Centrality and prestige

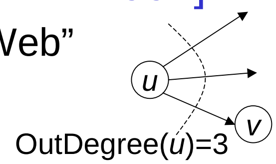
How important is a node?

- Degree, min-max radius, …
- Pagerank
- Maximum entropy network flows
- HITS and stochastic variants
- Stability and susceptibility to spamming
- Hypergraphs and nonlinear systems
- Using other hypertext properties
- Applications: Ranking, crawling, clustering, detecting obsolete pages

Prestige as Pagerank [BrinP1997]

- "Maxwell's equation for the Web"

$$PR(v) = \sum_{(u,v)\in E} \frac{PR(u)}{\text{OutDegree}(u)}$$



- *PR* converges only if *E* is aperiodic and irreducible; make it so:

$$PR(v) = \frac{d}{N} + (1-d) \sum_{(u,v)\in E} \frac{PR(u)}{\text{OutDegree}(u)}$$

- *d* is the (tuned) probability of "teleporting" to one of *N* pages uniformly at random
- (Possibly) unintended consequences: topic sensitivity, stability

Prestige as network flow

- y_{ij} =#surfers clicking from *i* to *j* per unit time
- Hits per unit time on page *j* is $H_j = \sum_{(i,j)\in E} y_{ij}$
- Flow is conserved at $\forall j : \sum_{(i,j)\in E} y_{ij} = \sum_{(j,k)\in E} y_{jk}$
- The total traffic is $Y = \sum_j H_j = \sum_{i,j} y_{ij}$
- Normalize: $p_{ij} = y_{ij} / Y$
  Can interpret p_{ij} as a probability
- Standard Pagerank corresponds to one solution: $p_{ij} = H_i / (Y \,\text{OutDegree}(i))$
- Many other solutions possible

Maximum entropy flow [Tomlin2003]

- Flow conservation modeled using feature

$$\forall r = 1,\ldots,N:\ f_r(x_{ij}) = \begin{cases} +1 & j = r, (i,r) \in E \\ -1 & i = r, (r,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

- And the constraints $0 = E(f_r(x_{ij})) = \sum_{i,j} p_{ij} f_r(x_{ij})$
- Goal is to maximize $-\sum_{i,j} p_{ij} \log p_{ij}$ subject to $\sum_{i,j} p_{ij} = 1$
- Solution has form $p_{ij} = \exp(\lambda_0 - \lambda_i + \lambda_j)$
- λ_i is the "hotness" of page *i*

Maxent flow results

Test	PageRank	TrafficRank	HOTness
1	0.6443	2.275	0.4610
2	1.242	1.417	1.160

λ_i ranking is better than Pagerank; H_i ranking is worse

Two IBM intranet data sets with known top URLs

Average rank (10^6) of known top URLs when sorted by Pagerank

Depth up to which dmoz.org URLs are used as ground truth

H_i

λ_i

Level	Number	PageRank	TrafficRank	HOTness
1	27	0.753	6.404	1.656
2	4258	3.143	2.862	2.614
3	65343	4.448	4.385	3.949
4	228943	4.686	4.887	4.286
5	427578	4.817	5.127	4.438
∞	990354	5.236	5.677	4.812

(Smaller rank is better)

Average rank (10^8)

HITS [Kleinberg1997]

- Two kinds of prestige
  - Good hubs link to good authorities
  - Good authorities are linked to by good hubs

$$a(v) = \sum_{(u,v)\in E} h(u); \quad h(u) = \sum_{(u,v)\in E} a(v)$$

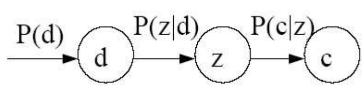
- Eigensystems of EE^{T} (h) and $E^{\mathrm{T}}E$ (a)
- Whereas Pagerank uses the eigensystem of $dJ + (1-d)L^T$ where

$$J = \begin{bmatrix} 1/N & \cdots & 1/N \\ \vdots & \ddots & \vdots \\ 1/N & \cdots & 1/N \end{bmatrix} \text{ and } L(i,j) = \begin{cases} 1/\mathrm{OutDegree}(i) & (i,j)\in E \\ 0 & \text{otherwise} \end{cases}$$

- Query-specific graph; drop same-site links

Dyadic interpretation [CohnC2000]

- Graph includes many communities z
  - Query="Jaguar" gets auto, game, animal links
- Each URL is represented as two things
  - A document d
  - A citation c



- Max $\sum_{(d,c)\in E} \Pr(d,c)$

$$\begin{aligned} &= \sum_{(d,c)\in E} \Pr(d)\Pr(c \mid d) \\ &\approx \sum_{(d,c)\in E} \Pr(d) \sum_z \Pr(z \mid d)\Pr(c \mid z) \end{aligned}$$

- Guess number of aspects zs and use [Hofmann 1999] to estimate Pr(c|z)
- These are the most authoritative URLs

Dyadic results for "Machine learning"

Top citations by $P(c\|z)$, computed by PHITS algorithm:	
factor 1	(Reinforcement Learning)
0.0108	Learning to predict by the methods of temporal differences. Sutton
0.0066	Neuronlike adaptive elements that can solve difficult learning control problems. Barto et al
0.0065	Practical Issues in Temporal Difference Learning. Tesauro.
factor 2	(Rule Learning)
0.0038	Explanation-based generalization: a unifying view. Mitchell et al
0.0037	Learning internal representations by error propagation. Rumelhart et al
0.0036	Explanation-Based Learning: An Alternative View. DeJong et al
factor 3	(Neural Networks)
0.0120	Learning internal representations by error propagation. Rumelhart et al
0.0061	Neural networks and the bias-variance dilemma. Geman et al
0.0049	The Cascade-Correlation learning architecture. Fahlman et al
factor 4	(Theory)
0.0093	Classification and Regression Trees. Breiman et al
0.0066	Learnability and the Vapnik-Chervonenkis dimension, Blumer et al
0.0055	Learning Quickly when Irrelevant Attributes Abound. Littlestone
factor 5	(Probabilistic Reasoning)
0.0118	Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Pearl.
0.0094	Maximum likelihood from incomplete data via the em algorithm. Dempster et al
0.0056	Local computations with probabilities on graphical structures... Lauritzen et al
factor 6	(Genetic Algorithms)
0.0157	Genetic Algorithms in Search, Optimization, and Machine Learning. Goldberg
0.0132	Adaptation in Natural and Artificial Systems. Holland
0.0096	Genetic Programming: On the Programming of Computers by Means of Natural Selection. Koza

Clustering based on citations + ranking within clusters

Spamming link-based ranking

- Recipe for spamming HITS
  - Create a hub linking to genuine authorities
  - Then mix in links to your customers' sites
  - Highly susceptible to adversarial behavior
- Recipe for spamming Pagerank
  - Buy a bunch of domains, cloak IP addresses
  - Host a site at each domain
  - Sprinkle a few links at random per page to other sites you own
  - Takes more work than spamming HITS

Stability of link analysis [NgZJ2001]

- Compute HITS authority scores and Pagerank
- Delete 30% of nodes/links at random
- Recompute and compare ranks; repeat
- Pagerank ranks more stable than HITS authority ranks
  - Why?
  - How to design more stable algorithms?

