Proximity Search in Social Networks

(Mining the Web)

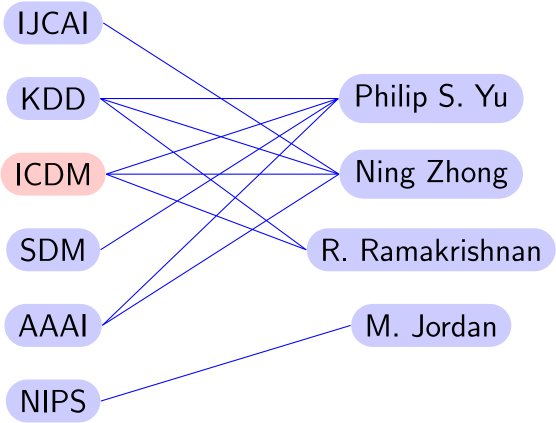
Soumen Chakrabarti

2024-10-31

Overview

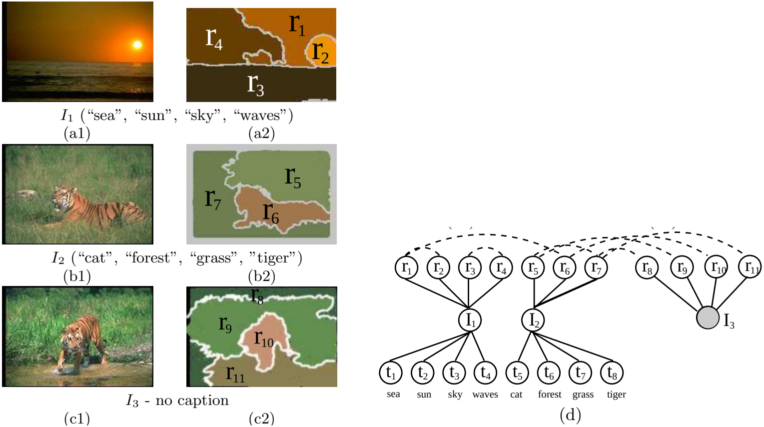
- Ranking nodes in a graph: widely applicable
- Graph proximity an important tool for ranking
- Desiderata of a proximity definition
- Hitting time, commute time, random walk with restarts
- Centerpiece subgraphs
- PageRank, large-scale computation
- Personalized PageRank, OBJECTRANK, HUBRANK
- Exponential and other decay profiles
- Hyperlink Induced Topic Search (HITS)
- HITS variants: SALSA, PHITS
- Topology sensitivity, score and rank stability
- Learning to rank in graph data models

Neighborhood search on graphs



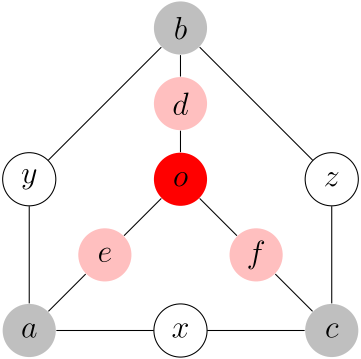
- Find and rank conferences closely related to ICDM

Captioning images [1]



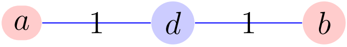
- Color regions as nodes
- Keywords in captions are nodes too
- "Heat up" test image, collect "hot" words

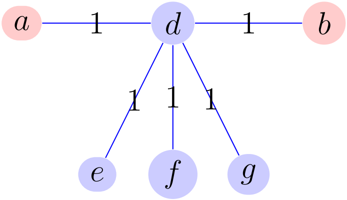
Centrality [2]



- Given a set of designated nodes a, b, c
- Find hub nodes well-connected to all of them
- o is best, followed by d, e, f; x, y, z are worst

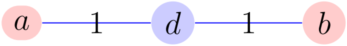
Proximity: Why not shortest path

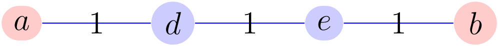




- Shortest path length between a and b is 2 in both cases
- In the second case, d is like someone who delivers pizza to both a and b, but many others too
- Proximity between a and b should be larger in the first case

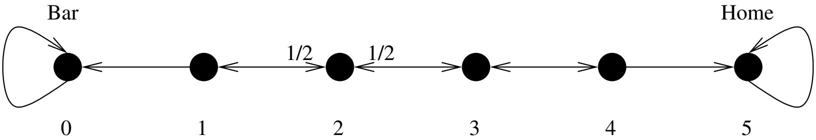
Proximity: Why not max flow





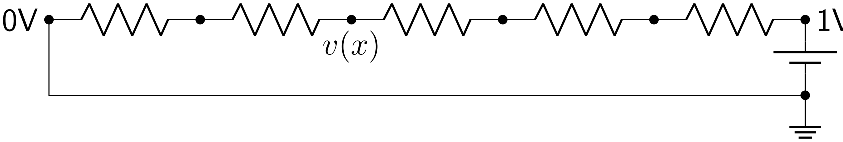
- Max flow in both cases is 1
- Intuitively, proximity between a and b is less in the second case
- Several signals needed:
  - Short path
  - Many paths
  - Few distractions on the way

Random walk (directed chain)



- Consider random walk on graph above
- State 0 and $N = 5$ are sink states
- Start from state $x \in [1, N-1]$
- $p(x)$ is the probability of reaching home before bar
- Aka escape probability
- $p(0) = 0$, $p(N) = 1$
- $p(x) = \frac{1}{2}p(x-1) + \frac{1}{2}p(x+1)$ for $x \in [1, N-1]$

Electrical network (chain)



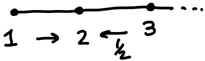
- Consider a 1V battery connected to five resistors R in series
- Node 0 is ground, node $N = 5$ is at 1V
- $v(x)$ is the voltage at node x
- For each $x = 1, 2, \ldots, N-1$,
$$\frac{v(x-1) - v(x)}{R} + \frac{v(x+1) - v(x)}{R} = 0$$
(no charge accumulates at node x)
- Solving for $v(x)$, we get same form as $p(x)$
$$v(x) = \frac{v(x+1) + v(x-1)}{2}$$

Undirected chain graph

- Undirected chain with nodes $1, 2, \ldots, n-1, n$
- Walk begins at node 1
  is expected walk duration finite?
- Walk terminates on visiting node n for the first time
- At each step, walker goes from a node to a neighbor uniformly at random (no trap node if graph connected)
- Define $c(i)$ as the expected number of visits to node i
- $c(n) = 1$ (the only visit to node n at which point walk terminates)
- Half the time surfer is at $n-1$, they go to visit n
  $\implies \frac{1}{2}c(n-1) = c(n) = 1 \implies c(n-1) = 2$
- Half the time surfer is at 2, they move to 1 next
  $\implies c(1) = 1 + \underbrace{\tfrac{1}{2}c(2)}$

Undirected chain graph (2)

- The 1 is for the initial visit to node 1, unaccounted by visits to other nodes
- If surfer is at 2 now, they must have been at node 1 or 3 in previous step $\implies c(2) = c(1) + \frac{c(3)}{2}$ ✓

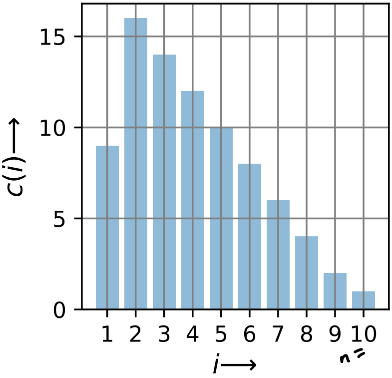


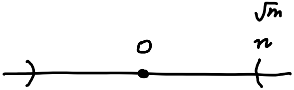
- For nodes $3 \leq i \leq n-2$, can show $c(i) = \dfrac{c(i-1) + c(i+1)}{2}$
- HW Can show that the general solution is

$$c(i) = \begin{cases} n-1, & i = 1 \\ 2(n-i) & 2 \leq i \leq n-1 \\ 1 & i = n \end{cases}$$

Before reaching n for the first time, we will visit 1 a total of n-1 times!

Undirected chain graph (3)





n^2 steps
m steps

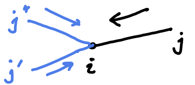
Toss fair coin m times
imbalance $\leq O(\sqrt{m})$
can show $\Omega(\sqrt{m})$ (?)

