Specific instance ***Out-link normalized Rank (Onorm-Rank)***:
$\text{Iop}(\cdot) = L^T\, \text{Dout}^{-1/2}\, (\cdot)$, $\text{Oop}(\cdot) = \text{Dout}^{-1/2}\, L\, (\cdot)$
applied to x and y: $x = \text{Iop}(y)$, $y = \text{Oop}(x)$

In-link normalized Rank (Inorm-Rank):
$\text{Iop}(\cdot) = \text{Din}^{-1/2}\, L^T\, (\cdot)$, $\text{Oop}(\cdot) = L\, \text{Din}^{-1/2}\, (\cdot)$

Symmetric normalized Rank (Snorm-Rank):
$\text{Iop}(\cdot) = \text{Din}^{-1/2}\, L^T\, \text{Dout}^{-1/2}\, (\cdot)$, $\text{Oop}(\cdot) = \text{Dout}^{-1/2}\, L\, \text{Din}^{-1/2}\, (\cdot)$

Some properties of Snorm-Rank:
$x = \text{Iop}(y) = \text{Iop}(\text{Oop}(x)) \rightarrow \lambda x = A^{(S)}\, x$
with $A^{(S)} = \text{Din}^{-1/2}\, L^T\, \text{Dout}^{-1}\, L\, \text{Din}^{-1/2}$
$\rightarrow$ Solution: $\lambda = 1$, $x = \text{din}^{1/2}$
and analogously for hub scores: $\lambda y = H^{(S)}\, y \rightarrow \lambda = 1$, $y = \text{dout}^{1/2}$

Experimental Results

Construct neighborhood graph from result of query "star"
Compare authority-scoring ranks

HITS	OnormRank	PageRank
1 www.starwars.com	www.starwars.com	www.starwars.com
2 www.lucasarts.com	www.lucasarts.com	www.lucasarts.com
3 www.jediknight.net	www.jediknight.net	www.paramount.com
4 www.sirstevesguide.com	www.paramount.com	www.4starads.com/roman
5 www.paramount.com	www.sirstevesguide.com	www.starpages.net
6 www.surfthe.net/swma/	www.surfthe.net/swma/	www.dailystarnews.com
7 insurrection.startrek.com	insurrection.startrek.com	www.state.mn.us
8 www.startrek.com	www.fanfix.com	www.star-telegram.com
9 www.fanfix.com	shop.starwars.com	www.starbulletin.com
10 www.physics.usyd.edu.au/.../starwars	www.physics.usyd.edu.au/.../starwars	www.kansascity.com
		...
		19 www.jediknight.net
		21 insurrection.startrek.co
		23 www.surfthe.net/swma

Bottom line:
Differences between all kinds of authority ranking methods are fairly minor !

More LAR (Link Analysis Ranking) Methods

HubAveraging (similar to ONorm for hubs):

$$a(q) = \sum_{p \in IN(q)} h(p) \qquad h(p) = \frac{1}{|OUT(p)|} \sum_{q \in OUT(p)} a(q)$$

AuthorityThreshold (only k best authorities per hub):

$$a(q) = \sum_{p \in IN(q)} h(p) \qquad h(p) = \frac{1}{k} \sum_{q \in OUT-k(p)} a(q)$$

with $\mathrm{OUT-k}(p) = \mathrm{argmax-k}_q \{a(q) \mid q \in OUT(p)\}$

Max (AuthorityThreshold with k=1):

$$a(q) = \sum_{p \in IN(q)} h(p) \qquad h(p) = a(\mathrm{argmax}_q \{a(q) \mid q \in OUT(p)\})$$

BreadthFirstSearch (transitive citations up to depth k):

$$a(q) = \sum_{j=1}^{k} \left(\frac{1}{2}\right)^{j-1} |N^{(j)}(q)|$$

where $N^{(j)}(q)$ are nodes that have a path to q by alternating o OUT and i IN steps with j=o+i

LAR as Bayesian Learning

Postulate prob. model for p → q: $P[p \to q] = \frac{\exp(h_p a_q + e_p)}{1 + \exp(h_p a_q + e_p)}$

with parameters θ = (h_1, ..., h_n, a_1, ..., a_n, e_1, ..., e_n)

Postulate prior f(θ) for parameters θ:
normal distr. (μ,σ) for each e_i, exponential distr. (λ=1) for each a_i, h_i

Posterior f(θ|G) for links i →j ∈ G: $f(\theta \mid G) \sim f(G \mid \theta) f(\theta)$

Theorem:

$$f(\theta \mid G) \sim \Pi_{i=1..n} e^{-h_i - a_i - (e_i - \mu)^2 / 2\sigma^2} \cdot \Pi_{(i,j) \in G} e^{a_j h_i + e_i} / \Pi_{i,j} (1 + e^{a_j h_i + e_i})$$

Estimate $\hat{\theta} := E[\theta \mid G]$ **using numerical algorithms**

Alternative simpler model: $P[p \to q] = \frac{h_p a_q}{1 + h_p a_q}$

LAR Quality Measures: Score Distances

Consider two n-dimensional authority score vectors a and b

d_1 distance: $d_1(a,b) = \min_{\alpha,\beta \geq 1} \sum_{i=1..n} | \alpha a_i - \beta b_i |$

with scaling weights α, β to compensate normalization distortions

could alternatively use Lq norm rather than L1

LAR Quality Measures: Rank Distances

Consider top-k of two rankings $\tau 1$ and $\tau 2$ or full permutations of 1..n

- ***overlap similarity OSim*** $(\tau 1, \tau 2) = | \text{top}(k,\tau 1) \cap \text{top}(k,\tau 2) | / k$
- ***Kendall's τ measure KDist*** $(\tau 1, \tau 2) =$

$$\frac{|\{(u,v) \mid u,v \in U, u \neq v, \text{ and } \tau 1, \tau 2 \text{ disagree on relative order of } u,v\}|}{|U| \cdot (|U|-1)}$$

with U = top(k,τ1) $\cup$ top(k,τ2) (with missing items set to rank k+1)

with ties in one ranking and order in the other, count p with $0 \leq p \leq 1$
$\rightarrow$ p=0: weak KDist, $\rightarrow$ p=1: strict KDist

- ***footrule distance Fdist***$(\tau 1, \tau 2) = \frac{1}{|U|} \sum_{u \in U} | \tau 1(u) - \tau 2(u) |$

(normalized) Fdist is upper bound for KDist
and Fdist/2 is lower bound

LAR Similarity

Two LAR algorithms A and B are **similar on the class $\mathcal{G}$** of graphs with n nodes **under authority distance measure d** if

for $n \to \infty$: **$\max \{d(A(G),B(G)) \mid G \in \mathcal{G}\} = o(M_n(d,L_q))$**

where $M_n(d,L_q)$ is the maximum distance under d for any two n-dimensional vectors x and y that have L_q norm 1 (which is $\Theta(n^{1-1/q})$ for d_1 distance and L_q norm)

Two LAR algorithms A and B are **weakly (strictly) rank-similar on the class $\mathcal{G}$** of graphs with n nodes under **weak (strict) rank distance r** if for $n \to \infty$: **$\max \{r(A(G),B(G)) \mid G \in \mathcal{G}\} = o(1)$**

Theorems:

SALSA and Indegree are similar and strictly rank-similar.

No other LAR algorithms are known to be similar or weakly rank-similar.

LAR Stability

For graphs G=(V,E) and G‘=(V,E‘) the **link distance d_{link}** is:

$$\mathbf{d_{link}(G,G') = |(E \cup E') - (E \cap E')|}$$

For graph $G \in \mathcal{G}$, we define $C_k(G) = \{G' \in \mathcal{G} \mid d_{link}(G,G') \leq k\}$

LAR algorithm A is **stable** on the class $\mathcal{G}$ of graphs with n nodes under authority distance measure d if for every k > 0 for $n \to \infty$:

$$\mathbf{\max \{d(A(G),A(G')) \mid G \in \mathcal{G},\ G' \in C_k(G)\} = o(M_n(d,L_q))}$$

LAR algorithm A is **weakly (strictly) rank-stable** on the class $\mathcal{G}$ of graphs with n nodes under weak (strict) rank distance r if for every k > 0 for $n \to \infty$:

$$\mathbf{\max \{r(A(G),A(G')) \mid G \in \mathcal{G},\ G' \in C_k(G)\} = o(1)}$$

Theorems:

Indegree is stable.

No other LAR algorithm is known to be stable or weakly rank-stable (but some are under modified stability definitions).

PageRank is stable with high probability for power-law graphs.

LAR Experimental Comparison: Queries

Table I. Query Statistics

Query	Nodes	Hubs	Authorities	Links	Med out	Avg out	ACC size	Users
abortion	3340	2299	1666	22287	3	9.69	1583	22
affirmative action	2523	1954	4657	866	1	2.38	752	7
alcohol	4594	3918	1183	16671	2	4.25	1124	8
amusement parks	3410	1893	1925	10580	2	5.58	1756	8
architecture	7399	5302	3035	36121	3	6.81	3003	8
armstrong	3225	2684	889	8159	2	9.17	806	8
automobile industries	1196	785	561	3057	2	3.89	443	7
basketball	6049	5033	1989	24409	3	4.84	1941	12
blues	5354	4241	1891	24389	2	5.75	1838	8
cheese	3266	2700	1164	11660	2	4.31	1113	5
classical guitar	3150	2318	1350	12044	3	5.19	1309	8
complexity	3564	2306	1951	13481	2	5.84	1860	4
computational complexity	1075	674	591	2181	2	3.23	497	4
computational geometry	2292	1500	1294	8189	3	5.45	1246	3
death penalty	4298	2659	2401	21956	3	8.25	2330	9
genetic	5298	4293	1732	19261	2	4.48	1696	7
geometry	4326	3164	1815	13363	2	4.22	1742	7
globalization	4334	2809	2135	17424	2	8.16	1965	5
gun control	2955	2011	1455	11738	3	5.83	1334	7
iraq war	3782	2604	1860	15373	3	5.90	1738	8
jaguar	2820	2268	936	8392	2	3.70	846	5
jordan	4009	3355	1061	10937	2	3.25	991	4
moon landing	2188	1316	1179	5597	2	4.25	623	8
movies	7967	6624	2573	28814	2	4.34	2409	10
national parks	4757	3968	1260	14156	2	3.56	1112	6
net censorship	2598	1618	1474	7888	2	4.87	1375	4
randomized algorithms	742	502	341	1205	1	2.40	259	5
recipes	5243	4375	1508	18152	2	4.14	1412	10
roswell	2790	1973	1303	8487	2	4.30	1186	4
search engines	11659	7577	6209	292236	5	38.56	6157	5
shakespeare	4383	3660	1247	13575	2	3.70	1199	6
table tennis	1948	1489	803	5465	2	3.67	745	6
weather	8011	6464	2852	34672	3	5.36	2775	9
vintage cars	3460	2044	1920	12796	3	6.26	1580	5