HITS Authority

1	3	1	1	1
2	5	3	3	2
3	12	6	6	3
4	52	20	23	4
5	171	119	99	5
6	135	56	40	8
10	179	159	100	7
8	316	141	170	6

Pagerank

1	1	1	1	1
2	2	2	2	2
3	5	6	4	5
4	3	5	5	4
5	6	3	6	3
6	4	4	3	6
7	7	7	7	7
8	8	8	8	9

Stability depends on graph and params

- Auth score is eigenvector for $E^T E = S$, say
- Let $\lambda_1 > \lambda_2$ be the first two eigenvalues
- There exists an S' such that
  - S and S' are close $\|S-S'\|_F = O(\lambda_1 - \lambda_2)$
  - But $\|u_1 - u'_1\|_2 = \Omega(1)$
- Pagerank p is eigenvector of $(\varepsilon U + (1-\varepsilon)E)^T$
  - U is a matrix full of $1/N$ and ε is the jump prob
  - If set C of nodes are changed in any way, the new Pagerank vector p' satisfies

$$\|p' - p\|_2 \le \left(2\sum_{u\in C} p_u\right)/\varepsilon$$

Randomized HITS

- Each half-step, with probability ε, teleport to a node chosen uniformly at random

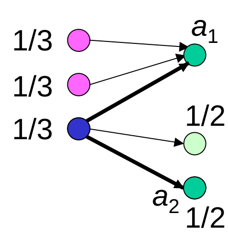
$$a^{(t+1)} = \varepsilon\vec{1} + (1-\varepsilon)E_{\text{row}}^{T}h^{(t)}$$
$$h^{(t+1)} = \varepsilon\vec{1} + (1-\varepsilon)E_{\text{col}}a^{(t)}$$

- Much more stable than HITS
- Results meaningful too
  - ε near 1 will always stabilize
  - Here ε was 0.2

Randomized HITS	1	3	3	2	1
	4	1	1	1	2
	2	2	2	3	4
	3	4	4	4	3
	5	6	6	6	5
	6	5	5	5	6
	7	7	7	7	7
	8	8	8	8	8
Pagerank	1	1	1	1	2
	3	2	2	2	1
	2	3	3	3	3
	4	4	4	4	4
	5	6	7	5	5
	6	7	6	6	6
	7	5	5	7	7
	8	9	9	9	11

Another random walk variation of HITS

- SALSA: Stochastic HITS [Lempel+2000]
- Two separate random walks
  - From authority to authority via hub
  - From hub to hub via authority



- Transition probability Pr($a_i \rightarrow a_j$) =

$$\sum_{h:(h,a_i),(h,a_j)\in E} \frac{1}{\text{InDegree}(a_i)} \frac{1}{\text{OutDegree}(h)}$$

- If transition graph is irreducible, $\pi_a \propto \text{InDegree}(a)$
- For disconnected components, depends on relative size of bipartite cores
- Avoids dominance of larger cores

SALSA sample result (“movies”)

url	title	cat	weight
http://go.msn.com/npl/msnt.asp	MSN.COM	(3)	0.1673
http://go.msn.com/bql/whitepages.asp	White Pages - msn.com	(3)	0.1672
http://go.msn.com/bsl/webevents.asp	Web Events	(3)	0.1672
http://go.msn.com/bql/scoreboards.asp	MSN Sports scores	(3)	0.1672

HITS: The Tightly-Knit Community (TKC) effect

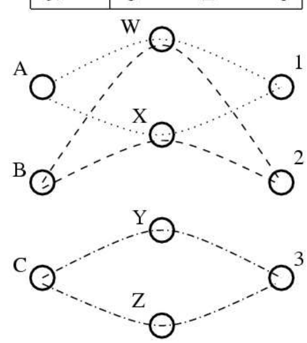
SALSA: Less TKC influence (but no reinforcement!)

url	title	cat	weight
http://us.imdb.com/	The Internet Movie Database	(3)	0.2533
http://www.mrshowbiz.com/	Mr Showbiz	(3)	0.2233
http://www.disney.com/	Disney.com–The Web Site for Families	(3)	0.2200
http://www.hollywood.com/	Hollywood Online:...all about movies	(3)	0.2134
http://www.imdb.com/	The Internet Movie Database	(3)	0.2000
http://www.paramount.com/	Welcome to Paramount Pictures	(3)	0.1967
http://www.mca.com/	Universal Studios	(3)	0.1800
http://www.discovery.com/	Discovery Online	(3)	0.1550
http://www.film.com/	Welcome to Film.com	(3)	0.1533
http://www.mgmua.com/	mgm online	(3)	0.1300

Links in relational data [GibsonKR1998]

- (Attribute, value) pair is a node
  - Each node *v* has weight w_v
- Each tuple is a hyperedge
  - Tuple *r* has weight x_r
- HITS-like iterations to update weight w_v
  - For each tuple $r = (v, u_1, \ldots, u_k)$
    $$x_r = \otimes(w_{u_1}, \ldots, w_{u_k})$$
  - Update weight
    $$w_v \leftarrow \sum_r x_r$$
- Combining operator $\otimes$ can be sum, max, product, L_p avg, etc.