- Node 1 visited (can only turn around) about half as often as node 2 (passing left and right), then decreases monotonicaly
- Expected number of steps to go from 1 to n is sum of node visits, which is $\Theta(n^2)$

General resistive network

- For general graph, $r_{ij} = r_{ji}$ is the direct resistance between nodes i, j
- Let $g_{ij} = g_{ji} = 1/r_{ij} = 1/r_{ji}$ be the direct conductance
- $N(i) = \{j : \{i, j\} \in E\}$ are neighbors of i

$\frac{v(j)-v(i)}{r_{ij}}$



- Current flowing on the edge $\{i, j\}$ from j to i is $g_{ji}(v(j) - v(i))$
- No charge accumulation at any node i:

$$\forall i : \sum_{j \in N(i)} g_{ji}(v(j) - v(i)) = 0$$

(Kirchoff's current law)

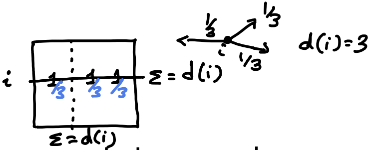
- Collecting terms, we get $v(i) = \dfrac{\sum_{j \in N(i)} g_{ji} v(j)}{\sum_{j \in N(i)} g_{ji}}$
- Voltage at node i is conductance-weighted average of voltage of neighbors j

General resistive network (2)

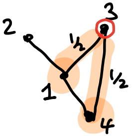
- Some nodes (e.g. connected to battery and ground) have pinned voltage
- Others can be estimated by iterative averaging
- Or by solving simultaneous linear equations
- The function v is called harmonic
- Minimizes the Laplacian $\sum_{\{i,j\}\in E} g_{ij}(v(i) - v(j))^2$
- HW Set the derivative of the Laplacian wrt any $v(i)$ to zero and solve for $v(i)$
- Will get the harmonic function described earlier

Random walks in undirected graphs

- Assume connected, undirected, unweighted graph all $g_{ij} = 1$
- Therefore no dead end node
- Symmetric node-to-node adjacency matrix $M \in \{0,1\}^{|V|\times|V|}$, where $M(i,j) = [\![i \text{ links to } j]\!]$ = $[\![j \text{ links to } i]\!]$
- If a node has ~~out~~degree $\cancel{\Delta}$ $d(u)$, walk to a neighbor uniformly at random w.p. $1/d(u)$
- Let $d(u)$ be degree of u, $d(u) \geq 1$ ✓
- Divide each row i of M by $d(i)$



- Row-stochastic matrix C, no longer symmetric in general
- p^0 is an initial multinomial probability distribution of being at each node initially
- Easy to see that if p^{t-1} is the distribution for time step $t-1$, then $p^t = C^\top p^{t-1}$ is the distribution for time step t



$$M = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 \\ 1 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 \\ 1/2 & 0 & 1/2 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 \\ 1 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 \\ 1/2 & 0 & 1/2 & 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{bmatrix}$$

at 3 surely

$$\underbrace{\begin{bmatrix} x_1^{(t)} & x_2^{(t)} & x_3^{(t)} & x_4^{(t)} \end{bmatrix}}_{\Sigma = 1} \begin{bmatrix} & & \\ & C & \\ & & \end{bmatrix} = \begin{bmatrix} x_1^{(t+1)} & x_2^{(t+1)} & x_3^{(t+1)} & x_4^{(t+1)} \end{bmatrix}$$

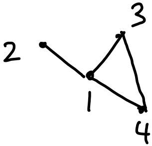
(1st order Markov model)

$$Pr(j \text{ @ } t+1) = \sum_{i=1}^{n} Pr(i \text{ @ } t)\, C[i,j] \quad ✓$$

$$p^{(t)} \, C = p^{(t+1)} \qquad ✓ \quad C^T p = 1\, p$$

steady-state visit prob. vector

$$t \to \infty \qquad p^{(t)} \approx p^{(t+1)} = ?$$



$$C = \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 1 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{bmatrix}$$

$$\begin{bmatrix} \frac{3}{8} & \frac{1}{8} & \frac{2}{8} & \frac{2}{8} \end{bmatrix} \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 1 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{bmatrix} = \begin{bmatrix} \underbrace{\frac{1}{8} + \frac{1}{8} + \frac{1}{8}}_{= \frac{3}{8}}, & \frac{1}{8}, & \frac{2}{8}, & \frac{2}{8} \end{bmatrix}$$

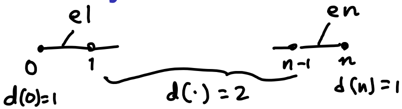
Random walks in undirected graphs (2)

- Two ways to write this:
  - Write p as a row vector and $p^t = p^{t-1}C$
  - Or write p as a column vector and $p^t = C^\top p^{t-1}$
- A stationary distribution p^0 is such that $p^1 = p^0$ $= \pi$
- Clearly a stationary distribution is an eigenvector of $C^\top$ with eigenvalue 1 ✓
- Claim: $\pi(u) = \frac{d(u)}{2m}$ is stationary, where m is the number of edges ▸ HW

  total degree
- Starting at node i, return time $R(i)$ is the expected number of steps after which we return to i next
- Claim: $R(i) = 1/\pi(i)$ $= \frac{2m}{d(i)}$

Long time between visits to low-degree nodes
Short – – – high-degree nodes

Undirected chain graph: stationary distribution and return times



- $n+1$ nodes $0, 1, \ldots, n$
- n undirected edges $\{i, i+1\}$ for $i = 0, 1, \ldots, n-1$
- Uniform transition probabilities
- Note, no loops at 0 and n , no sinks, infinite walk
- Connected, has well defined stationary distribution
- $\pi(0) = \pi(n) = 1/2n$, $\pi(1) = \cdots = \pi(n-1) = 2/2n = 1/n$
- $R(0) = R(n) = 2n$, $R(1) = \cdots = R(n-1) = n$

Expected time to reach a destination node

- Undirected chain graph again
- Start at node x
- $m(x)$ is the expected number of steps needed to reach 0 or N for the first time
- Is $m(x)$ finite for all x?
- Some properties

$$m(0) = m(N) = 0$$
$$m(x) = \tfrac{1}{2}m(x+1) + \tfrac{1}{2}m(x-1) + 1 \qquad x \neq 0, N$$

- From x go left or right, paying $+1$
- Then, from $x-1$ or $x+1$, recursively reach 0 or N

Expected time to reach a destination node (2)

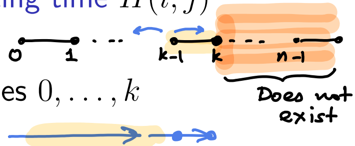
- In other words,

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1/2 & 1 & -1/2 & 0 & 0 & 0 \\ 0 & -1/2 & 1 & -1/2 & 0 & 0 \\ 0 & 0 & -1/2 & 1 & -1/2 & 0 \\ 0 & 0 & 0 & -1/2 & 1 & -1/2 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} m(0) \\ m(1) \\ m(2) \\ m(3) \\ m(4) \\ m(5) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

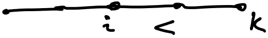
- Another linear system, not exactly the same as for voltages

Hitting time, commute time, cover time, . . .

- Starting at node i, the expected number of steps to reach node j for the first time is the hitting time $H(i, j)$
- What is $H(k-1, k)$?



- Consider the chain graph with nodes $0, \ldots, k$
- $R(k) = 2k$ $\quad \pi(k) = \frac{1}{2k}$
- To return from k to k, we need to step to $k-1$ and then hit k from $k-1$
- Therefore $\underbrace{R(k)}_{2k} = 1 + H(k-1, k)$
- Note that nodes to the right of k are irrelevant to this argument
- Thus, $H(k-1, k) = 2k - 1$



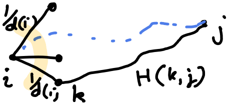
- Although with probability $1/2$, we walk straight from $k-1$ to k, the expected number of steps is large!

scales as $2k - 1$

$G(i,j) = H(i,j) + H(j,i)$

Hitting time, commute time, cover time, ... (2)

- How about $H(i,k)$ with $1 \le i < k \le n-1$?
- To hit k, must reach $k-1$ earlier
- $H(i,k) = H(i,k-1) + H(k-1,k) = H(i,k-1) + 2k-1$
- Unrolling the recurrence, $H(i,k) = k^2 - i^2$
- Special case: $H(0,k) = k^2$
- In a general graph,



$$H(i,j) = 1 + \frac{1}{d(i)} \sum_{(i,k)\in E} H(k,j)$$

Q: Is $H(i,j) = H(j,i)$ always?