Experimental setup:

- **34 queries**
- **rootsets of 200 pages each obtained from Google**
- **basesets computed using Google with first 50 predecessors per page**

Source: Borodin et al., ACM TOIT 2005

LAR Experimental Comparison: Precision@10

Table III. Relevance Ratio

Query	Hits	PageRank	InDegree	Salsa	HubAvg	Max	AT-med	AT-avg	BFS	Bayesian	SBayesian
abortion	90%	70%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	90%	**100%**
affirmative action	70%	50%	50%	50%	10%	10%	10%	10%	**80%**	40%	50%
alcohol	**90%**	60%	**90%**	**90%**	**90%**	80%	80%	80%	**90%**	80%	**90%**
amusement parks	**100%**	30%	30%	50%	0%	90%	10%	0%	80%	**100%**	30%
architecture	10%	**70%**	**70%**	**70%**	10%	60%	**70%**	10%	60%	10%	**70%**
armstrong	20%	**50%**	20%	20%	20%	20%	20%	20%	**50%**	20%	20%
automobile industries	10%	10%	20%	30%	10%	10%	10%	10%	**60%**	20%	20%
basketball	0%	70%	20%	20%	0%	10%	10%	10%	**100%**	10%	20%
blues	60%	**80%**	60%	60%	70%	60%	70%	70%	50%	60%	60%
cheese	0%	20%	30%	30%	10%	0%	0%	10%	**50%**	0%	30%
classical guitar	**90%**	50%	70%	70%	50%	80%	50%	50%	**90%**	70%	70%
complexity	0%	50%	50%	50%	0%	**90%**	**90%**	0%	80%	0%	50%
computational complexity	90%	70%	90%	90%	90%	90%	90%	90%	90%	**100%**	90%
computational geometry	**100%**	40%	70%	70%	70%	**100%**	70%	70%	**100%**	80%	70%
death penalty	**100%**	70%	90%	90%	70%	**100%**	**100%**	**100%**	**100%**	**100%**	90%
genetic	**100%**	70%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	90%	**100%**	**100%**
geometry	**90%**	20%	**90%**	**90%**	**90%**	**90%**	**90%**	80%	**90%**	80%	**90%**
globalization	**100%**	70%	90%	90%	**100%**	**100%**	**100%**	**100%**	90%	**100%**	90%
gun control	0%	50%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	80%	**100%**
iraq war	40%	30%	30%	30%	10%	20%	20%	10%	**90%**	40%	30%
jaguar	0%	**30%**	0%	0%	0%	0%	0%	0%	10%	0%	0%
jordan	0%	30%	30%	30%	40%	**100%**	**100%**	**100%**	40%	30%	30%
moon landing	0%	30%	20%	20%	0%	0%	0%	0%	**100%**	0%	20%
movies	10%	20%	50%	40%	50%	**70%**	**70%**	**70%**	60%	60%	50%
national parks	0%	50%	10%	10%	**80%**	**80%**	**80%**	0%	70%	0%	10%
net censorship	0%	30%	80%	80%	60%	**90%**	**90%**	**90%**	80%	70%	80%
randomized algorithms	70%	**80%**	**80%**	**80%**	40%	50%	50%	50%	60%	**80%**	70%
recipes	0%	20%	70%	70%	30%	90%	90%	**100%**	80%	80%	70%
roswell	0%	20%	40%	40%	**70%**	**70%**	60%	0%	60%	10%	40%
search engines	80%	90%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	90%	80%	**100%**
shakespeare	**100%**	70%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
table tennis	90%	60%	**100%**	**100%**	90%	90%	90%	90%	90%	90%	**100%**
weather	80%	50%	80%	80%	60%	80%	80%	80%	**90%**	80%	80%
vintage cars	20%	10%	60%	60%	20%	60%	60%	20%	**70%**	40%	60%
avg	47%	48%	61%	62%	51%	67%	64%	54%	**78%**	56%	61%
max	**100%**	90%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
min	0%	**10%**	0%	0%	0%	0%	0%	0%	**10%**	0%	0%
stdev	43%	23%	31%	31%	38%	36%	36%	42%	21%	37%	31%

Source: Borodin et al., ACM TOIT 2005

LAR Experimental Comparison: Key Authorities

Table II. High Relevance Ratio

Query	HITS	PAGERANK	INDEGREE	SALSA	HUBAVG	MAX	AT-MED	AT-AVG	BFS	BAYESIAN	SBAYESIAN
abortion	30%	10%	**40%**	**40%**	30%	**40%**	30%	30%	30%	30%	**40%**
affirmative action	30%	0%	40%	40%	0%	0%	0%	0%	**60%**	30%	40%
alcohol	60%	30%	60%	60%	60%	50%	50%	50%	**70%**	50%	60%
amusement parks	50%	10%	30%	40%	0%	70%	10%	0%	40%	**90%**	30%
architecture	0%	30%	**70%**	**70%**	0%	60%	60%	0%	50%	0%	**70%**
armstrong	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
automobile industries	0%	10%	10%	20%	0%	0%	0%	0%	**40%**	0%	10%
basketball	0%	**60%**	20%	20%	0%	10%	10%	10%	**60%**	10%	20%
blues	**60%**	40%	40%	40%	50%	50%	50%	50%	20%	40%	40%
cheese	0%	0%	0%	0%	0%	0%	0%	0%	**10%**	0%	0%
classical guitar	**40%**	30%	30%	30%	0%	**40%**	0%	0%	**40%**	30%	30%
complexity	0%	30%	20%	20%	0%	**70%**	50%	0%	50%	0%	20%
computational complexity	30%	30%	30%	30%	**40%**	30%	30%	30%	20%	30%	30%
computational geometry	40%	20%	40%	40%	40%	40%	**50%**	40%	40%	30%	40%
death penalty	70%	30%	70%	70%	50%	**80%**	**80%**	**80%**	**80%**	**80%**	60%
genetic	**80%**	20%	70%	70%	60%	70%	70%	70%	60%	**80%**	70%
geometry	60%	10%	50%	50%	40%	**70%**	**70%**	60%	60%	60%	50%
globalization	0%	**30%**	20%	20%	0%	0%	0%	0%	**30%**	0%	20%
gun control	0%	50%	**70%**	**70%**	60%	60%	60%	60%	60%	50%	**70%**
iraq war	0%	10%	10%	10%	0%	10%	0%	0%	**40%**	10%	10%
jaguar	0%	**20%**	0%	0%	0%	0%	0%	0%	10%	0%	0%
jordan	0%	10%	20%	20%	20%	**40%**	**40%**	**40%**	30%	20%	20%
moon landing	0%	20%	10%	10%	0%	0%	0%	0%	**80%**	0%	10%
movies	10%	10%	30%	30%	40%	**70%**	**70%**	**70%**	40%	50%	30%
national parks	0%	50%	10%	10%	**60%**	**60%**	**60%**	0%	50%	0%	10%
net censorship	0%	20%	**80%**	**80%**	60%	**80%**	**80%**	**80%**	**80%**	70%	**80%**
randomized algorithms	0%	**40%**	10%	10%	0%	0%	0%	0%	10%	10%	10%
recipes	0%	10%	60%	60%	10%	60%	60%	**70%**	50%	50%	60%
roswell	0%	0%	0%	0%	0%	**10%**	**10%**	0%	**10%**	0%	0%
search engines	60%	70%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	90%	50%	**100%**
shakespeare	0%	20%	50%	50%	50%	**70%**	**70%**	**70%**	60%	50%	50%
table tennis	**50%**	20%	**50%**	**50%**	**50%**	**50%**	**50%**	**50%**	40%	40%	**50%**
weather	60%	20%	60%	60%	30%	60%	50%	50%	**70%**	60%	60%
vintage cars	0%	0%	**40%**	**40%**	0%	**40%**	**40%**	0%	30%	10%	**40%**
avg	21%	22%	36%	37%	25%	41%	37%	30%	**44%**	30%	36%
max	80%	70%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	90%	90%	**100%**
min	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
stdev	27%	17%	26%	26%	28%	30%	31%	32%	23%	28%	26%

Is there a winner at all?