	Attribute		
Tuple	a	b	c
1.	A	W	1
2.	A	X	1
3.	B	W	2
4.	B	X	2
5.	C	Y	3
6.	C	Z	3



Distilling links in relational data

	Author	Author	Forum	Year
Theory	0.1811: Chen	0.1629: Chen	0.317: TCS	0.563: 1995
	0.1185: Chang	0.1289: Wang	0.2264: IPL	0.5596: 1994
	0.1147: Zhang	0.1007: COMPREVS	0.195: LNCS	0.4332: 1996
	0.1119: Agarwal	0.1: Li	0.1633: INFCTRL	0.151: 1993
	0.1104: Bellare	0.09004: Rozenberg	0.1626: DAMATH	0.07487: 1992
	0.1053: Gu	0.08837: Igarashi	0.1464: JPDC	0.0155: 1976
	0.09467: Dolev	0.08171: Sharir	0.1371: SODA	0.005276: 1972
	0.08571: Hemaspaand	0.0797: Huang	0.1266: STOC	0.004569: 1985
	0.08423: Farach	0.07924: Maurer	0.1074: IEEETC	0.001891: 1973
	0.08232: Ehrig	0.0783: Lee	0.1002: JCSS	0.001679: 1970
Database	-0.1195: Stonebrake	-0.1068: Wiederhold	-0.384: IEEEDataEng	-0.2165: 1986
	-0.1059: Agrawal	-0.09615: David	-0.256: VLDB	-0.1983: 1987
	-0.1023: Wiederhold	-0.08343: DeWitt	-0.2392: SIGMOD	-0.1519: 1989
	-0.07735: Abiteboul	-0.07643: Richard	-0.1504: PODS	-0.14: 1988
	-0.06783: Yu	-0.07454: Stonebrak	-0.1397: ACMTDS	-0.11: 1984
	-0.06722: Navathe	-0.07403: Michael	-0.1176: IEEETransa	-0.06571: 1975
	-0.06615: Litwin	-0.07159: Robert	-0.04832: IEEETrans	-0.06431: 1990
	-0.06308: Bernstein	-0.06843: Jagadish	-0.04658: IEEETechn	-0.03765: 1980
	-0.0623: Jajodia	-0.06722: James	-0.04533: WorkshopI	-0.03238: 1974
	-0.0587: Motro	-0.0671: Navathe	-0.03581: IEEEDBEng	-0.02873: 1982

Searching and annotating graph data

Searching graph data

- Nodes in graph contain text
  - Random→Intelligent surfer [RichardsonD2001]
  - Topic-sensitive Pagerank [Haveliwala2002]
  - Assigning image captions using random walks [PanYFD2004]
- Query is a set of keywords
  - All keywords may not match a single node
  - Implicit joins [Hulgeri+2002, Agrawal+2002]
  - Or rank aggregation [Balmin+2004] required

Intelligent Web surfer

$$\Pr_q(j) = (1-\beta)\Pr'_q(j) + \beta \sum_{(i,j)\in E} \Pr_q(i)\Pr_q(i \to j)$$

Keyword

Probability of teleporting to node *j*

Probability of walking from *i* to *j* wrt *q*

Relevance of node *k* wrt *q*

$$\Pr'_q(j) = \frac{R_q(j)}{\sum_{k\in V} R_q(k)} \qquad \Pr'_q(i \to j) = \frac{R_q(j)}{\sum_{(i,k)\in E} R_q(k)}$$

Pick out-link to walk on in proportion to relevance of target out-neighbor

$$\mathrm{PR}_Q(j) = \sum_{q\in Q} \Pr(q)\Pr_q(j)$$

Query=set of words

Pick a query word per some distribution, e.g. IDF

Implementing the intelligent surfer

Table 1: Results on *educrawl*

Query	QD-PR	PR
chinese association	10.75	6.50
computer labs	9.50	13.25
financial aid	8.00	12.38
intramural	16.5	10.25
maternity	12.5	6.75
president office	5.00	11.38
sororities	13.75	7.38
student housing	14.13	10.75
visitor visa	19.25	12.50
Average	**12.15**	**10.13**

Table 2: Results on *WebBase*

Query	QD-PR	PR
alcoholism	11.50	11.88
architecture	8.45	2.93
bicycling	8.45	6.88
rock climbing	8.43	5.75
shakespeare	11.53	5.03
stamp collecting	9.13	10.68
vintage car	13.15	8.68
Thailand tourism	16.90	9.75
Zen Buddhism	8.63	10.38
Average	**10.68**	**7.99**

- $\mathrm{PR}_Q(j)$ approximates a walk that picks a query keyword using $\Pr(q)$ at every step
- Precompute and store $\Pr_q(j)$ for each keyword q in lexicon: space blowup = avg doc length
- Query-dependent PR rated better by volunteers

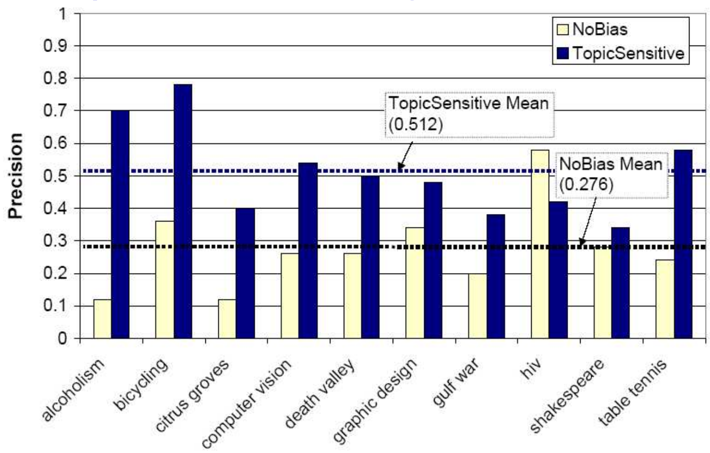
Topic-sensitive Pagerank

- High overhead for per-word Pagerank
- Instead, compute Pageranks for some collection of broad topics $\mathrm{PR}_c(j)$
  - Topic c has sample page set S_c
  - Walk as in Pagerank
  - Jump to a node in S_c uniformly at random
- "Project" query onto set of topics

$$\Pr(c \mid Q) \propto \Pr(c) \prod_{q \in Q} \Pr(q \mid c)$$

- Rank responses by projection-weighted Pageranks $\mathrm{Score}(Q, j) = \sum_c \Pr(c \mid Q)\, \mathrm{PR}_c(j)$

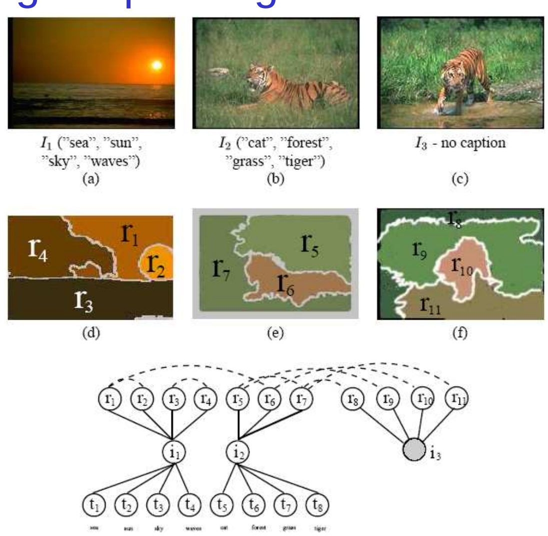
Topic-sensitive Pagerank results



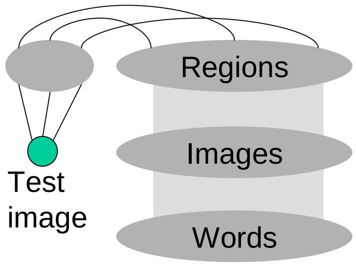
- Users prefer topic-sensitive Pagerank on most queries to global Pagerank + keyword filter