- Again, a linear recurrence
- Starting at node i, cover time is the expected number of steps to visit all nodes
- Cover time of chain graph with nodes $0,..,n$ starting at node 0? $= H(n) = n^2$
- Cover time of complete graph?

$< n^2$ $= n^2$ $> n^2$

Start at node 0

$$\Pr(\text{reach node 1 first time at time } t) = \left(\frac{n-2}{n-1}\right)^{t-1} \frac{1}{n-1}$$

$$H(0,1) = \sum_{t \geq 1} t \left(\frac{n-2}{n-1}\right)^{t-1} \frac{1}{n-1} \overset{\text{ex}}{=} n-1$$

τ_i is the first (random) time step in which i distinct nodes have been visited.

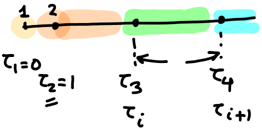
$$0 = \tau_1 < 1 = \tau_2 < \tau_3 \ldots < \tau_n = ?$$

suppose i nodes have been visited at least once each

$$\Pr(\text{next step visits a new unvisited node}) = \frac{n-i}{n-1}$$

$$\mathbb{E}(\tau_{i+1} - \tau_i) = \frac{n-1}{n-i}$$

= mean of a geom. distrib. with $p = \frac{n-i}{n-1}$

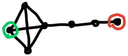


$$\mathbb{E}(\tau_n) = \sum_{i=1}^{n-1} \mathbb{E}[\tau_{i+1} - \tau_i] = \sum_{i=1}^{n-1} \frac{n-1}{n-i}$$

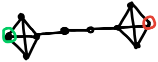
Telescopes

$$\approx n \log n < n^2 \quad \ldots \text{ expected}$$

<u>Ex</u>



Kite graph



Dumb-bell graph

Hitting time?

Cover time?

PageRank

→ { Visit rate/prob (steady state) as a measure of popularity | in degree | graph clicks

. . . we are involved in an "infinite regress": [an actor's status] is a function of the status of those who choose him; and their [status] is a function of those who choose them, and so ad infinitum.

(Children w/ largest number of direct friends not often elected)

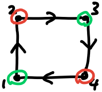
Directed

Seeley, 1949

$\boxed{c}$ ‖

- Random surfer roams around graph $G = (V, E)$
- Probability of walking from node i to j is $\Pr(j|i) = C(j, i)$ Transition matrix
- C is a $|V| \times |V|$ nonnegative matrix; each column sums to 1 (what about dead-end nodes?)
- Steady-state probability of visiting node i is its prestige
- The graph is directed and may have dead ends and components that are not strongly connected; quite different and messy compared to the undirected case

Ways to handle dead-end nodes



$Cp = p$

Amputation: Remove dead-ends, may cause other nodes to become dead-ends, keep removing

- How to assign scores to the removed nodes? $CCCp$

Self-loop: Each dead-end node i links to itself – creates a trap

- Still trapped at i; need to escape/restart

Sink node: Dead-end nodes link to a sink node, which links to itself (full status

- ~~Reasonable~~, but probability of visiting sink node means nothing

Makes significant difference to node ranks HW

Brin, Page
Motwani,
Winograd
1998

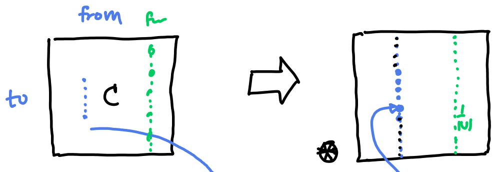
Solution

Dead end nodes → "teleportation"

Each step, w.p. $\begin{cases} (1-\alpha) & \text{we jump to a random node} \\ \alpha & \text{we walk node} \to \text{neighbor} \end{cases}$

1

0 incase of dead end nodes



$\Sigma > 0$ — not dead end

$\Sigma = 0$ — dead end

sparse

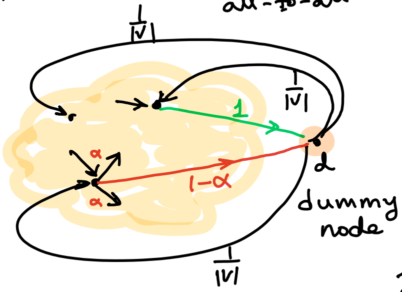
$\underbrace{\times\,\alpha + \frac{1}{|V|}(1-\alpha)}$

dense all-to-all

⊕ Modified C has no dead end

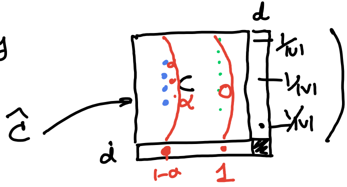
⊖ C is dense — all-to-all transitions possible

$C\,p = p$



- Closely related but not identical to ⊛

⊕ Sparse transitions



$|V| \times \frac{1}{|V|} = 1$

Each col adds to 1

$\hat{C}\,\hat{p} = \hat{p}$ - - - - relation between - - - - p — wanted

Steady state probabilities

Long after the walk gets under way, at any time step, the probability that the random surfer is at a given node

Need two conditions for well-defined steady-state probabilities of being in each state/node

- E must be irreducible: should be able to reach any v starting from any u A single SCC 
- E must be aperiodic: There must exist some ℓ_0 such that for every $\ell \geq \ell_0$, G contains a cycle of length ℓ

- PageRank
- Scalable computation
- "Personalization"
- Ent-rel search

Teleport

visit prob $\rightarrow$ $p = \alpha C p + (1-\alpha) r_q$ — Multi (nodes) — query

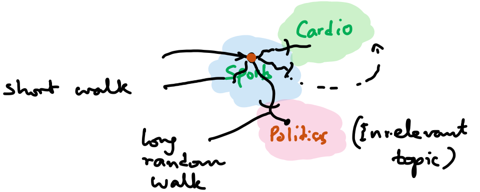
- Simple way to satisfy these conditions: all-to-all transitions

$$\tilde{C} = \overset{\text{walk}}{\alpha C} + \overset{\text{teleport}}{(1-\alpha)} \frac{1}{|V|} \mathbf{1}_{|V|\times|V|}$$

  $\mathbf{1}_{|V|\times|V|}$ is a matrix filled with 1s; $\tilde{C}$ also has columns summing to 1
- Random surfer walks with probability α, jumps with probability $1-\alpha$
- What is the "right" value of α?
- Is α a device to make E irreducible and aperiodic, or does it serve other purposes?

α – as large as possible (?) while keeping random walk well-defined

Topical locality in web graph
Best values of α depend on typical diameter of topical clusters



Keeping the graph sparse

$p^{(t+1)} \leftarrow \tilde{C}\, p^{(t)}$

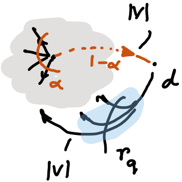
dence ⇒ :(

- Solve $p = \alpha C p + (1-\alpha)\mathbf{1}_{|V|\times 1}$ for steady-state visit probability $p \in \mathbb{R}^{|V|\times 1}$, with $p_i \geq 0$, $\|p\|_1 = \sum_i p_i = 1$
- Consider

from

$$\hat{C} = \begin{array}{c} |V| \\ d \end{array} \left[\begin{array}{c|c} \alpha C_{|V|\times|V|} & \frac{\mathbf{1}_{|V|\times 1}}{|V|} \\ \hline (1-\alpha)\mathbf{1}_{1\times|V|} & 0 \end{array}\right]$$

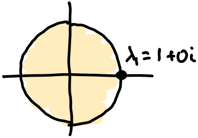
α, $(1-\alpha)$, d, r_q, to



- Dummy node d outside V, larger graph called $\hat{G}$
- Transition from every node $v \in V$ to d
- And a transition from d back to every node $v \in V$
- Recurrence can now be written as $\hat{p} = \hat{C}\hat{p}$ Homogeneous
- What is the relation between p and $\hat{p}$? HW ✓

reconstruct

HW $\hat{C}$ has an eigenvalue of 1



Power iterations

$u_1 = Au_1$

- Drop the hat for simpler notation: solve $x = Ax$
- Initialize $x^{(0)}$ "suitably" (random will do)
- Repeat $x^{(k)} \leftarrow Ax^{(k-1)}$ until $\|x^{(k)} - x^{(k-1)}\|$ becomes small
- For general A we would need to keep $\|x\|$ in control by rescaling after each iteration
- If A is stochastic, this is not (in principle) required
- A has eigen values $1 = \lambda_1 > |\lambda_2| \geq |\lambda_3| \geq \cdots$
- Suppose we can write $x = c_1u_1 + c_2u_2 + c_3u_3 + \cdots$ where u_i is the eigen vector associated with λ_i