Source: Borodin et al., ACM TOIT 2005

LAR Results for Query „Classical Guitar“ (1)

HITS	PAGERANK	INDEGREE
1. (1.000) **earlyromanticguitar.com URL:www.earlyromanticguitar.com**	1. (1.000) Take Note Publishing Limited www.takenote.co.uk	1. (1.000) Guitar Alive-GuitarAlive—www.guitaralive.com
2. (0.927) **Empty title field URL: classicalguitar.freehosting.ne**	2. (0.724) *Guitar music, guitar books a URL:www.booksforguitar.com*	2. (0.895) **earlyromanticguitar.com URL:www.earlyromanticguitar.com**
3. (0.889) *Adirondack Spruce.com URL:adirondackspruce.com*	3. (0.588) **Registry of Guitar Tutors URL:www.registryofguitartutors.com**	3. (0.889) **Empty title field URL:www.guitarfoundation.org**
4. (0.766) *The Classical Guitar Homepag URL:www.ak-c.demon.nl*	4. (0.528) Guitar Alive-GuitarAlive—www.guitaralive.com	4. (0.882) *Hitsquad.com-Musicians Web URL:www.hitsquad.com*
5. (0.732) Guitar Alive-GuitarAlive—www.guitaralive.com	5. (0.416) *Hitsquad.com-Musicians Web URL:www.hitsquad.com*	5. (0.850) Hitsquad Privacy Policy www.hitsquad.com/privacy.shtml
6. (0.681) **Empty title field URL:www.guitarfoundation.org**	6. (0.413) Hitsquad Privacy Policy www.hitsquad.com/privacy.shtml	6. (0.850) Advertising on Hitsquad Musi www.hitsquad.com/advertising.s
7. (0.676) *GUITAR REVIEW URL:www.guitarreview.com*	7. (0.413) Advertising on Hitsquad Musi www.hitsquad.com/advertising.s	7. (0.784) *Adirondack Spruce.com URL:adirondackspruce.com*
8. (0.644) *Avi Afriat—Classical guita URL:afriat.tripod.com*	8. (0.387) **Empty title field URL: www.guitarfoundation.org**	8. (0.778) **Empty title field URL: classicalguitar.freehosting.ne**
9. (0.605) **The Classical Guitar Home Pa URL:www.guitarist.com/cg/cg.htm**	9. (0.343) **Guitar Foundation of America URL:64.78.54.231**	9. (0.765) *Empty title field URL: www.vicnet.net.au/simieasyjamn*
10. (0.586) *Empty title field URL:www.duolenz.com*	10. (0.322) Vivendi Universal www.vivendiuniversal.com	10. (0.739) *The Classical Guitar Homepag URL:www.ak-c.demon.nl*

Source: Borodin et al., ACM TOIT 2005

LAR Results for Query „Classical Guitar“ (2)

HubAvg	Max	AT-med
1. (1.000) *Hitsquad.com-Musicians Web URL:www.hitsquad.com*	1. (1.000) Guitar Alive-GuitarAlive—www.guitaralive.com	1. (1.000) *Hitsquad.com-Musicians Web URL:www.hitsquad.com*
2. (0.995) Hitsquad Privacy Policy www.hitsquad.com/privacy.shtml	2. (0.619) **earlyromanticguitar.com URL:www.earlyromanticguitar.com**	2. (0.983) Hitsquad Privacy Policy www.hitsquad.com/privacy.shtml
3. (0.995) Advertising on Hitsquad Musi www.hitsquad.com/advertising.s	3. (0.506) **Empty title field URL:www.guitarfoundation.org**	3. (0.983) Advertising on Hitsquad Musi www.hitsquad.com/advertising.s
4. (0.856) *Empty title field URL: www.vicnet.net.au/simieasyjamn*	4. (0.451) *Adirondack Spruce.com URL:adirondackspruce.com*	4. (0.880) *Empty title field URL: www.vicnet.net.au/simieasyjamn*
5. (0.135) *AMG All Music Guide URL:www.allmusic.com*	5. (0.441) **Empty title field URL: classicalguitar.freehosting.ne**	5. (0.205) *AMG All Music Guide URL:www.allmusic.co* http://www.:
6. (0.132) Free Music Download, MP3 Mus ubl.com	6. (0.378) *GUITAR REVIEW URL:www.guitarreview.com*	6. (0.193) Free Music Download, MP3 Mus ubl.com
7. (0.130) *2000 Guitars Database URL:dargo.vicnet.net.au/guitar/lis*	7. (0.377) *The Classical Guitar Homepag URL:www.ak-cdemon.nl*	7. (0.169) *2000 Guitars Database URL:dargo.vicnet.net.au/guitar/lis*
8. (0.115) Guitar Alive-GuitarAlive—www.guitaralive.com	8. (0.371) **The Classical Guitar Home Pa URL:www.guitarist.com/cg/cg.htm**	8. (0.132) Guitar Alive-GuitarAlive—www.guitaralive.com
9. (0.096) CDNOW www.cdnow.com/from=sr-767167	9. (0.336) *Hitsquad.com-Musicians Web URL:www.hitsquad.com*	9. (0.129) CDNOW www.cdnow.com/from=sr-767167
10. (0.056) *OLGA—The On-Line Guitar Ar URL:www.olga.net*	10. (0.312) Hitsquad Privacy Policy www.hitsquad.com/privacy.shtml	10. (0.082) *OLGA—The On-Line Guitar Ar URL:www.olga.net*

Source: Borodin et al., ACM TOIT 2005

LAR Results for Query „Classical Guitar“ (3)

AT-AVG	BFS	BAYESIAN
1. (1.000) *Hitsquad.com-Musicians Web URL:www.hitsquad.com*	1. (1.000) **Empty title field URL: classicalguitar.freehosting.ne**	1. (3.66) **earlyromanticguitar. com URL:www. earlyromanticguitar.com**
2. (0.986) Hitsquad Privacy Policy www.hitsquad.com/privacy.shtml	2. (0.991) **earlyromanticguitar. com URL: www.**	2. (3.54) **Empty title field URL: classicalguitar.freehosting.ne**
3. (0.986) Advertising on Hitsquad Musi www.hitsquad.com/ advertising.s	3. (0.974) *Adirondack Spruce.com URL:adirondackspruce.com*	3. (3.53) *Adirondack Spruce.com URL:adirondackspruce.com*
4. (0.906) *Empty title field URL: www.vicnet.net.au / simieasyjamn*	4. (0.962) **Empty title field URL:www.guitarfoundation.org**	4. (3.42) *Hitsquad.com—Musicians Web URL:www.hitsquad.com*
5. (0.210) *AMG All Music Guide URL:www.allmusic.com*	5. (0.945) Guitar Alive-GuitarAlive—www.guitaralive.com	5. (3.38) Guitar Alive-GuitarAlive—www.guitaralive.com
6. (0.199) Free Music Download, MP3 Mus ubl.com	6. (0.933) *The Classical Guitar Homepag URL:www.ak-c.demon.nl*	6. (3.33) **Empty title field URL:www.guitarfoundation.org**
7. (0.179) *2000 Guitars Database URL:dargo.vicnet.net.au / guitar / lis*	7. (0.917) **The Classical Guitar Home Pa URL:www.guitarist. com/cg/cg.htm**	7. (3.33) Hitsquad Privacy Policy www.hitsquad.com/privacy.shtml
8. (0.135) CDNOW www.cdnow. com/from=sr-767167	8. (0.898) *Avi Afriat—Classical guita URL:afriat.tripod.com*	8. (3.33) Advertising on Hitsquad Musi www.hitsquad.com/ advertising.s
9. (0.122) Guitar Alive-GuitarAlive—www.guitaralive.com	9. (0.889) *GUITAR REVIEW URL:www.guitarreview.com*	9. (3.27) *The Classical Guitar Homepag URL:www.ak-c.demon.nl*
10. (0.080) *OLGA—The On-Line Guitar Ar URL:www.olga.net*	10. (0.881) *Empty title field URL:www.duolenz.com*	10. (3.25) *GUITAR REVIEW URL:www.guitarreview.com*

Source: Borodin et al., ACM TOIT 2005

5.4 Topic-specific PageRank [Haveliwala 2003]

Given: a (small) set of topics c_k, each with a set T_k of authorities
(taken from a directory such as ODP (www.dmoz.org)
or bookmark collection)

Key idea :
change the PageRank random walk by **biasing the random-jump probabilities** to the topic authorities T_k:

$$\vec{r}_k = \varepsilon\, \vec{p}_k + (1-\varepsilon) A' \vec{r}_k$$ with $A'_{ij} = 1/\text{outdegree}(j)$ for $(j,i) \in E$, 0 else

with $(p_k)_j = 1/|T_k|$ for $j \in T_k$, 0 else (instead of $p_j = 1/n$)

Approach:
1) Precompute topic-specific Page-Rank vectors r_k
2) Classify user query q (incl. query context) w.r.t. each topic c_k
   $\rightarrow$ probability $w_k := P[c_k \mid q]$
3) Total authority score of doc d is $\sum_k w_k\, r_k(d)$

Experimental Evaluation: Setup

Setup: based on Stanford WebBase (120 Mio. pages, Jan. 2001)

contains ca. 300 000 out of 3 Mio. ODP pages

considered 16 top-level ODP topics

link graph with 80 Mio. nodes of size 4 GB

on 1.5 GHz dual Athlon with 2.5 GB memory and 500 GB RAID

25 iterations for all 16+1 PR vectors took 20 hours

random-jump prob. ε set to 0.25 (could be topic-specific, too ?)

35 test queries: classical guitar, lyme disease, sushi, etc.

Quality measures: consider top k of two rankings $\tau 1$ and $\tau 2$ (k=20)

- ***overlap similarity OSim*** $(\tau 1,\tau 2) = |\, \text{top}(k,\tau 1) \cap \text{top}(k,\tau 2)\, | \,/\, k$
- ***Kendall's τ measure KSim*** $(\tau 1,\tau 2) =$

$$\frac{|\{(u,v)\,|\,u,v \in U, u \neq v, \textit{ and } \tau 1, \tau 2 \textit{ agree on relative order of } u,v\}|}{|U| \cdot (|U|-1)}$$

with $U = \text{top}(k,\tau 1) \cup \text{top}(k,\tau 2)$

Experimental Evaluation Results (1)

- Ranking similarities between most similar PR vectors:

	OSim	KSim
(Games, Sports)	0.18	0.13
(No Bias, Regional)	0.18	0.12
(Kids&Teens, Society)	0.18	0.11
(Health, Home)	0.17	0.12
(Health, Kids&Teens)	0.17	0.11

- User-assessed precision at top 10 (# relevant docs / 10) with 5 users:

	No Bias	Topic-sensitive
alcoholism	0.12	0.7
bicycling	0.36	0.78
death valley	0.28	0.5
HIV	0.58	0.41
Shakespeare	0.29	0.33
micro average	0.276	0.512

Experimental Evaluation Results (2)

- Top 3 for query "bicycling"
  (classified into sports with 0.52, regional 0.13, health 0.07)

	No Bias	Recreation	Sports
1	www.RailRiders.com	www.gorp.com	www.multisports.com
2	www.waypoint.org	www.GrownupCamps.com	www.BikeRacing.com
3	www.gorp.com	www.outdoor-pursuits.com	www.CycleCanada.com

- Top 5 for query context "blues" (user picks entire page)
  (classified into arts with 0.52, shopping 0.12, news 0.08)