Image captioning

- Segment images into regions
- Image has caption words
- Three-layer graph: image, regions, caption words
- Threshold on region similarity to connect regions (dotted)



Random walks with restarts



Query			
Truth	cat, grass, tiger, water	mane, cat, lion, grass	sun, water, tree, sky
GCap	grass, cat, tiger, water	lion, grass, cat, mane	tree, water, buildings, sky
	(a)	(b)	(c)

- Find regions in test image
- Connect regions to other nodes in the region layer using region similarity
- Random walk, restarting at test image node
- Pick words with largest visit probability

Proximity search: two paradigms

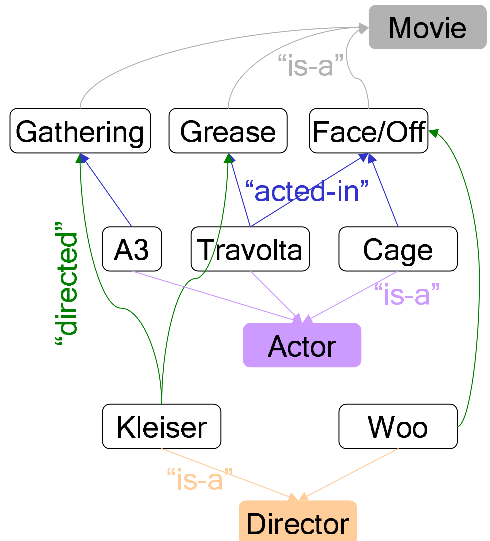
- A single node as query response
  - Find node that matches query terms…
  - …or is “near” nodes matching query terms

  [Goldman+ 1998]
- A connected subgraph as query response
  - Single node may not match all keywords
  - No natural “page boundary”

  [Bhalotia+2002, Agrawal+2002]

Single-node response examples

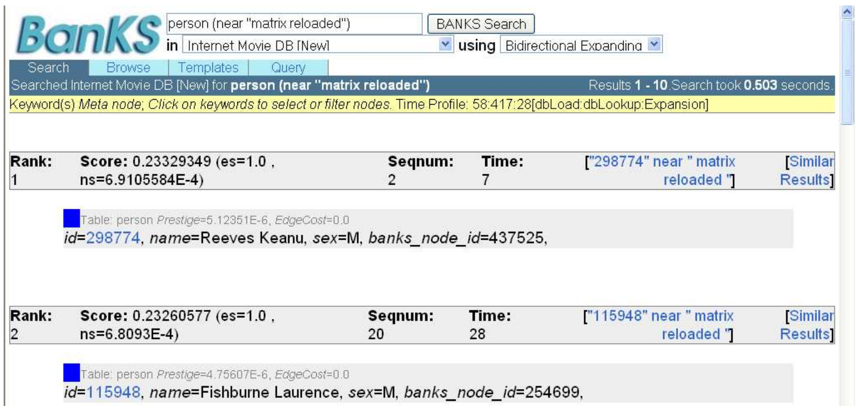
- Travolta, Cage
  - Actor, Face/Off
- Travolta, Cage, Movie
  - Face/Off
- Kleiser, Movie
  - Gathering, Grease
- Kleiser, Woo, Actor
  - Travolta



Basic search strategy

- Node subset A activated because they match query keyword(s)
- Look for node near nodes that are activated
- Goodness of response node depends
  - Directly on degree of activation
  - Inversely on distance from activated node(s)

Proximity query: screenshot



http://www.cse.iitb.ac.in/banks/

Ranking a single node response

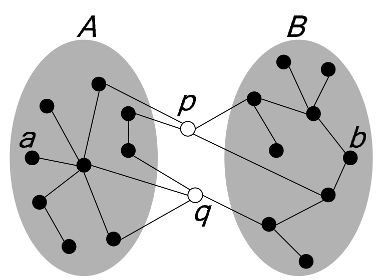
- Activated node set *A*
- Rank node *r* in "response set" *R* based on proximity to nodes *a* in *A*
  - Nodes have relevance ρ_R and ρ_A in [0,1]
  - Edge costs are "specified by the system"
- *d*(*a*,*r*) = cost of shortest path from *a* to *r*
- Bond between *a* and *r* $$b(a,r) = \frac{\rho_A(a)\rho_R(r)}{d(a,r)^t}$$
- Parameter *t* tunes relative emphasis on distance and relevance score
- Several ad-hoc choices

Scoring single response nodes

- Additive $$\text{score}(r) = \sum_{a \in A} b(a,r)$$
- Belief $$\text{score}(r) = 1 - \prod_{a \in A} \left(1 - b(a,r)\right)$$
- Goal: list a limited number of find nodes with the largest scores
- Performance issues
  - Assume the graph is in memory?
  - Precompute all-pairs shortest path ($|V|^3$)?
  - Prune unpromising candidates?

Hub indexing

- Decompose APSP problem using sparse vertex cuts
  - $|A|+|B|$ shortest paths to p
  - $|A|+|B|$ shortest paths to q
  - $d(p,q)$
- To find $d(a,b)$ compare
  - $d(a \rightarrow p \rightarrow b)$ not through q
  - $d(a \rightarrow q \rightarrow b)$ not through p
  - $d(a \rightarrow p \rightarrow q \rightarrow b)$
  - $d(a \rightarrow q \rightarrow p \rightarrow b)$



- Greatest savings when |A|≈|B|
- Heuristics to find cuts, e.g. large-degree nodes

ObjectRank [Balmin+2004]

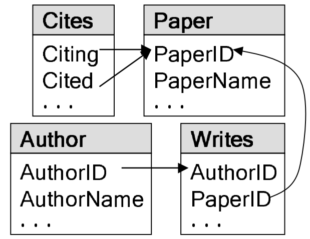
- Given a data graph with nodes having text
- For each keyword precompute a keyword-sensitive Pagerank [RichardsonD2001]
- Score of a node for multiple keyword search based on fuzzy AND/OR
  - Approximation to Pagerank of node with restarts to nodes matching keywords
- Use Fagin-merge [Fagin2002] to get best nodes in data graph