First order

$$
\begin{aligned}
x^{(1)} &= Ax^{(0)} = c_1Au_1 + c_2Au_2 + c_3Au_3 + \cdots \\
&= c_1u_1 + c_2\lambda_2u_2 + c_3\lambda_3u_3 + \cdots
\end{aligned}
$$

- As we continue,
$$x^{(k)} = A^kx^{(0)} = c_1u_1 + \overset{\to 0}{\lambda_2^k}c_2u_2 + \overset{\to 0}{\lambda_3^k}c_3u_3 + \cdots \xrightarrow[k\to\infty]{} c_1u_1$$

Speeding up power iterations

Remember last three iterates

- Suppose at some iteration we are lucky and

$$x^{(k-2)} = u_1 + c_2 u_2 \quad \dots$$

- In the next two iterations,

$$x^{(k-1)} = Ax^{(k-2)} = u_1 + c_2\lambda_2 u_2$$

$$\text{and} \quad x^{(k)} = Ax^{(k-1)} = u_1 + c_2\lambda_2^2 u_2$$

- Using the three equalities, solve for

$$\checkmark\, x^{(k)} \xleftarrow{\text{tune}} (u_1)_i = x_i^{(k-2)} - c_2(u_2)_i$$

Instead of

$x^{(k)} \leftarrow A\,x^{(k-1)}$

$$= x_i^{(k-2)} - \frac{(x_i^{(k-1)} - x_i^{(k-2)})^2}{x_i^{(k)} - 2x_i^{(k-1)} + x_i^{(k-2)}}$$

Approx

Interpolation pseudocode

- In reality, we won't get lucky
- Therefore, repeat tuning x once in a while
- Can do interpolation of higher order
- Tradeoff between number and cost of iterations

2

```
Input: Right-stochastic matrix A, initial probability vector
  x^(0) with ||x^(0)||_1, termination tolerance ε.
repeat
    x^(k) ← A x^(k−1)
    once in a while, tune x^(k)
    k ← k + 1
until ||x^(k) − x^(k−1)||_1 < ε
```

Input: Right-stochastic matrix A, initial probability vector $x^{(0)}$ with $\|x^{(0)}\|_1$, termination tolerance ϵ.

repeat

$x^{(k)} \leftarrow A\,x^{(k-1)}$

once in a while, tune $x^{(k)}$

$k \leftarrow k+1$

until $\|x^{(k)} - x^{(k-1)}\|_1 < \epsilon$

1 Warm-start

Personalized PageRank

- Key is to allow general query-dependent teleport r
- Multinomial distribution over nodes
- Usually very sparse, will exploit this later
- Equations remain quite similar

$$p_r = \alpha C p_r + (1-\alpha) r$$

Dense

$$p_r = (1-\alpha)(\mathbb{I} - \alpha C)^{-1} r$$

$0 < \alpha < 1$

$\alpha^k \to 0$

Approximate the sum

$p(r)$?

$$p_r = (1-\alpha)\boxed{\left(\sum_{k \geq 0} \alpha^k C^k\right)} r$$

convergent

- Note, p_r is a linear function of r

$$p_{ar} = a p_r$$
$$p_{r_1 + r_2} = p_{r_1} + p_{r_2}$$

vec

$\vec{term} = (1-\alpha)\vec{r}$
$\vec{sum} = 0$
for iterations ...
sum ← sum + term
term ← αC term

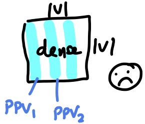
- However, $(\mathbb{I} - \alpha C)^{-1}$ is dense and never computed explicitly

Personalized PageRank (2)

$$\delta_u = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad u$$

- Impulse teleport δ_u defined as $\delta_u(v) = [\![v = u]\!]$
- Personalized PageRank vector (PPV) for node u, denoted PPV_u, is p_{δ_u} $r = \delta_u$ "basis"
- By linearity, if PPV_u is available for all u, p_r for arbitrary teleport r can be computed as $\sum_u r(u)\,\mathrm{PPV}_u$
- The problem is that it is time and space consuming to compute and store PPV_u for all u

$\in \mathbb{R}^{|V|}$ $|V|$



Approx and truncate, store sparse columns

Compute all N PPVs ?

$$q \to \begin{cases} q_{c1} \\ q_{c2} \\ \vdots \end{cases}$$

$$\mathrm{score}(u|q) = \sum_c q_c\, p_{r_c}$$

Topic-sensitive pagerank.

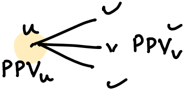
- Classify each page into a topic
- For each topic c
  $r_c \leftarrow$ unif over pages
  p_{r_c}

The pushdown property

- From definition, $p_r = \alpha C p_r + (1-\alpha) r$
- Claim: $p_r = p_{\alpha C r} + (1-\alpha) r$
- αC has been "pushed down" to the subscript
- Note, $\alpha C r$ is not a valid multinomial teleport distribution; this is just an identity

$$\begin{aligned}
p_{\alpha C r} + (1-\alpha) r &= (1-\alpha)\left(\sum_{k\geq 0} \alpha^k C^k\right) \alpha C r + (1-\alpha) r \\
&= (1-\alpha)\left(\sum_{k\geq 1} \alpha^k C^k\right) r + (1-\alpha) \alpha^0 \underbrace{C^0}_{\mathbb{I}} r \\
&= (1-\alpha)\left(\sum_{k\geq 0} \alpha^k C^k\right) r = p_r
\end{aligned}$$

Hub decomposition



- If we don't know PPV_u but have PPV_v available for all outneighbors, we can reconstruct PPV_u

HW
$$\mathrm{PPV}_u = \alpha \sum_{(u,v)\in E} C(v,u)\,\mathrm{PPV}_v + (1-\alpha)\delta_u$$

- Need to prove that

$$\alpha \sum_{(u,v)\in E} C(v,u)\,\mathrm{PPV}_v = \sum_{(u,v)\in E} \alpha C(v,u) p_{\delta_v} = p_{\alpha C \delta_u}$$

- By linearity,

$$\text{Lhs} = \sum_{(u,v)\in E} \alpha C(v,u) p_{\delta_v} = p_{\left(\sum_{(u,v)\in E} \alpha C(v,u)\right)\delta_v}$$

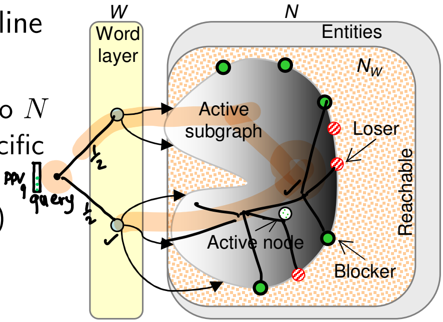
and
$$\sum_{(u,v)\in E} C(v,u)\delta_v = C\delta_u$$

HW

and the result follows

Sparse teleport

- Choose hub node subset $H \subset V$
- Precompute and store PPV_h for all $h \in H$
- Prepare entity graph N offline
- On query submission ...
  - Add word nodes W, link to N
  - Quickly identify query-specific active subgraph boundary (Active $\subset$ Reachable $\subset N$)



$\underset{v}{\mathrm{argmax}^{(K)}}\ \mathrm{PPV}_q \to$ response

- Blockers are nodes in H whose PPVs have been precomputed and stored
- Losers are nodes too "far" from word nodes to influence word PPVs appreciably

Iterative PPV updates

- Set $\widehat{\mathrm{PPV}}_u = \delta_u$ for losers u
- Load approximate $\widehat{\mathrm{PPV}}_u$ from cache for blockers u
- For active nodes u that are not blockers or losers, update
$$\widehat{\mathrm{PPV}}_u \leftarrow \alpha \sum_{(u,v)\in E} C(v,u)\widehat{\mathrm{PPV}}_v + (1-\alpha)\delta_u$$
until convergence (using Decomposition Theorem)
- Can show PPV convergence similar to Jeh and Widom, even using fixed approximate $\widehat{\mathrm{PPV}}_u$ for blockers and losers
- Add up word PPVs and report top-k entity nodes