	No Bias	Arts	Health
1	news.tucows.com	www.britannia.com	www.baltimorepsych.com
2	www.emusic.com	www.bandhunt.com	www.ncpamd.com/seasonal
3	www.johnholleman.com	www.artistinformation.com	www.ncpamd.com/Women's_M
4	www.majorleaguebaseball	www.billboard.com	www.wingofmadness.com
5	www.mp3.com	www.soul-patrol.com	www.countrynurse.com

Personalized PageRank

Goal: Efficient computation and efficient storage of user-specific personalized PageRank vectors (PPR)

PageRank equation: $\vec{r}_k = \varepsilon \vec{p}_k + (1-\varepsilon) A' \vec{r}_k$

Theorem:

Let u_1 and u_2 be personal preference vectors for random-jump targets, and let r_1 and r_2 denote the corresponding PPR vectors.
Then for all $\alpha_1, \alpha_2 \geq 0$ with $\alpha_1 + \alpha_2 = 1$ the following holds:
$\alpha_1 r_1 + \alpha_2 r_2 = (1-\varepsilon) A' (\alpha_1 r_1 + \alpha_2 r_2) + \varepsilon (\alpha_1 u_1 + \alpha_2 u_2)$

Corollary:

For preference vector u with m non-zero components
and base vectors e_p with $(e_p)_i = 1$ for i=p, 0 for i≠p, the following holds:

$$u = \sum_{p=1}^{m} \alpha_p \cdot e_p \qquad \text{with constants } \alpha_1 \ldots \alpha_m$$

and $$r = \sum_{p=1}^{m} \alpha_p \cdot r_p \qquad \text{for PPR vector r}$$

Exploiting Click Streams

Simple idea: Modify HITS or Page-Rank algorithm by weighting edges with the relative frequency of users clicking on a link (as observed by DirectHit, Alexa, etc.)

More sophisticated approach (Chen et al.:2002):

Consider link graph A and
link-visit matrix V (V_{ij}=1 if user i visits page j, 0 else)
Define

authority score vector: $a = \beta A^T h + (1- \beta)V^T u$
hub score vector: $h = \beta A a + (1- \beta)V^T u$
user importance vector: $u = (1- \beta)V(a+h)$

with a tunable parameter β (β=1: HITS, β=0: DirectHit)

claims to achieve higher precision than HITS, according to experimental results (with β=0.6) for some Webqueries such as „daily news“:

HITS top results: pricegrabber, gamespy, fileplanet, sportplanet, etc.
Chen et al. method: news.com, bbc, cnn, google, lycos, etc.

Link Analysis based on Implicit Links (1)

Apply simple data mining to browsing sessions of many users,
where each session i is a sequence (pi_1, pi_2, ...) of visited pages:

consider all pairs (pi_j, pi_{j+1}) of successively visited pages,
compute their total frequency f, and
selected those with f above some min-support threshold

Construct implicit-link graph with the selected page pairs as edges
and their normalized total frequencies f as edge weights.

Apply edge-weighted Page-Rank for authority scoring,
and linear combination of relevance and authority for overall scoring.

Link Analysis based on Implicit Links (2)

Experimental results (Xue et al.:2003):

performed on 4-month server-side (UC Berkeley) click-stream log with some „data cleaning“:

300 000 sessions of 60 000 users visiting 170 000 pages with 200 000 explicit links

2-item frequent itemset mining yields

336 812 implicit links (incl. 22 122 explicit links)

Results for query „vision“:

	implicit PR	explicit PR	weighted HITS	DirectHit
1	vision group	some paper	Forsyth‘s book	workshop on vision
2	Forsysth‘s book	vision group	vision group	some student‘s resume
3	book 3rd edition	student resume	book 3rd edition	special course
4	workshop on vision	some talk slides	Leung‘s publ.	Forsyth‘s book
...				

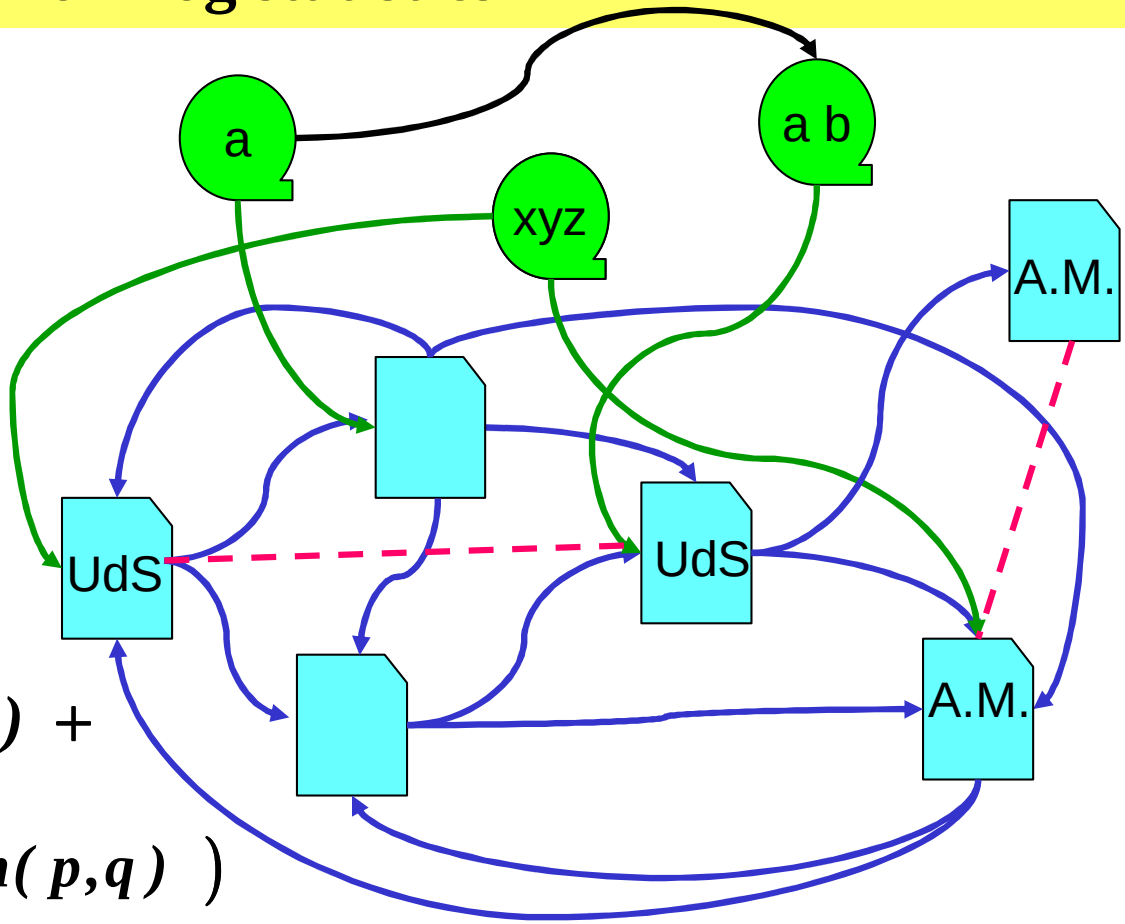
not clear to me
if any method is really better

Exploiting Query Logs and Click Streams

from PageRank: uniformly random choice of links + random jumps

to QRank: + query-doc transitions + query-query transitions
+ doc-doc transitions on implicit links (w/ thesaurus)
with probabilities estimated from log statistics

$$PR(q) = \varepsilon \cdot j(q) + (1-\varepsilon) \cdot \sum_{p \in IN(q)} PR(p) \cdot t(p,q)$$



$$QR(q) = \varepsilon \cdot j(q) + (1-\varepsilon) \cdot \Big(\alpha \sum_{p \in explicitIN(q)} PR(p) \cdot t(p,q) + (1-\alpha) \sum_{p \in implicitIN(q)} PR(p) \cdot sim(p,q) \Big)$$

Preliminary Experiments

Setup:

70 000 Wikipedia docs, 18 volunteers posing Trivial-Pursuit queries
ca. 500 queries, ca. 300 refinements, ca. 1000 positive clicks
ca. 15 000 implicit links based on doc-doc similarity

Results (assessment by blind-test users):

- **QRank top-10 result preferred over PageRank in 81% of all cases**
- **QRank has 50.3% precision@10, PageRank has 33.9%**

Untrained example query „philosophy“:

	PageRank	QRank
1.	Philosophy	Philosophy
2.	GNU free doc. license	GNU free doc. license
3.	Free software foundation	Early modern philosophy
4.	Richard Stallman	Mysticism
5.	Debian	Aristotle

5.5 Efficiency of PageRank Computation (1)

Speeding up convergence of the Page-Rank iterations

Solve **Eigenvector equation $\lambda x = Ax$**
(with dominant Eigenvalue $\lambda_1=1$ for ergodic Markov chain)
by power iteration: $x^{(i+1)} = Ax^{(i)}$ until $\|x^{(i+1)} - x^{(i)}\|_1$ is small enough

Write **start vector $x^{(0)}$ in terms of Eigenvectors $u_1, ..., u_m$**:
$x^{(0)} = u_1 + \alpha_2 u_2 + ... + \alpha_m u_m$
$x^{(1)} = Ax^{(0)} = u_1 + \alpha_2 \lambda_2 u_2 + ... + \alpha_m \lambda_m u_m$ with $\lambda_1 - |\lambda_2| = \varepsilon$ (jump prob.)
$x^{(n)} = A^n x^{(0)} = u_1 + \alpha_2 \lambda_2^n u_2 + ... + \alpha_m \lambda_m^n u_m$