Connected subgraph as response

- Single node may not match all keywords
  - No natural “page boundary”
  - On-the-fly joins make up a “response page”
- Two scenarios
  - Keyword search on relational data
    - Keywords spread among normalized relations
  - Keyword search on XML-like or Web data
    - Keywords spread among DOM nodes and subtrees

Keyword search on relational data

- Tuple = node
- Some columns have text
- Foreign key constraints = edges in schema graph→
- Query = set of terms
- No natural notion of a document
  - Normalization
  - Join may be needed to generate results
  - Cycles may exist in schema graph: 'Cites'



AuthorID	PaperID
A1	P1
A2	P2
A3	P2

AuthorID	AuthorName
A1	Chaudhuri
A2	Sudarshan
A3	Hulgeri

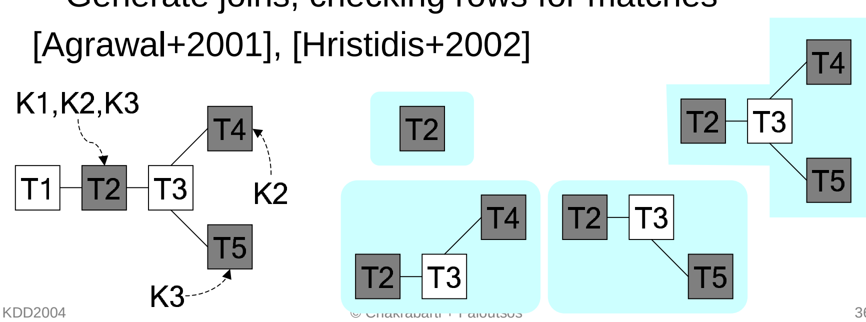
Citing	Cited
P2	P1

PaperID	PaperName
P1	DBXplorer
P2	BANKS

DBXplorer and DISCOVER

- Enumerate subsets of relations in schema graph which, when joined, may contain rows which have *all* keywords in the query
  - "Join trees" derived from schema graph
- Output SQL query for each join tree
- Generate joins, checking rows for matches

[Agrawal+2001], [Hristidis+2002]



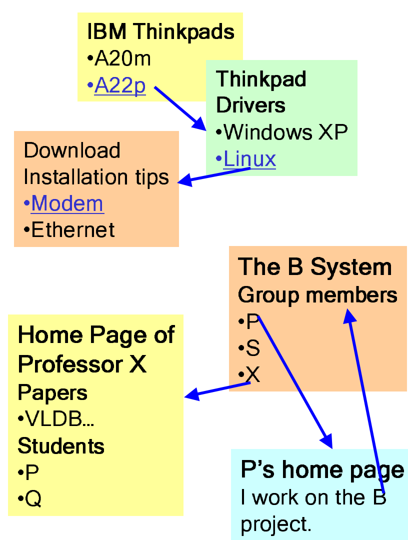
Discussion

- Exploits relational schema information to contain search
- Pushes final extraction of joined tuples into RDBMS
- Faster than dealing with full data graph directly

- Coarse-grained ranking based on schema tree
- Does not model proximity or (dis) similarity of individual tuples
- No recipe for data with less regular (e.g. XML) or ill-defined schema

Motivation from Web search

- "Linux modem driver for a Thinkpad A22p"
  - Hyperlink path matches query collectively
  - Conjunction query would fail
- Projects where X and P work together
  - Conjunction may retrieve wrong page
- General notion of graph proximity



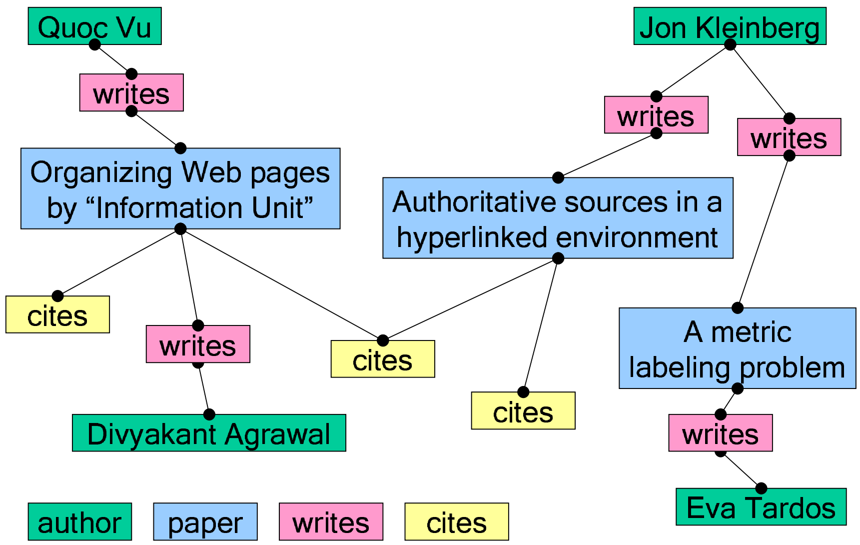
Data structures for search

- Answer = tree with at least one leaf containing each keyword in query
  - Group Steiner tree problem, NP-hard
- Query term t found in source nodes S_t
- Single-source-shortest-path SSSP iterator
  - Initialize with a source (near-) node
  - Consider edges backwards
  - getNext() returns next nearest node
- For each iterator, each visited node v maintains for each t a set $v.R_t$ of nodes in S_t which have reached v

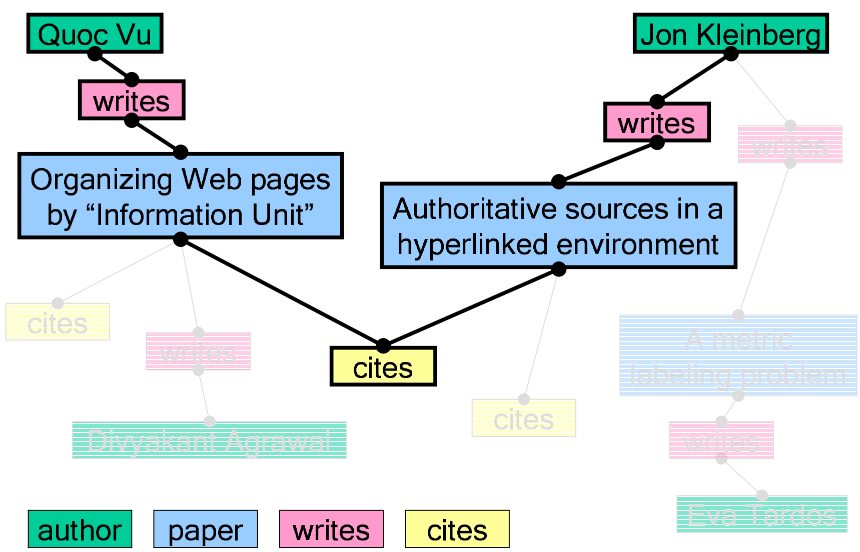
Generic expanding search

- Near node sets S_t with $S = \cup_t S_t$
- For all source nodes $\sigma \in S$
  - create a SSSP iterator with source σ
- While more results required
  - Get next iterator and its next-nearest node v
  - Let t be the term for the iterator's source s
  - crossProduct = $\{s\} \times \Pi_{t' \neq t} v.R_{t'}$
  - For each tuple of nodes in crossProduct
    - Create an answer tree rooted at v with paths to each source node in the tuple
  - Add s to $v.R_t$