PageRank by asynchronous "message passing"

residual energy

Recall $p_r = \alpha C p_r + (1-\alpha) r = (1-\alpha)\left(\sum_{k\geq 0} \alpha^k C^k\right) r$

term   sum

Energy

$q \leftarrow r, p \leftarrow \vec{0}$

while $|q|$ is large **do**

  $p \leftarrow p + (1-\alpha) q$

  $q \leftarrow \underbrace{\alpha C}_{<1} q$ mat-vec prod.

iters	p
0	$\vec{0}$
1	$(1-\alpha)r$
2	$(1-\alpha)r + (1-\alpha)\alpha C r$

Residual

$q \leftarrow r, p \leftarrow \vec{0}$

while some $q(u)$ large **do**   victim

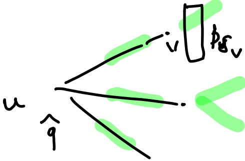
  $\hat{q} \leftarrow q(u), q(u) \leftarrow 0$ drain energy of u

  $p(u) \leftarrow p(u) + (1-\alpha)\hat{q}$

  for $(u, v) \in E$ **do**

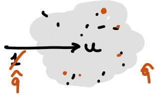
    $q(v) \leftarrow q(v) + \alpha C(v, u)\hat{q}$

Advantage of asynchronous version is that it can use PPVs at hub nodes



Stopping at hubs

Precomp and store $\mathrm{PPV}_u = p_{\delta_u}$
only for $u \in H \subset V$



$q \leftarrow r, \hat{p} \leftarrow \vec{0}$

while some $q(u)$ is large **do**

  $\hat{q} \leftarrow q(u), q(u) \leftarrow 0$ drain victim

  if $u \in H$ **then**

    $\hat{p} \leftarrow \hat{p} + \hat{q}\,\mathrm{PPV}_u$ /* $\hat{q}$ blocked */

  else

    $\hat{p}(u) \leftarrow \hat{p}(u) + (1-\alpha)\hat{q}$

    for $(u, v) \in E$ **do**

      $q(v) \leftarrow q(v) + \alpha C(v, u)\hat{q}$ /* $\hat{q}$ "pushed" */

Correctness

- $\hat{p} + p_q$ is invariant across the while loop
- In the beginning, $\hat{p} = \vec{0}$, $q = r$, $p_q = p_r$
- At the end, $q \approx \vec{0}$

$\therefore$ $p_q \approx (1-\alpha)(\mathbb{I} - \alpha C)^{-1}\vec{0} \approx \vec{0}$

$\therefore$ $\hat{p} \approx p_r$

Termination

- Suppose we quit when $\|q\|_1 < \epsilon_{\mathsf{push}}$
- Each iteration reduces $\|q\|_1$ by $\geq (1-\alpha)\epsilon_{\mathsf{push}}$
- Begin with $\|q\|_1 = \|r\|_1$ ($= 1$ for legal teleports)

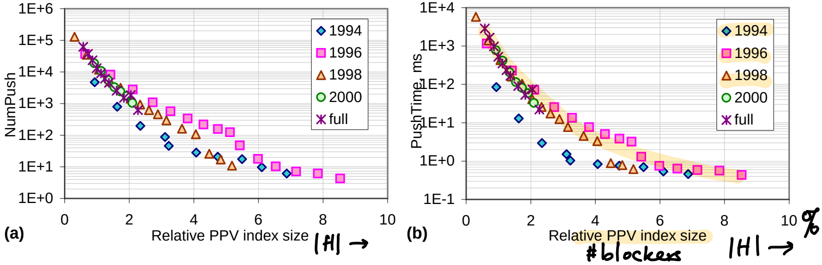
$\therefore$ At most $\|r\|_1/(1-\alpha)\epsilon_{\mathsf{push}}$ loop iterations

A more efficient implementation

$q \leftarrow r, N_{H,r} \leftarrow \vec{0}, B_{H,r} \leftarrow \vec{0}$
while $\|q\|_1 > \epsilon_{\text{push}}$ **do**
    pick node u with largest $q(u) > 0$ /* delete-max */
    $\hat{q} \leftarrow q(u)$, $q(u) \leftarrow 0$
    if $u \in H$ **then**
        $B_{H,r}(u) \leftarrow B_{H,r}(u) + \hat{q}$ /* blocker score */
    else
        $N_{H,r}(u) \leftarrow N_{H,r}(u) + (1-\alpha)\hat{q}$
        for each out-neighbor v of u **do**
            $q(v) \leftarrow q(v) + \alpha C(v,u)\hat{q}$ /* increase-key */
return $N_{H,r} + \sum_{h \in H} B_{H,r}(h)\,\mathrm{PPV}_h$

$N_{H,r}(u)$ is PageRank transmitted from r to u bypassing H

Trading PPV index space vs. time



- Temporal snapshots of CiteSeer graph
- HubRank algorithm for selecting H – blockers
- Query time tracks number of pushes closely
- "Relative PPV index size" means bytes in PPV index divided by bytes to store edge list
- Query time decays steeply with index size
- Large changes in graph sizes (1994–now) have little effect on tradeoff profile

PageRank variants and applications

- Sampling nodes from the Web graph
- Topic-sensitive PageRank / personalized
- Page staleness – reversed edges
- Link spam and link farm detection

queries

r_q < past visits

- Automatic topic classifier on every crawled page
- Separate Web graph into topical subgraphs
- Precomp & store PageRank for each topic separately

Given query q
- find closest topic(s)
- Use pagerank only from this/these in scoring.

PageRank

a → b

high prestige ⇒ high prestige

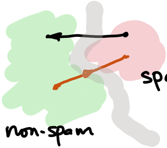
Staleness

a ⇄ b

a ⇐ b "stale"

a stale

base staleness score
- 404 100% stale
- change detect
- Random URL pads https://host/path



✓ SpamRank

Sampling from graphs

- Real social networks can be very large
- Algorithms need to be tested for scalability
- Try on representative sample graphs of increasing size
- Given a social network, how to draw a ‘representative’ sample?
- How to define ‘representative’?
- Sampling from Large Graphs

Uniform sampling: Motivation

- What fraction of Web pages are in the `.jp` domain?
- To start with, the Web has an infinite number of pages
- But crawlers usually turn it into a finite graph via policies that limit link expansion
- Consider that finite graph as the universe
- Why might the above question be useful?
- E.g., if you want to partition a world-wide crawl ahead of time between data centers
- E.g., if you want to build an estimate of word IDF without crawling the whole Web
- E.g., you are Dmoz.org and want to plan resources for major topics in proportion to the population (number of pages) on each topic

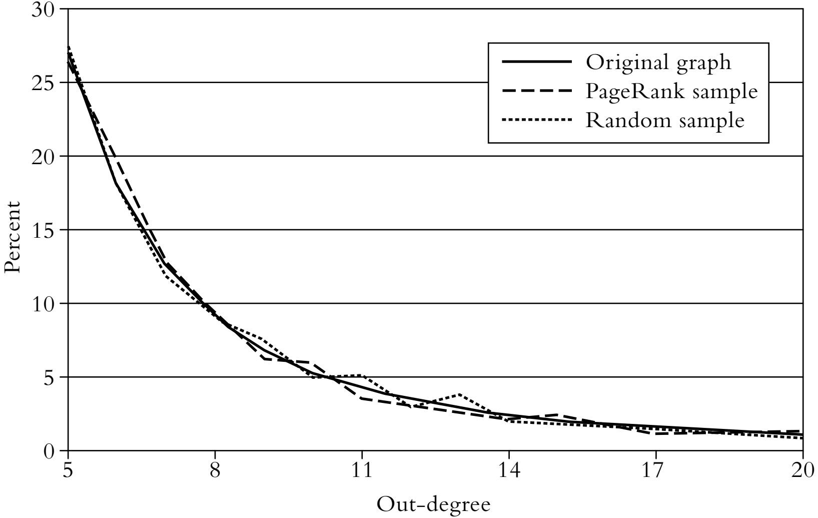
Forward walk like PageRank

- Starting from some seed nodes
- Do a PageRank-like random walk
- Except that we cannot teleport to a node uniformly at random, or we would have already solved the problem
- Instead teleport uniformly to a node encountered thus far in the random walk
- Causes an initial bias keeping us too close to home
- Sampling the trails uniformly is a bad idea
- Because large-PageRank nodes are visited more frequently by definition
- One way to compensate is to sample the trails with probability inversely proportional to estimated PageRank