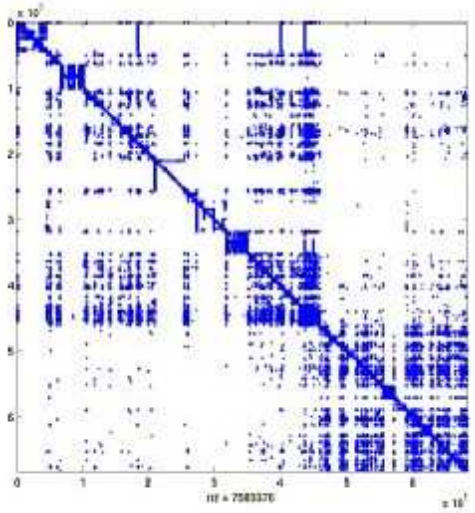
Aitken Δ^2 extrapolation:
assume $x^{(k-2)} \approx u_1 + \alpha_2 u_2$ (disregarding all "lesser" EVs)
$\rightarrow x^{(k-1)} \approx u_1 + \alpha_2 \lambda_2 u_2$ and $x^{(k)} \approx u_1 + \alpha_2 \lambda_2^2 u_2$
$\rightarrow$ after step k: solve for u_1 and u_2 and rewrite $x^{(k)}$ in terms of
$\Delta_k(u_1) = x^{(k)} - x^{(k-1)}$ and $\Delta_k^2(u_1) = \Delta_k(u_1) - \Delta_{k-1}(u_1)$
$\rightarrow x^{(k)*} := x^{(k)} - \Delta_k(u_1)^2 / \Delta_k^2(u_1)$ which converges faster than $x^{(k)}$

can be extended to quadratic extrapolation using first 3 EVs
speeds up convergence by factor of 0.3 to 3

Efficiency of PageRank Computation (2)

Exploit block structure of the link graph:

1) partitition link graph by domain names
2) compute local PR vector of pages within each block → LPR(i) for page i
3) compute block rank of each block:
   a) block link graph B with $B_{IJ} = \sum_{i \in I, j \in J} A_{ij} \cdot LPR(i)$
   b) run PR computation on B, yielding BR(I) for block I
4) Approximate global PR vector using LPR and BR:
   a) set $x_j^{(0)}$:= LPR(j) · BR(J) where J is the block that contains j
   b) run PR computation on A



(b) Stanford/Berkeley

speeds up convergence by factor of 2 in good "block cases"
unclear how effective it would be on Geocities, AOL, T-Online, etc.

Much adoo about nothing ?
Couldn't we simply initialize the PR vector with indegrees?

Efficiency of Storing PageRank Vectors

Memory-efficient encoding of PR vectors
(important for large number of topic-specific vectors)

16 topics * 120 Mio. pages * 4 Bytes would cost 7.3 GB

Key idea:
- map real PR scores to n cells and encode cell no into ceil($\log_2$ n) bits
- approx. PR score of page i is the mean score of the cell that contains i
- should use non-uniform partitioning of score values to form cells

Possible encoding schemes:
- ***Equi-depth partitioning***: choose cell boundaries such that $\sum_{i \in cell\ j} PR(i)$ is the same for each cell
- ***Equi-width partitioning with log values***: first transform all PR values into log PR, then choose equi-width boundaries
- Cell no. could be variable-length encoded (e.g., using Huffman code)

5.6 Online Page Importance

Goals:

- Compute Page-Rank-style authority measure online
  without having to store the complete link graph
- Recompute authority incrementally as the graph changes

Key idea:

- Each page holds some „cash“ that reflects its importance
- When a page is visited, it distributes its cash among its successors
- When a page is not visited, it can still accumulate cash
- This random process has a stationary limit
  that captures importance of pages

OPIC Algorithm
(Online Page Importance Computation)

Maintain for each page i (out of n pages):
C[i] – cash that page i currently has and distributes
H[i] – history of how much cash page has ever had in total
plus global counter
G – total amount of cash that has ever been distributed

```
for each i do { C[i] := 1/n; H[i] := 0 }; G := 0;
do forever {
   choose page i (e.g., randomly);
   H[i] := H[i] + C[i];
   for each successor j of i do C[j] := C[j] + C[i] / outdegree(i);
   G := G + C[i];
   C[i] := 0;
};
```

Note: 1) every page needs to be visited infinitely often (fairness)
2) the link graph L‘ is assumed to be strongly connected

OPIC Importance Measure

At each step t an estimate of the importance of page i is:
$(H_t[i] + C_t[i]) / (G_t + 1)$ (or alternatively: $H_t[i] / G_t$)

Theorem:
Let $X_t = H_t / G_t$ denote the vector of cash fractions accumulated by pages until step t.
The limit $X = \lim_{t\to\infty} X_t$ exists with $|X|_1 = \Sigma_i X[i] = 1$.

with crawl strategies such as:
- random
- greedy: read page i with highest cash C[i]
  (fair because non-visited pages accumulate cash until eventually read)
- cyclic (round-robin)

Adaptive OPIC for Evolving Link Graph

Consider „time“ window [now-T, now] where time is the value of G

Estimated importance of page i is: $X_{now}[i] = (H_{now}[i] - H_{now-T}[i]) / T$

For fixed window size T maintain for every page i:
$C_t[i]$ and G_t for each time t that i was visited within the window

For variable window size k maintain for every page i:
$C_t[i]$ and G_t for each time t of the last k visits of i

For two-point estimate (e.g., previous and current crawl):

- maintain two history vectors H[1..n] and G[1..n]
  for accumulated cash H[i] in current crawl
  and time G[i] of previous crawl
- set

$$H[i] := \begin{cases} H[i] \cdot \dfrac{T - (G - G[i])}{T} + C[i] & \text{if } G - G[i] < T \\ C[i] \cdot \dfrac{T}{G - G[i]} & \text{if } G - G[i] \geq T \end{cases}$$

Implementation and Experiments

essentially a crawler distributed across 4 PCs
with hash-based partitioning of URLs
uses Greedy-style crawl strategy and two-point interpolation

can track „accumulated cash“ of href targets without visiting them !

Web crawl visited 400 Mio. pages and
computed importance of 1 Bio. pages
over several months.

Web Dynamics

- large-scale study by [D. Fetterly et al.: WWW 2003]: weekly crawl of 150 Mio. pages over 11-week time period efficiently detecting significant changes (with high prob.) based on fingerprints of 5-word shingles
- sampling-based study by [A. Ntoulas et al.: WWW 2004]: weekly crawl of 154 sites over 1-year time period

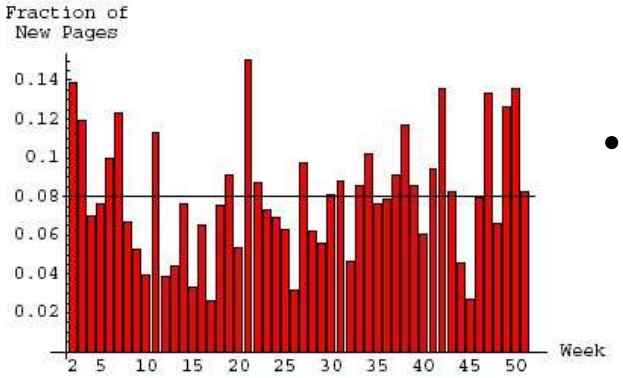


Figure 1: Fraction of new pages between successive snapshots.

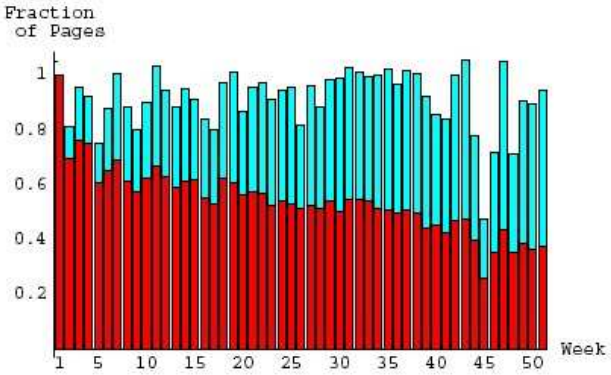


Figure 2: Fraction of pages from the first crawl still existing after n weeks (dark bars) and new pages (light bars).

each week 8% new Web pages
(additional or replacing old ones)
and 25% new links are created,
and 5% of existing pages are updated

Source: A. Ntoulas, J. Cho, C. Olston: WWW 2004

Time-aware Link Analysis

Motivation:

What's important and new on the Web?

(or was important as of ..., or is gaining importance ...)

Main technique:

use time information to bias transition and jump prob's in PR

$$TR(q) = \varepsilon \cdot j(q) + (1-\varepsilon) \cdot \sum_{p \in IN(q)} TR(p) \cdot t(p,q)$$

with j and t dependent on freshness and activity of nodes and edges:

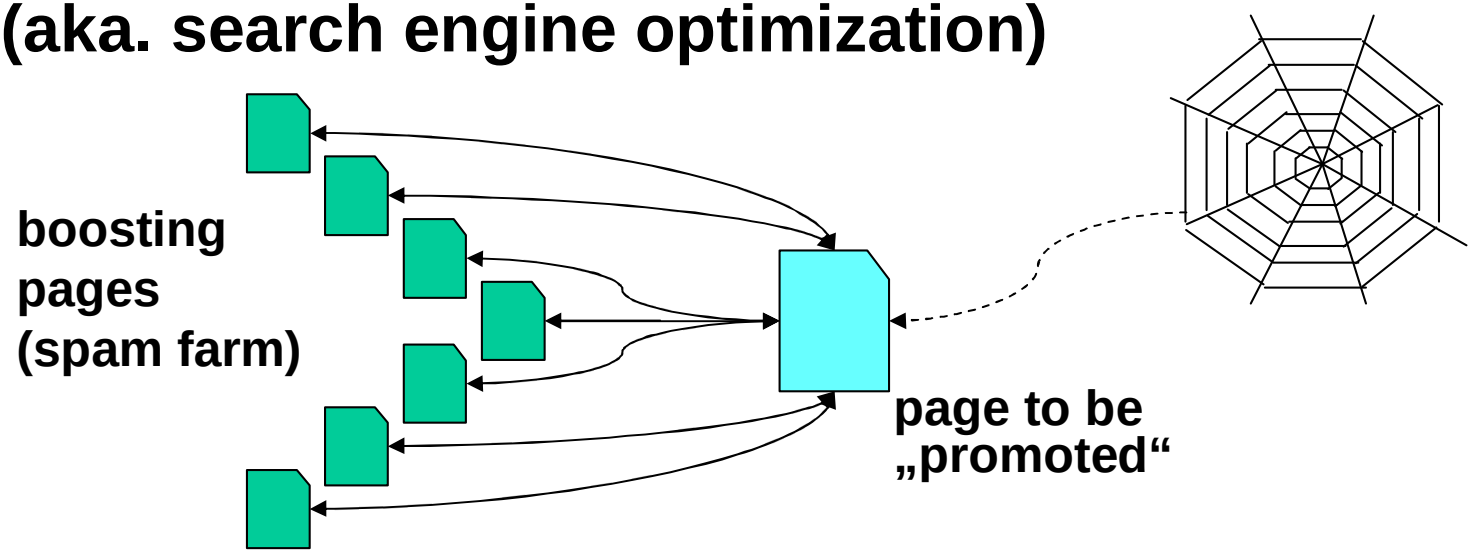
$$t(x,y) = \alpha \cdot f(y) + \beta \cdot f(x,y) + \gamma \cdot avg\{f(v,y) \mid v \in IN(y)\}$$