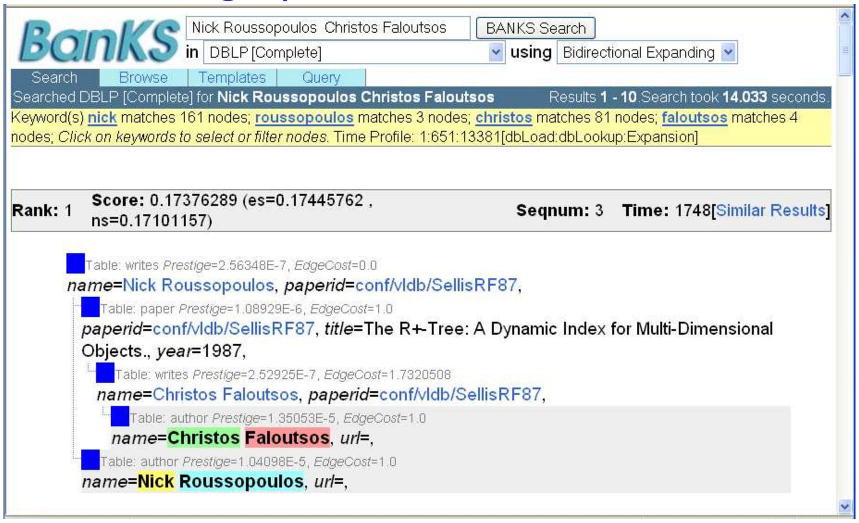
Search example (“Vu Kleinberg”)



First response



Subgraph search: screenshot

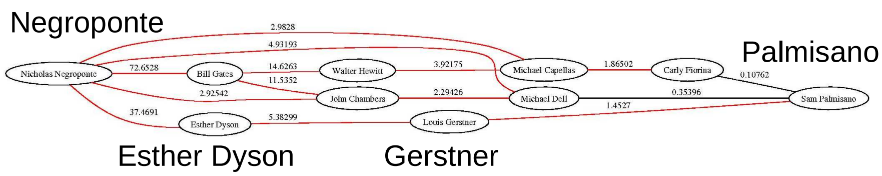


http://www.cse.iitb.ac.in/banks/

Similarity, neighborhood, influence

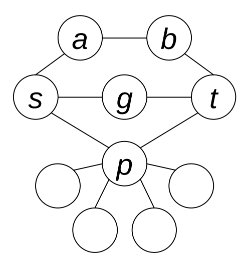
Why are two nodes similar?

- What is/are the best paths connecting two nodes explaining why/how they are related?
  - Graph of co-starring, citation, telephone call, …
- Graph with nodes *s* and *t*; budget of *b* nodes
- Find "best" *b* nodes capturing relationship between *s* and *t* [FaloutsosMT2004]:
  - Proposing a definition of goodness
  - How to efficiently select best connections



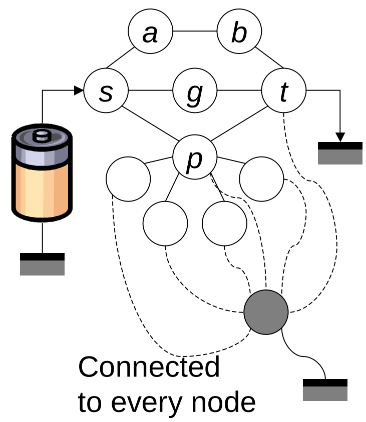
Simple proposals that do not work

- Shortest path
  - Pizza boy *p* gets same attention as *g*
- Network flow
  - $s \rightarrow a \rightarrow b \rightarrow t$ is as good as $s \rightarrow g \rightarrow t$
- Voltage
  - Connect +1V at *s*, ground *t*
  - Both *g* and *p* will be at +0.5V
- Observations
  - Must reward parallel paths
  - Must reward short paths
  - Must penalize/tax pizza boys



Resistive network with universal sink

- Connect +1V to *s*
- Ground *t*
- Introduce universal sink
  - Grounded
  - Connected to every node
- Universal sink is a "tax collector"
  - Penalizes pizza boys
  - Penalizes long paths
- Goodness of a path is the electric current it carries



Resistive network algorithm

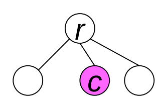
- Ohm's law: $I(u,v) = C(u,v)[V(u) - V(v)] \ \forall u,v$
- Kirchhoff's current law: $\forall v \neq s,t : \sum_u I(u,v) = 0$
- Boundary conditions (without sink): $V(s) = 1, \ V(t) = 0$
- Solution: $$V(u) = \frac{\sum_v V(v)C(u,v)}{\sum_w C(u,w)}, \text{ for } u \neq s,t$$
- Here *C*(*u*,*v*) is the conductance from *u* to *v*
- Add grounded universal sink *z* with *V*(*z*)=0
- Set $\forall u: \ C(u,z) = \alpha \sum_{w \neq z} C(u,w)$
- Display subgraph carrying high current

Distributions coupled via graphs

- Hierarchical classification
  - Document topics organized in a tree
- Mapping between ontologies
  - Can Dmoz label help labeling in Yahoo?
- Hypertext classification
  - Topic of Web page better predicted from hyperlink neighborhood
- Categorical sequences
  - Part-of-speech tagging, named entity tagging
  - Disambiguation and linkage analysis

Hierarchical classification

- Obvious approaches
  - Flatten to leaf topics, losing hierarchy info
  - Level-by-level, compounding error probability
- Cascaded generative model



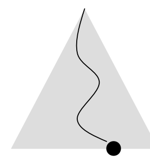
$$\Pr(c \mid d) = \Pr(r \mid d)\Pr(c \mid d, r)$$