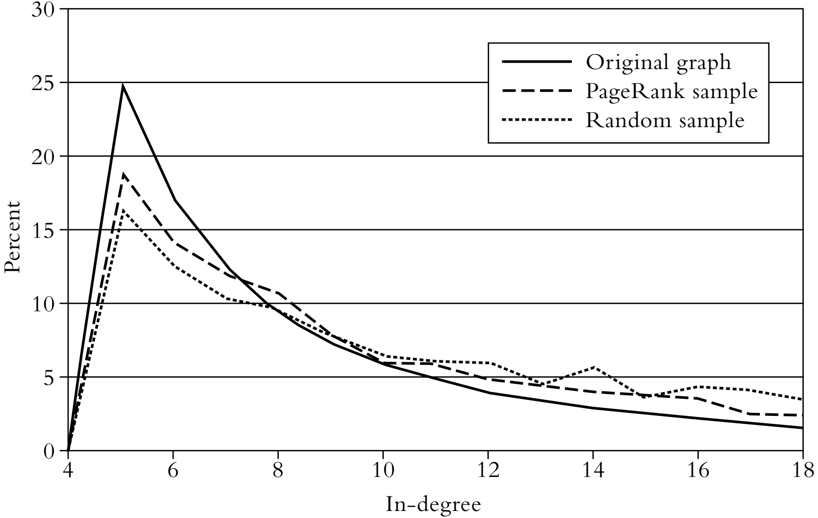
Biased sampling heuristic

- Run a random walk for some time, then sample from the traces
$$\Pr(v \text{ is sampled}) = \Pr(v \text{ is crawled}) \; \Pr(v \text{ is sampled} | v \text{ is crawled})$$
- Let PageRank be p^*
- In a walk of length w, node v is visited about $w\,p^*(v)$ times
- If walks at of length less than $\sqrt{|V|}$, any node is visited at most once whp HW
- Assuming the above, can approximate
$\Pr(v \text{ is crawled}) \approx \mathbb{E}(\text{number of times } v \text{ is visited}) = w\,p^*(v)$
- Therefore we must set
$$\Pr(v \text{ is sampled} | v \text{ is crawled}) \propto \frac{1}{p^*(v)}$$
- Note, p^* not known either, must crudely approximate

Forward random walks: out-degree



Forward random walks: in-degree



Bidirectional walk on regular graphs

- We have seen that in undirected graphs, $\pi(u) = \mathrm{Degree}(u)/2|E|$
- If all nodes have the same degree (regular graph), $\pi(u) = 1/|V|$ for all u
- Estimate an upper bound Δ to the maximum node degree
- For each node u, add $(\Delta - \mathrm{Degree}(u))/2$ undirected self-loops
- (Almost) regular graph

Ideal “undirected” random walk

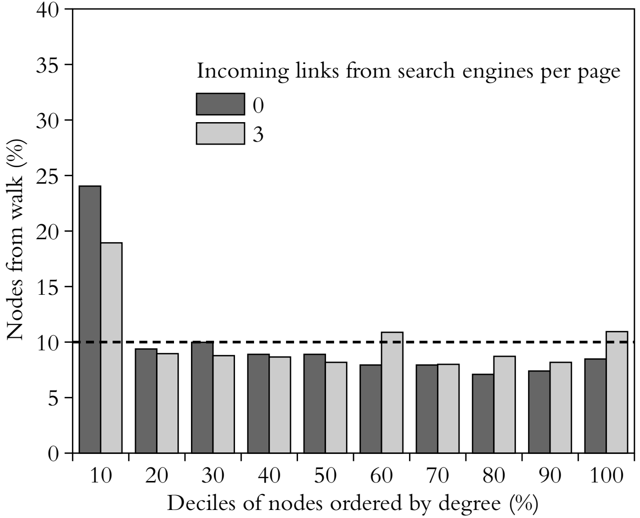
```
pick starting node
for given walk length do
    consider current node v on the walk
    self-loop at v a random number of times, which is distributed geometrically with mean 1 − N_v / N_max
    pick the next node uniformly at random from in- and out-neighbors of v
```

Practical “undirected” random walk

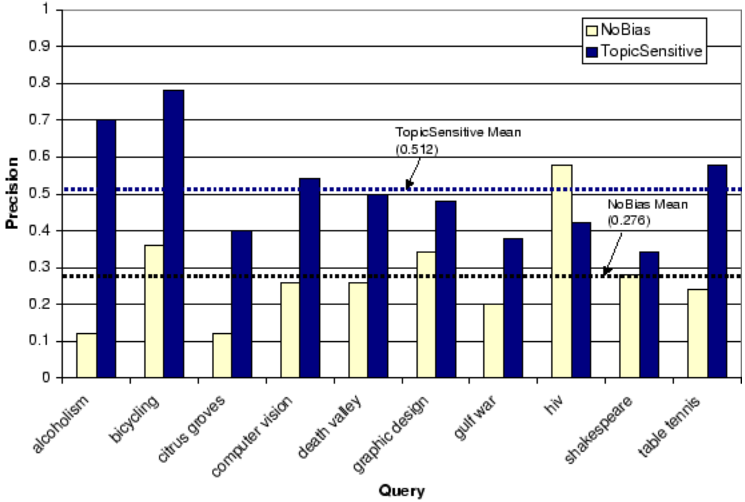
pick start node u and set $I_u = O_u = \emptyset$
for given walk length **do**
  consider current node v on the walk
  if v has not been visited before **then**
    get in-neighbors of v using a search engine
    get out-neighbors of v by scanning for `HREF`s
    add new neighbors w to I_v and O_v only if w has not been visited already
  self-loop at v as in the Ideal Random Walk
  pick the next node uniformly at random from $I_v \cup O_v$

Made with Xodo PDF Reader and Editor

"Undirected" random walks result



Topic-sensitive Pagerank



- Details of how query is “projected” to topic space
- Clear improvement in precision

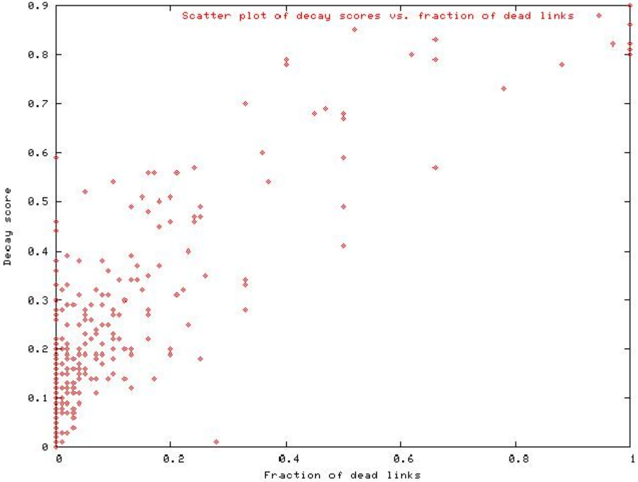
Page staleness

"A page is stale if it is inaccessible, or if it links to many stale pages"—to find how stale a page u is,

$v \leftarrow u$
for ever **do**
    if page v is inaccessible **then**
        return $s(u) = 1$
    toss a coin with head probability σ
    if head **then return** $s(u) = 0$ /* with probability σ */
    else
        choose $w : (v, w) \in E$ with probability $\propto C(w, v)$
        $v \leftarrow w$

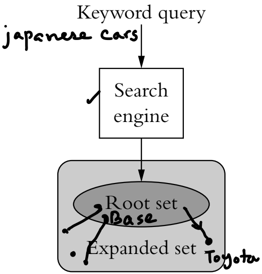
$$s(u) = \begin{cases} 1, & u \in D \\ (1-\sigma)\sum_v C(v,u)\, s(v), & \text{otherwise} \end{cases}$$

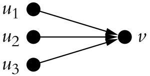
Page staleness: Experience



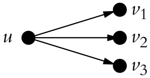
Staleness of a page is generally larger than the fraction of dead links on the page would have you believe