Some experimental results:

- **on 200 000 Web pages about Olympic Games, crawled 9 times between July 26 and Sept 1, 2004**
- **blind-user preference tests for 10 queries, e.g.: „summer olympics", „olympic torch relay", „Ian Thorpe"**
- **TR ranking mostly preferred over PR ranking**

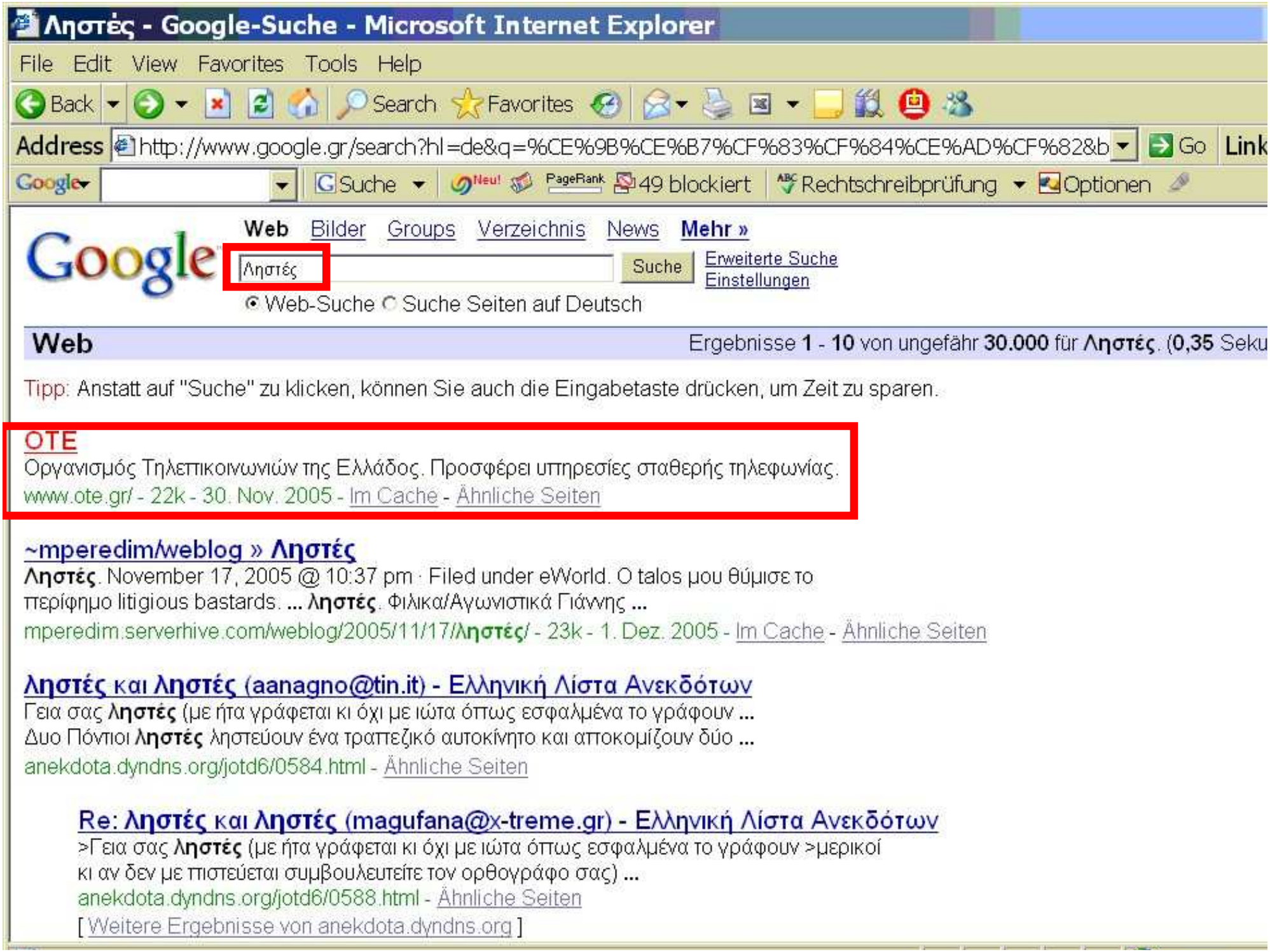
5.7 Spam: Not Just for E-mail Anymore

Distortion of search results by „spam farms“ (aka. search engine optimization)



Susceptibility to manipulation and lack of trust model is a major problem:

- **2004 DarkBlue SEO Challenge: „*nigritude ultramarine*“ extremely „successful“**
- **Pessimists estimate 75 Mio. out of 150 Mio. Web hosts are spam**
- **Recent example: Ληστές**
  http://www.google.gr/search?hl=el&q=%CE%BB%CE%B7%CF%83%CF%84%CE%AD%C
  unclear borderline between spam and community opinions


Ληστές - Google-Suche - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www.google.gr/search?hl=de&q=%CE%9B%CE%B7%CF%83%CF%84%CE%AD%CF%82&b
Go Link
Google Suche Neu! PageRank 49 blockiert Rechtschreibprüfung Optionen
Web Bilder Groups Verzeichnis News Mehr »
Google
Ληστές
Suche
Erweiterte Suche
Einstellungen
Web-Suche Suche Seiten auf Deutsch
Web
Ergebnisse 1 - 10 von ungefähr 30.000 für Ληστές. (0,35 Seku
Tipp: Anstatt auf "Suche" zu klicken, können Sie auch die Eingabetaste drücken, um Zeit zu sparen.
OTE
Οργανισμός Τηλεπικοινωνιών της Ελλάδος. Προσφέρει υπηρεσίες σταθερής τηλεφωνίας.
www.ote.gr/ - 22k - 30. Nov. 2005 - Im Cache - Ähnliche Seiten
~mperedim/weblog » Ληστές
Ληστές. November 17, 2005 @ 10:37 pm · Filed under eWorld. Ο talos μου θύμισε το περίφημο litigious bastards. ... Ληστές. Φιλικά/Αγωνιστικά Γιάννης ...
mperedim.serverhive.com/weblog/2005/11/17/ληστές/ - 23k - 1. Dez. 2005 - Im Cache - Ähnliche Seiten
Ληστές και Ληστές (aanagno@tin.it) - Ελληνική Λίστα Ανεκδότων
Γεια σας Ληστές (με ήτα γράφεται κι όχι με ιώτα όπως εσφαλμένα το γράφουν ...
Δυο Πόντιοι Ληστές ληστεύουν ένα τραπεζικό αυτοκίνητο και αποκομίζουν δύο ...
anekdota.dyndns.org/jotd6/0584.html - Ähnliche Seiten
Re: Ληστές και Ληστές (magufana@x-treme.gr) - Ελληνική Λίστα Ανεκδότων
>Γεια σας Ληστές (με ήτα γράφεται κι όχι με ιώτα όπως εσφαλμένα το γράφουν >μερικοί κι αν δεν με πιστεύεται συμβουλευτείτε τον ορθογράφο σας) ...
anekdota.dyndns.org/jotd6/0588.html - Ähnliche Seiten
[Weitere Ergebnisse von anekdota.dyndns.org]

OTE - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites

Address http://www.ote.gr/ Go Lir

Google Suche Neu! PageRank 49 blockiert Rechtschreibprüfung Optionen

Η Εταιρεία | Υπηρεσίες Προϊόντα | Πάροχοι | Εταιρική Κοινωνική ευθύνη | Επενδυτικές Σχέσεις | Γραφείο Τύπου

χρυσός οδηγός

OTE share

Sitemap | Επικοινωνία | Αναζήτηση

ΣΤΑΘΕΡΗ ΤΗΛΕΦΩΝΙΑ & ΥΠΗΡΕΣΙΕΣ ΔΕΔΟΜΕΝΩΝ

ΤΕΛΕΥΤΑΙΕΣ ΕΙΔΗΣΕΙΣ

- OTE Q3 05 RESULTS UNDER US GAAP CONFERENCE CALL & WEBCAST DETAILS

COMDEX GREECE 2005 -
dte (DIGITAL TECHNOLOGY EXPO)

Ο Ο.Τ.Ε., συμμετείχε, ως χορηγός, από κοινού με την OTENET, στην Έκθεση Ψηφιακής Τεχνολογίας dte (Digital Technology Expo) - COMDEX Greece η οποία πραγματοποιήθηκε στο Εκθεσιακό Κέντρο Expo Athens.

Σημαντική παρουσία είχε και το Μουσείο Τηλ/νιών του ΟΤΕ, με εκθέματα σχετικά με την εξέλιξη των τηλεπικοινωνιών από την αρχαιότητα μέχρι σήμερα.

Παράλληλα, στην dte-Comdex Greece 2005 έλαβαν χώρα σημαντικές συνεδριακές εκδηλώσεις όπως, τα Συνέδρια "VoIP-Wireless & Mobility", "Resellers Channel"και "CEO Future Watch", στα οποία κατέθεσαν τις απόψεις τους εξέχουσες προσωπικότητες του κλάδου. (Ανθούσα Αττικής 18-20 Νοεμβρίου 2005).

www.comdexgreece.gr >>

cosmoTE | OTENET | INFOTE | cosmoONE | OTE plus

Web Spam Generation

Content spam:

- **repeat words (boost tf)**
- **weave words/phrases into copied text**
- **manipulate anchor texts**

Example:
Remember not only online learning to say the right doctoral degree thing in the right place, but far cheap tuition more difficult still, to leave career unsaid the wrong thing at university the tempting moment.