  - Pr(c|d,r) estimated as Pr(c|r)Pr(d|c)/Z(r)
  - Estimate of Pr(d|c) makes naïve independence assumptions if d has high dimensionality
  - Pr(c|d,r) tends to 0/1 for large dimensions and
  - Mistake made at shallow levels become irrevocable

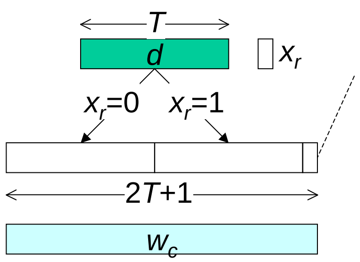
Conditional model on topic tree

- Each node has an associated bit *X*
- Propose a parametric form

$$\Pr(X_c = 1 \mid d, x_r) = \frac{\exp(w_c \cdot F(d, x_r))}{1 + \exp(w_c \cdot F(d, x_r))}$$



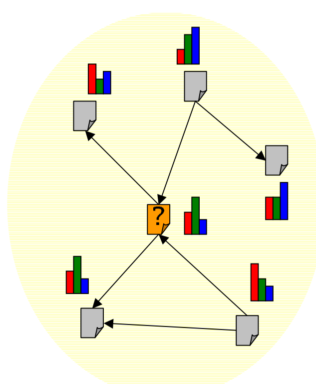
- Each training instance sets one path to 1, all other nodes have *X*=0



	%Accuracy	
	SVM 1v1	Ctree
Reuters 1%	41.9	47.3
Reuters 5%	68	71
News20 1%	17.2	21.8

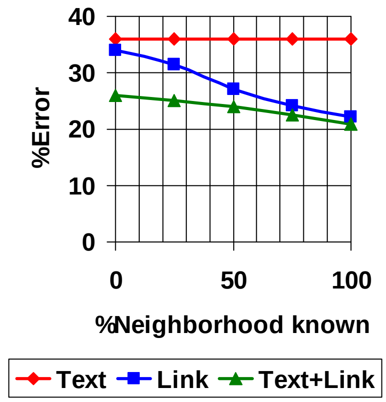
Hypertext classification

- *c*=class, *t*=text, *N*=neighbors
- Text-only model: Pr[*t*|*c*]
- Using neighbors' text to judge my topic: Pr[*t*, *t*(*N*) | *c*]
- Better model: Pr[*t*, *c*(*N*) | *c*]
- Non-linear relaxation



Generative graphical model: results

- 9600 patents from 12 classes marked by USPTO
- Patents have text and cite other patents
- Expand test patent to include neighborhood
- ‘Forget’ fraction of neighbors’ classes



Discriminative graphical model

- OA(*X*) = direct attributes of node *X*
- LD(*X*) = link-derived features of node *X*
  - Mode-link: most frequent label of neighbors(*X*)
  - Count-link: histogram of neighbor labels
  - Binary-link: 0/1 histogram of neighbor labels

$$\Pr(c \mid w_o, \mathrm{OA}(X)) = 1 \big/ \exp(-w_o^T \mathrm{OA}(X) + 1)$$

Neighborhood model params — Local model params

$$\Pr(c \mid w_l, \mathrm{LD}(X)) = 1 \big/ \exp(-w_l^T \mathrm{LD}(X) + 1)$$

$$\hat{C}(X) = \arg\max_c \Pr(c \mid \mathrm{OA}(X)) \Pr(c \mid \mathrm{LD}(X))$$

- Iterate as in generative case

Discriminative model: results [Li+2003]

	Content-Only	Flat-Mode	Flat-Binary	Flat-Count	Mode-Link	Binary-Link	Count-Link
Cora							
Avg. Accuracy	0.674	0.649	0.74	0.728	0.717	0.754	**0.758**
Avg. Precision	0.662	0.704	0.755	0.73	0.717	0.747	**0.759**
Avg. Recall	0.626	0.59	0.689	0.672	0.679	0.716	**0.725**
Avg. F1 Measure	0.643	0.641	0.72	0.7	0.697	0.731	**0.741**
CiteSeer							
Avg. Accuracy	0.607	0.618	0.634	0.644	0.658	0.664	**0.679**
Avg. Precision	0.551	0.55	0.58	0.579	**0.606**	0.597	0.604
Avg. Recall	0.552	0.547	0.572	0.573	0.601	0.597	**0.608**
Avg. F1 Measure	0.551	0.552	0.575	0.575	0.594	0.597	**0.606**
WebKB							
Avg. Accuracy	0.862	0.848	0.832	0.863	0.851	0.871	**0.877**
Avg. Precision	0.876	0.86	0.864	0.876	0.878	**0.879**	0.878
Avg. Recall	0.795	0.79	0.882	0.81	0.772	0.811	**0.83**
Avg. F1 Measure	0.832	0.821	0.836	0.84	0.82	0.847	**0.858**

- Binary-link and count-link outperform content-only at 95% confidence
- Better to separately estimate w_l and w_o
- In+Out+Cocitation better than any subset for LD

Sequential models

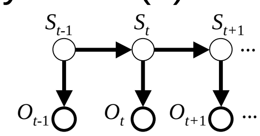
- Text modeled as sequence of tokens drawn from a large but finite vocabulary
- Each token has attributes
  - Visible: allCaps, noCaps, hasXx, allDigits, hasDigit, isAbbrev, (part-of-speech, wnSense)
  - Not visible: part-of-speech, (isPersonName, isOrgName, isLocation, isDateTime)
- Visible (symbols) and invisible (states) attributes of nearby tokens are dependent
- Application decides what is (not) visible
- Goal: Estimate invisible attributes

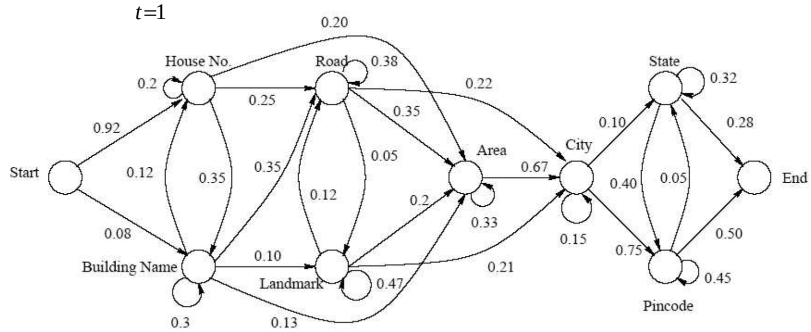
Hidden Markov model

- A generative sequential model for the joint distribution of states (s) and symbols (o)

$$\vec{s} = s_0, s_1, \ldots s_n \qquad \vec{o} = o_1, o_2, \ldots o_n$$

$$\Pr(\vec{s}, \vec{o}) = \prod_{t=1}^{|\vec{o}|} \Pr(s_{t-1} \to s_t) \Pr(s_t \uparrow o_t)$$