Hyperlink induced topic search (HITS) query-based





$a(v) = h(u_1) + h(u_2) + h(u_3)$



$h(u) = a(v_1) + a(v_2) + a(v_3)$

$\vec{a} \leftarrow (1, \ldots, 1)^T, \vec{h} \leftarrow (1, \ldots, 1)^T$
while $\vec{h}$ and $\vec{a}$ change "significantly" **do**
  $\vec{h} \leftarrow E\vec{a}$
  $\ell_h \leftarrow \left\|\vec{h}\right\|_1 = \Sigma_w h[w]$
  $h \leftarrow h/\ell_h$
  $\vec{a} \leftarrow E^T h_0 = E^T E \vec{a}_0$
  $\ell_a \leftarrow \left\|\vec{a}\right\|_1 = \Sigma_w a[w]$
  $\vec{a} \leftarrow \vec{a}/\ell_a$
end while

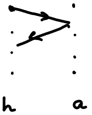
- Need for initial graph expansion
- Appeal to correlated link locality and content similarity
- Weakens rapidly with distance; radius-1 works ok
- Each node has a hub score and authority score
- Defined in terms of each other

Note: the graph being analyzed is query-dependent

HITS and SVD/LSI

Two notions of node quality
- original results/authorities
- surveys/hubs

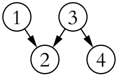
- Let $E(i,j) = [\![(i,j)$ is an edge$]\!]$
- Let h be the hub score vector and a be the authority score vector
- Iterations amount to $a \leftarrow E^\top h$ and $h \leftarrow Ea$
- In two iterations, $h \leftarrow EE^\top h$
- Power iterations converging to dominant eigenvector of $A = EE^\top$, a symmetric matrix
- A has $m = |V|$ eigenvectors, stack them vertically to get $U = u_{\cdot 1}, u_{\cdot 2}, \ldots, u_{\cdot m}$
- $U^\top A = \Lambda U^\top$, where Λ is a diagonal matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \geq 0$
- Meanwhile suppose the SVD of E is $E_{m\times m} = U_{m\times m}\Sigma_{m\times m}V^\top_{m\times m}$ where $U^\top U = \mathbb{I}_{m\times m}$ and $V'V = \mathbb{I}_{m\times m}$



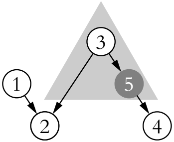
HITS and SVD/LSI (2)

- $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_m)$ of singular values, with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots \sigma_m = 0$, for some $0 < r \leq m$
- $A = EE^\top = U\Sigma\underline{V'V}\Sigma U' = U\Sigma\mathbb{I}\Sigma U' = U\Sigma^2 U'$, $\therefore AU = U\Sigma^2$, or $U'A = \Sigma^2 U'$

Topology sensitivity

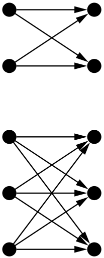


$$E = \begin{vmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{vmatrix} \quad ; \quad E^T E = \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{vmatrix}$$



$$E = \begin{vmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{vmatrix} \quad ; \quad {}^T E = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{vmatrix}$$

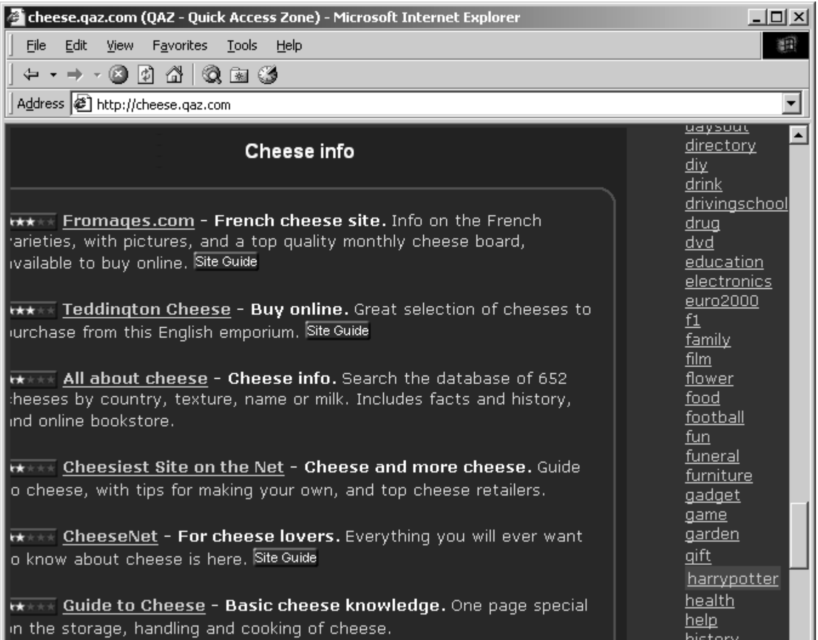
(a)



(b)

- In (a), old graph: $a_1 = a_3 = 0$, $a_2 = 2a_2 + a_4$, $a_4 = a_2 + a_4$
- New graph: $a_1 = a_3 = 0$, $a_2 = 2a_2 + a_5$, $a_4 = a_4$, $a_5 = a_2 + a_5$
- Dramatic demotion of authority of node 4
- In (b), larger core captures all hub and authority score; smaller core scores dwindle to zero

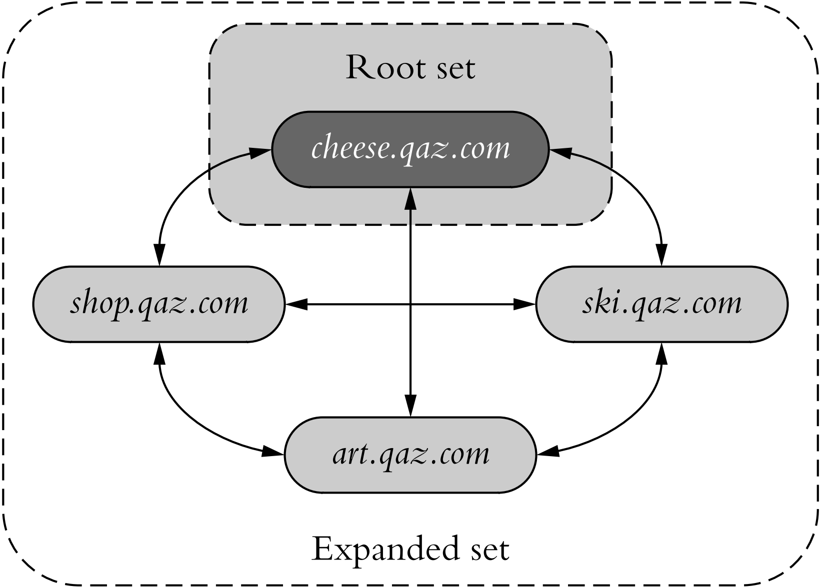
Clique attack



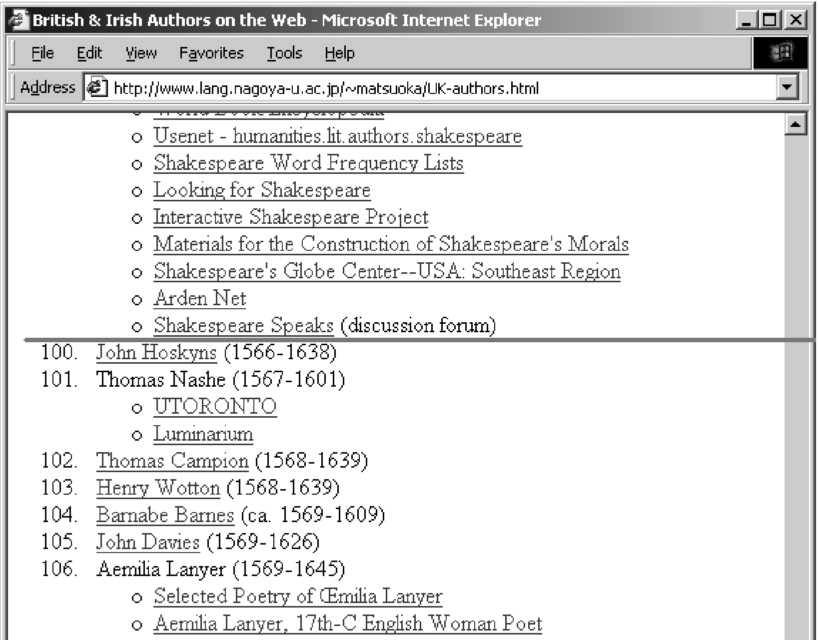
Hub $\neq$ authority

- Economics of creating (or corrupting) a hub not the same as an authority
- A node with high in-degree is likely to be a good authority
- But a node with high out-degree is not necessarily a good hub
- Also see the SALSA paper [11]