Link spam:

- **copy links from Web dir. and distort**
- **create honeypot page and sneak in links**
- **infiltrate Web directory**
- **purchase expired domains**
- **generate posts to Blogs, message boards, etc.**
- **build & run spam farm (collusion) + form alliances**

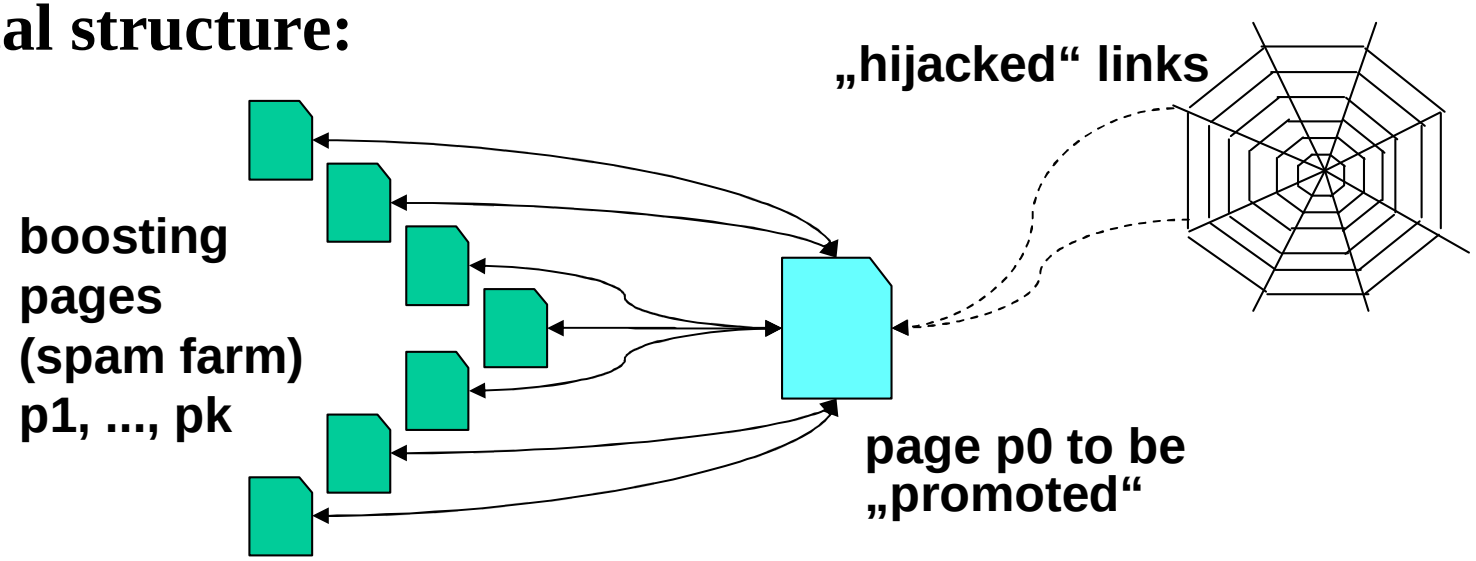
Hide/cloak the manipulation:

- **masquerade href anchors**
- **use tiny anchor images with background color**
- **generate different dynamic pages to browsers and crawlers**

Example:
read about my trip to Las Vegas .

Spam Farms and their Effect

Typical structure:



Web transfers to p0 the „hijacked“ score mass („leakage“)
$\lambda = \Sigma_{q \in IN(p0)-\{p1..pk\}}$ PR(q)/outdegree(q)

Theorem:

p0 obtains the following PR authority:

$$PR(p0) = \frac{1}{1-(1-\varepsilon)^2}\left((1-\varepsilon)\lambda + \frac{\varepsilon((1-\varepsilon)k+1)}{n}\right)$$

The above spam farm is optimal within some family of spam farms (e.g. letting hijacked links point to boosting pages).

Spam Countermeasures

- compute negative propagation of blacklisted pages (BadRank)
- compute positive propagation of trusted pages (TrustRank)
- detect spam pages based on statistical anomalies
- inspect PR distribution in graph neighborhood (SpamRank)
- learn spam vs. ham based on page and page-context features
- spam mass estimation (fraction of PR that is undeserved)
- probabilistic models for link-based authority

  (overcome the discontinuity from 0 outlinks to 1 outlink)

BadRank and TrustRank

BadRank:

start with explicit set B of blacklisted pages
define random-jump vector r by setting r_i=1/|B| if i∈ B and 0 else
propagate BadRank mass to predecessors

$$BR(p) = \beta\, r_p + (1-\beta)\sum_{q \in OUT(p)} BR(q) / \text{indegree}(q)$$

TrustRank:

start with explicit set T of trusted pages with trust values t_i
define random-jump vector r by setting $r_i = t_i$ / if i ∈ B and 0 else
propagate TrustRank mass to successors

$$TR(q) = \tau\, r_q + (1-\tau)\sum_{p \in IN(p)} TR(p) / \text{outdegree}(p)$$

Problems:

maintenance of explicit lists is difficult
difficult to understand (& guarantee) effects

Spam, Damn Spam, and Statistics

Spam detection based on **statistical deviation**:

- **content spam:**
  compare the word frequency distribution
  to the general distribution in „good sites“
- **link spam:**
  find outliers in outdegree and indegree distributions
  and inspect intersection

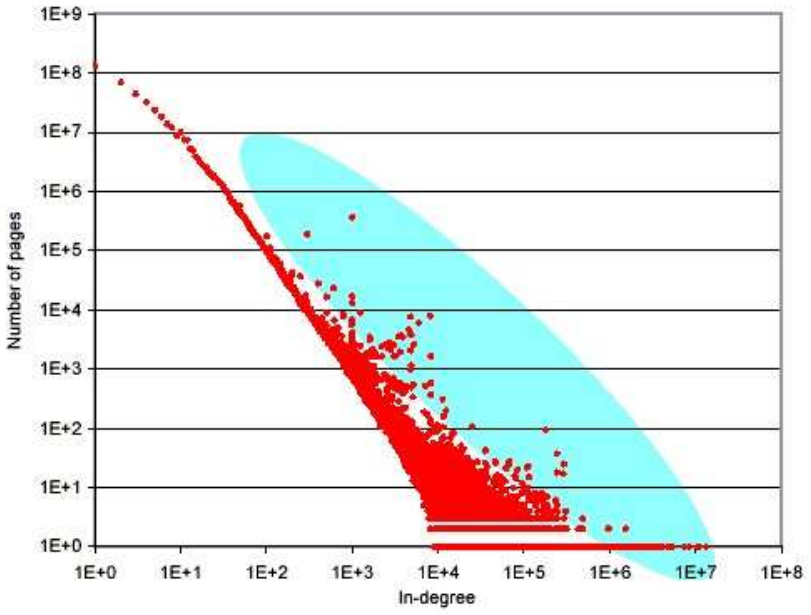


Figure 5: Distribution of in-degrees

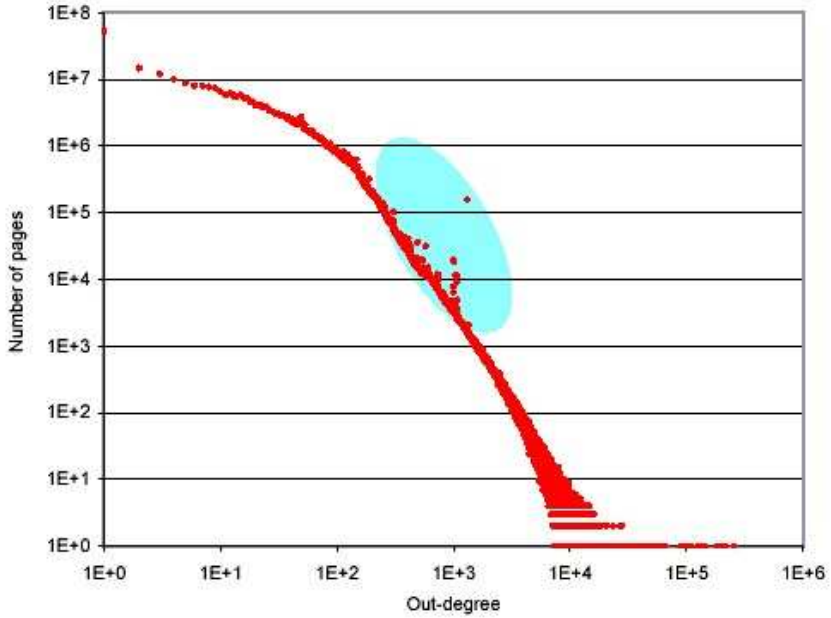


Figure 4: Distribution of out-degrees

Source: D. Fetterly, M. Manasse, M. Najork: WebDB 2004

SpamRank [Benczur et al. 2005]

Key idea:

Inspect PR distribution among a suspected page's neighborhood in a power-law graph

$\rightarrow$ should also be power-law distributed, and deviation is suspicious (e.g. pages that receive their PR from many low-PR pages)

3-phase computation:

1) **for each page q and supporter p compute approximate PPR(q) with random-jump vector r_p=1 and 0 otherwise**
   $PPR_p(q)$ is interpreted as support of p for q
2) **for each page p compute a penalty based on PPR vectors**
3) **define one PPR vector with penalties as random-jump prob's and compute SpamRank as „personalized" BadRank**

true authority = PageRank - SpamRank

SpamRank Details (1)

Phase 1:

$PPR_p(q)$ with singleton random-jump vector equals prob. that a random tour starting at p visits q:

$$PPR_p(q) = \sum_{\substack{tours\ t: \\ p \to q}} P[t]\varepsilon(1-\varepsilon)^{length(t)}$$

(geometric distr. tour length)

$$P[t : w_1 w_2 ... w_k \ \ of\ length\ k-1] = \prod_{i=1}^{k-1} 1/\,\text{outdegree}(w_i)$$

approximate $PPR_p(q)$ vectors by Monte Carlo simulation:

- **generate tours of length 1, 2, etc.**
- **generate random tour starting at p,**
- **count as „success“ with geometr. weight if q is end point**

ratio of „success“ to „trials“ is unbiased estimator of $PPR_p(q)$

SpamRank Details (2)

Phase 2:

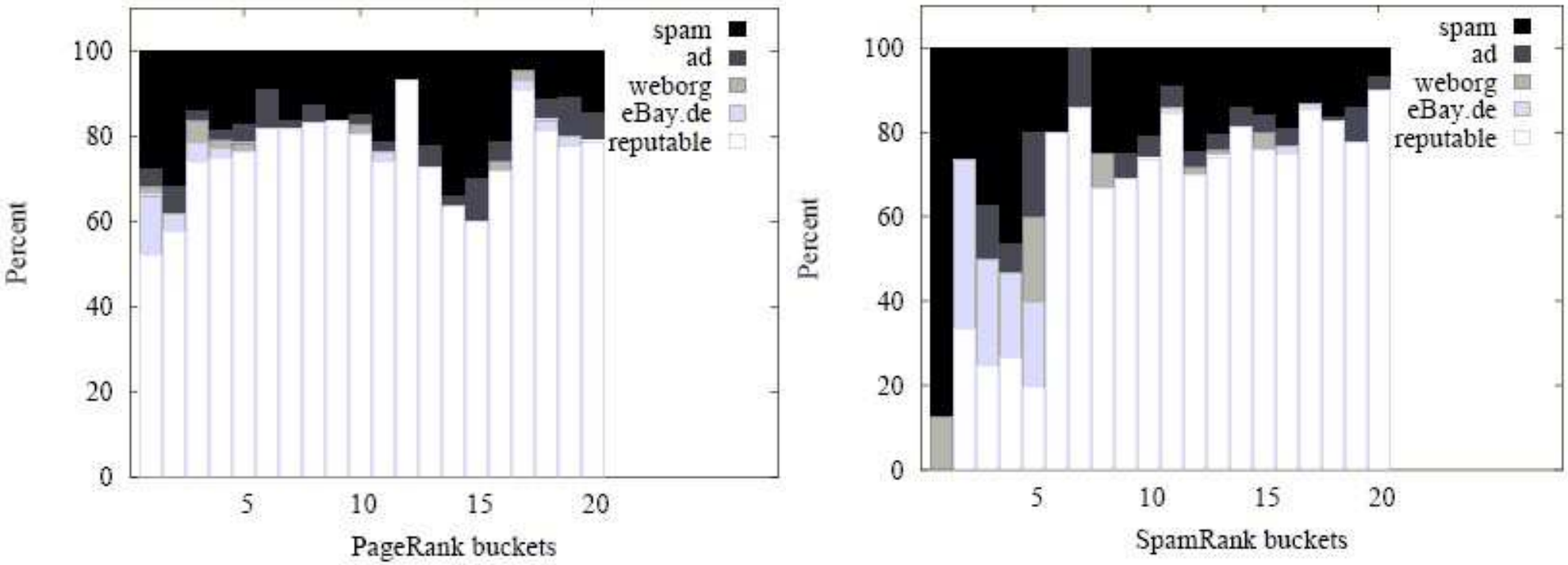
a) compute „abnormality“ for page q:

support of p for q := $PPR_p(q)$ for all pages p
compute histogram of support(p)*PR(p) values
compare to histogram of Pareto-distributed values
$\rho(q)$:= correlation coefficient between histograms

b) compute „penalty“ for pages p:

penalty(p) := 0;
for each page q do
if $\rho(q) < \rho_0(q)$ then
penalty(p) := penalty(p) + $(\rho_0(q) - \rho(q))$*support(p,q)
with „normally expected“ correlation $\rho_0(q)$ close to 1

SpamRank Experimental Results



Distribution of PageRank and SpamRank Mass over Web-Page Categories (1000 pages sample)

Source: Benczur et al., AIRWeb Workshop 2005

SpamMass Score [Gyöngyi et al. 2005]

PR contribution of page p to page q:

$$PC_p(q) = P[\,jump\ to\ \ p\,] \cdot \sum_{\substack{tours\ t:\\ p \to q}} P[t]\varepsilon(1-\varepsilon)^{length(t)}$$

compute by PPR with jump to p only

PR of page q: $PR(q) = \sum_{all\ \ pages\ p} PC_p(q)$

Approach:

assume Web W is partitioned into good pages W^+ and bad pages W^-

assume that „good core“ $V^+ \subset W^+$ is known

estimate SpamMass of page q: $SM(q) = PR(q) - \sum_{p \in V^+} PC_p(q)$

and relative SpamMass of q: $rSM(q) = SM(q) / PR(q)$

Learning Spam Features [Drost/Scheffer 2005]

Use classifier (e.g. Bayesian predictor, SVM) to predict „spam vs. ham“ based on page and page-context features

Most discriminative features are:

tfidf weights of words in p0 and IN(p0)
avg, #inlinks of pages in IN(p0)
avg. #words in title of pages in OUT(p0)
#pages in IN(p0) that have same length as some other page in IN(p0)
avg. # inlinks and outlinks of pages in IN(p0)
avg. #outlinks of pages in IN(p0)
avg. #words in title of p0
total #outlinks of pages in OUT(p0)
total #inlinks of pages in IN(p0)
clustering coefficient of pages in IN(p0) (#linked pairs / m(m-1) possible pairs)
total #words in titles of pages in OUT(p0)
total #outlinks of pages in OUT(p0)
avg. #characters of URLs in IN(p0)
#pages in IN(p0) and OUT(p0) with same MD5 hash signature as p0
#characters in domain name of p0
#pages in IN(p0) with same IP number as p0

But spammers may learn to adjust to the anti-spam measures. It‘s an arms race!

Ranking based on Historical Data

Google has filed US patent 20050071741

(see http://appft1.uspto.gov) for **ranking *criteria***:

- page inception date (e.g. domain registration or first crawl)
- change frequency and amount of page content change
- appearance and disappearance of links to a page
- change frequency of anchor texts
- freshness and churn of links and trust in links
- click-through rate of query-result pages (incl. purchased results)
- shift in keyword interpretation (e.g. 9-11)
- user behavior (e.g. via toolbar)

etc. etc.

many speculations about use of these criteria,
likely to be considered to combat spam (and for personalization?)

Link Analysis Summary

- **PageRank, HITS, etc. (LAR methods) are major achievements for better Web search**
- **LAR build on well-founded theory, but full understanding of sensitivity and special properties still missing**
- **Significance of LAR variants and details are debated**
- **Personalized Authority Scoring seems viable and promising**
- **Extensions for Web dynamics and other aspects are promising**
- **Link spam perceived as mega problem and addressed by statistical methods (but may need deeper adversary theory)**
- **LAR needs to be generalized to social networks and other reference graphs**

Additional Literature for Chapter 5

Link Analysis Principles & Algorithms:

- Chakrabarti, Chapter 7
- S. Chakrabarti: Using Graphs in Unstructured and Semistructured Data Mining, Tutorial Slides, ADFOCS Summer School 2004
- J.M. Kleinberg: Authoritative Sources in a Hyperlinked Environment, JACM 46(5), 1999
- S Brin, L. Page: The Anatomy of a Large-Scale Hypertextual Web Search Engine, WWW 1998
- K. Bharat, M. Henzinger: Improved Algorithms for Topic Distillation in a Hyperlinked Environment, SIGIR 1998
- J. Dean, M. Henzinger: Finding Related Pages in the WorldWideWeb, WWW 1999
- R. Lempel, S. Moran: SALSA: The Stochastic Approach for Link-Structure Analysis, ACM TOIS 19(2), 2001.
- A. Borodin, G.O. Roberts, J.S. Rosenthal, P. Tsaparas: Finding Authorities and Hubs from Link Structures on the World Wide Web, WWW 2001
- C. Ding, X. He, P. Husbands, H. Zha, H. Simon: PageRank, HITS, and a Unified Framework for Link Analysis, SIAM Int. Conf. on Data Mining, 2003.
- A. Borodin, G.O. Roberts, J.S. Rosenthal, P. Tsaparas: Link analysis ranking: algorithms, theory, and experiments. ACM TOIT 5(1), 2005
- M. Bianchini, M. Gori, F. Scarselli: Inside PageRank.ACM TOIT 5(1), 2005
- A.N. Langville, C.D. Meyer: Deeper inside PageRank. Internet Math., 1(3), 2004

Additional Literature for Chapter 5

Personalized Authority Scoring:

- Taher Haveliwala: Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search, IEEE Trans. on Knowledge and Data Engineering, 2003.
- S.D. Kamvar, T.H. Haveliwala, C.D. Manning, G.H. Golub: Extrapolation Methods for Accelerating PageRank Computations, WWW 2003
- S.D. Kamvar, T.H. Haveliwala, C.D. Manning, G.H. Golub: Exploiting the Block Structure of the Web for Computing PageRank, Stanford Technical Report, 2003
- G. Jeh, J. Widom: Scaling personalized web search, WWW 2003.
- J. Luxenburger, G. Weikum: Query-Log Based Authority Analysis for Web Information Search, WISE 2004

Online Page Importance and Web Dynamics:

- S. Abiteboul, M. Preda, G. Cobena: Adaptive on-line page importance computation, WWW 2003.
- K. Berberich, M. Vazirgiannis, G. Weikum: Time-aware Authority Ranking, Internet Mathematics 2005

Additional Literature for Chapter 5

Spam-Resilient Authority Scoring:

- Z. Gyöngyi, H. Garcia-Molina: Spam: It‘s Not Just for Inboxes Anymore, IEEE Computer 2005
- Z. Gyöngyi, H. Garcia-Molina: Link Spam Alliances, VLDB 2005
- Z. Gyöngyi, P. Berkhin, H. Garcia-Molina, J. Pedersen: Link Spam Detection based on Mass Estimation, Technical Report, Stanford, 2005
- Z. Gyöngyi, H. Garcia-Molina: Combating Web Spam with TrustRank, VLDB 2004
- D. Fetterly, M. Manasse, M. Najork: Spam, Damn Spam, and Statistics, WebDB 2005
- I. Drost, T. Scheffer: Thwarting the Nigritude Ultramarine: Learning to Identify Link Spam, ECML 2005
- A.A. Benczur, K. Csalongany, T. Sarlos, M. Uher: SpamRank – Fully Automatic Link Spam Detection, AIRWeb Workshop, 2005
- R. Guha, R. Kumar, P. Raghavan, A. Tomkins: Propagation of Trust and Distrust, WWW 2004