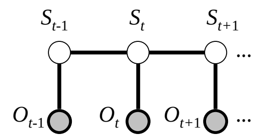


Using redundant token features

- Each *o* is usually a vector of features extracted from a token
- Might have high dependence/redundancy: hasCap, hasXx, isProperNoun
- Parametric model for $\Pr(s_t \uparrow o_t)$ needs to make naïve assumptions to be practical
- Overall joint model $\Pr(\underline{s},\underline{o})$ can be very inaccurate
- (Same argument as in naïve Bayes vs. SVM or maximum entropy text classifiers)

Discriminative graphical model

- Assume one-stage Markov dependence
- Propose direct parametric form for conditional probability of state sequence given symbol sequence



$$\Pr(\vec{s} \mid \vec{o}) = \frac{1}{\Pr(\vec{o})} \prod_{t=1}^{|\vec{o}|} \Pr(s_t \mid s_{t-1}) \Pr(o_t \mid s_t)$$

Model

$$= \frac{1}{Z(\vec{o})} \prod_{t=1}^{|\vec{o}|} \phi_s(s_t, s_{t-1}) \phi_o(s_t, s_{t-1}, \vec{o}, t)$$

Log-linear form

$$\phi_o(t) = \exp\left(\sum_k \lambda_k f_k(s_{t-1}, s_t, \vec{o}, t)\right)$$

Parameters to fit

Feature function; might depend on whole $\underline{o}$

Feature functions and parameters

$$L = \sum_{\langle s,o \rangle \in D} \log\left(\frac{1}{Z(\vec{o})} \prod_{t=1}^{|\vec{o}|} \exp\left(\sum_k \lambda_k f_k(s_t, s_{t-1}, \vec{o}, t) \right) \right) - \boxed{\sum_k \frac{\lambda_k^2}{2\sigma^2}}$$

Penalize overfitting

Maximize total conditional likelihood over all instances

- Find $\partial L / \partial \lambda_k$ for each k and perform a gradient-based numerical optimization
- Efficient for linear state dependence structure

Feature
inside-noun-phrase (o_{t-1})
stopword (o_t)
capitalized (o_{t+1})
word=the (o_t)
in-person-lexicon (o_{t-1})
word=in (o_{t+2})
capitalized (firstmention$_{t+1}$) & capitalized (firstmention$_{t+2}$)
word=Republic (o_{t+1})
word=RBI (o_t) & header=BASEBALL (o_t)
header=CRICKET (o_t) & English-county (o_t)
company-suffix-word (firstmention$_{t+2}$)
location (o_t) & POS=NNP (o_t) & capitalized (o_t) & stopword (o_{t-1})
moderately-rare-first-name (o_{t-1}) & very-common-last-name (o_t)
word=the (o_{t-2}) & word=of (o_t)

Conditional vs. joint: results

Penn Treebank: 45 tags, 1M words training data

DT NN NN , NN , VBZ RB JJ IN
The asbestos fiber , crocidolite, is unusually resilient once

PRP VBZ DT NNS , IN RB JJ NNS TO PRP VBG
it enters the lungs , with even brief exposures to it causing

NNS WDT VBP RP NNS JJ , NNS VBD .
symptoms that show up decades later , researchers said .

Algorithm/Features	%Error	%OOVE*
HMM with words	5.69	45.99
CRF with words	5.55	48.05
CRF with words and orthography	4.27	23.76

Out-of-vocabulary error

Orthography: Use words, plus overlapping features: isCap, startsWithDigit, hasHyphen, endsWith… -ing, -ogy, -ed, -s, -ly, -ion, -tion, -ity, -ies

Summary

- Graphs provide a powerful way to model many kinds of data, at multiple levels
  - Web pages, XML, relational data, images…
  - Words, senses, phrases, parse trees…
- A few broad paradigms for analysis
  - Eigen analysis, conductance, random walks
  - Coupled distributions between node attributes and graph neighborhood
- Several new classes of model estimation and inferencing algorithms
- Exciting new applications

References

- [BrinP1998] The Anatomy of a Large-Scale Hypertextual Web Search Engine, WWW.
- [GoldmanSVG1998] Proximity search in databases. VLDB, 26—37.
- [ChakrabartiDI1998] Enhanced hypertext categorization using hyperlinks. SIGMOD.
- [BikelSW1999] An Algorithm that Learns What's in a Name. Machine Learning Journal.
- [GibsonKR1999] Clustering categorical data: An approach based on dynamical systems. VLDB.
- [Kleinberg1999] Authoritative sources in a hyperlinked environment. JACM 46.

References

- [CohnC2000] Probabilistically Identifying Authoritative Documents, ICML.
- [LempelM2000] The stochastic approach for link-structure analysis (SALSA) and the TKC effect. Computer Networks 33 (1-6): 387-401
- [RichardsonD2001] The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank. NIPS 14 (1441-1448).
- [LaffertyMP2001] Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. ICML.
- [BorkarDS2001] Automatic text segmentation for extracting structured records. *SIGMOD.*

References

- [NgZJ2001] Stable algorithms for link analysis. *SIGIR.*
- [Hulgeri+2001] Keyword Search in Databases. IEEE Data Engineering Bulletin 24(3): 22-32.
- [Hristidis+2002] DISCOVER: Keyword Search in Relational Databases. VLDB.
- [Agrawal+2002] DBXplorer: A system for keyword-based search over relational databases. ICDE.
- [Fagin2002] Combining fuzzy information: an overview. SIGMOD Record 31(2), 109–118.
- [Chakrabarti2002] Mining the Web: Discovering Knowledge from Hypertext Data

References

- [Tomlin2003] A New Paradigm for Ranking Pages on the World Wide Web. WWW.
- [Haveliwala2003] Topic-Sensitive Pagerank: A Context-Sensitive Ranking Algorithm for Web Search. IEEE TKDE.
- [LuG2003] Link-based Classification. ICML.
- [FaloutsosMT2004] Connection Subgraphs in Social Networks. SIAM-DM workshop.
- [PanYFD2004] GCap: Graph-based Automatic Image Captioning. MDDE/CVPR.
- [Balmin+2004] Authority-Based Keyword Queries in Databases using ObjectRank. VLDB.