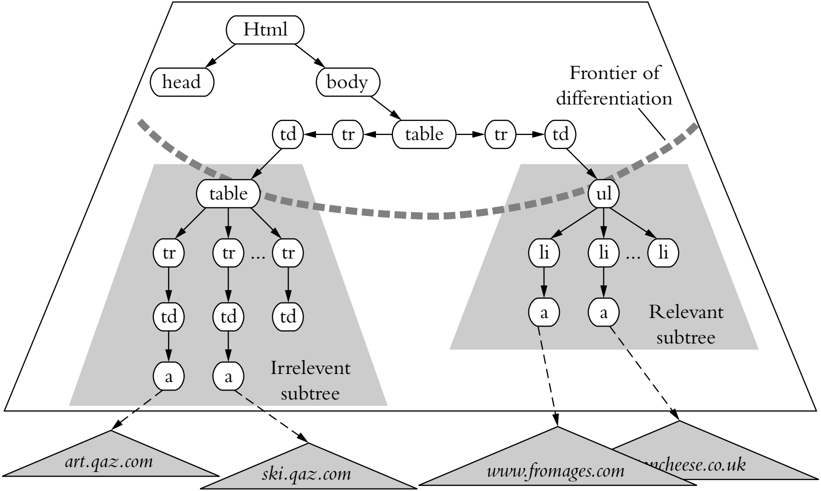
Clique attack



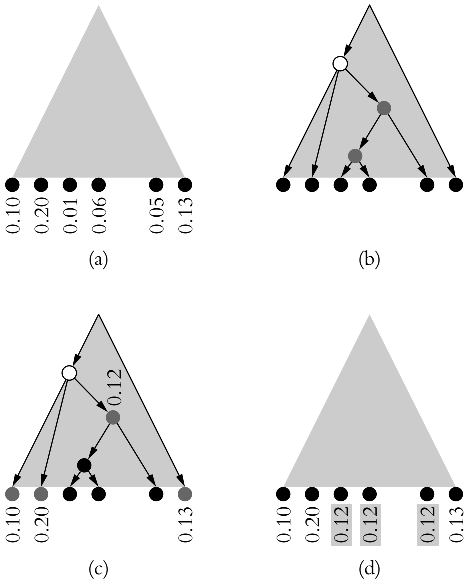
Topic drift in a single hub page



Segmenting the DOM into microhubs



Hub score aggregation on microhubs



- Authorities transfer score to hubs
- Don't aggregate all into one hub score
- Segment the DOM tree
- Aggregate only relevant subtree
- And push back to authorities
- Prevent irrelevant hub scores from teaming up

Deeper analysis of link-based ranking [12]

- "The seminal papers of Kleinberg [1998], and Brin and Page [1998] introduced the area of Link Analysis Ranking"
- "The ground-breaking work of Kleinberg [1998,1999], and Brin and Page [1998]"
- "This leads us to question the definition of the hub weight and consequently, the other symmetric nature of HITS."
- "we challenge both implicit properties of HITS"
- Nice summary of many HITS variations and their effect on ranking, with illustrative examples
- "HITS and PageRank algorithms emerge as the worst algorithms"
- "The qualitative evaluation reveals that all algorithms fall, to some extent, victim to topic drift. That is, they promote pages that are not related to the topic of the query."

Deeper analysis of link-based ranking [12] (2)

- Definition of expanded set collection ad-hoc
- For all algorithms, could be garbage in, garbage out
- Adding content analysis helps greatly

Stability — map

- PageRank vs. HITS
- Stability analysis — extremal cases, not typical or random perturbation
- Score vs. rank stability
- If a small fraction of the graph (structure or edge weights) changes, how severely can the node scores or ranks change?
- E.g., change a constant number of edges so that the new node ranking has $\Omega(n^2)$ discordant pairs wrt old ranking

PageRank score stability

- V kept fixed
- Nodes in $P \subset V$ get incident links changed in any way (additions and deletions)
- Thus G perturbed to $\tilde{G}$
- Let the random surfer visit (random) node sequence $X_0, X_1, \ldots$ in G, and $Y_0, Y_1, \ldots$ in $\tilde{G}$
- Coupling argument: instead of two random walks, we will design one joint walk on (X_i, Y_i) such that the marginals apply to G and $\tilde{G}$

Coupled random walks on G and $\tilde{G}$

- Pick $X_0 = Y_0 \sim \mathrm{Multi}(r)$
- At any step t, with probability $1 - \alpha$, reset both chains to a common node using teleport r: $X_t = Y_t \in_r V$
- With the remaining probability of α
  - If $x_{t-1} = y_{t-1} = u$, say, and u remained unperturbed from G to $\tilde{G}$, then pick one out-neighbor v of u uniformly at random from all out-neighbors of u, and set $X_t = Y_t = v$.
  - Otherwise, i.e., if $x_{t-1} \neq y_{t-1}$ or x_{t-1} was perturbed from G to $\tilde{G}$, pick out-neighbors X_t and Y_t independently for the two walks.

Analysis of coupled walks

Let $\delta_t = \Pr(X_t \neq Y_t)$; by design, $\delta_0 = 0$.

$$\begin{aligned}
\delta_{t+1} &= \Pr(\text{reset at } t+1)\ \Pr(X_{t+1} \neq Y_{t+1} | \text{reset at } t+1) + \\
&\qquad \Pr(\text{no reset at } t+1)\ \Pr(X_{t+1} \neq Y_{t+1} | \text{no reset at } t+1) \\
&= \Pr(\text{reset at } t+1)\, 0 + \alpha\ \Pr(X_{t+1} \neq Y_{t+1} | \text{no reset at } t+1) \\
&= \alpha\big(\Pr(\underline{X_{t+1} \neq Y_{t+1}}, X_t \neq Y_t | \text{no reset at } t+1) + \\
&\qquad \Pr(X_{t+1} \neq Y_{t+1}, X_t = Y_t | \text{no reset at } t+1)\big)
\end{aligned}$$

The event $X_{t+1} \neq Y_{t+1}, X_t = Y_t$ can happen only if $X_t \in P$.
Therefore we can continue the above derivation as follows:

$$\begin{aligned}
\delta_{t+1} &= \ldots \\
&\leq \alpha\big(\Pr(X_t \neq Y_t | \underline{\text{no reset at } t+1}) + \\
&\qquad \Pr(X_{t+1} \neq Y_{t+1}, X_t = Y_t, \underline{X_t \in P} | \text{no reset at } t+1)\big)
\end{aligned}$$

Analysis of coupled walks (2)

$$= \alpha\big(\Pr(X_t \neq Y_t) + \Pr(X_{t+1} \neq Y_{t+1}, X_t = Y_t, \underline{X_t \in P} | \text{no reset at } t+1)\big)$$

$$\leq \alpha\big(\Pr(X_t \neq Y_t) + \Pr(X_t \in P)\big)$$

$$= \alpha\left(\delta_t + \textstyle\sum_{u \in P} p_u\right),$$

(using $\Pr(H, J|K) \leq \Pr(H|K)$, and that events at time t are independent of a potential reset at time $t+1$)

Unrolling the recursion,

$\delta_\infty = \lim_{t\to\infty} \delta_t \leq \left(\sum_{u \in P} p_u\right) / (1-\alpha)$ HW

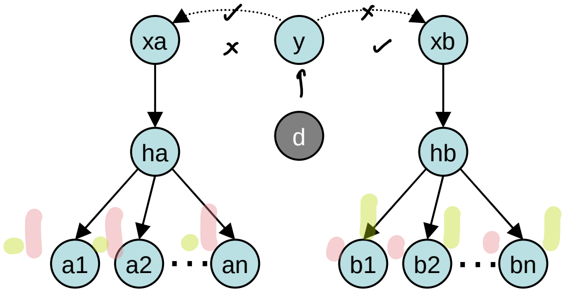
- Standard result: If the probability of a state disagreement between the two walks is bounded, then their Pagerank vectors must also have small L_1 distance to each other. In particular,

$$\|p - \tilde{p}\|_1 \leq \frac{2\sum_{u \in P} p_u}{1-\alpha}$$

Analysis of coupled walks (3)

- Lower the value of α, the more the random surfer teleports and more stable is the system
- Gives no direct guidance why α should not be set to exactly zero! (WAW talk)

PageRank rank stability: adversarial



- G formed by connecting y to x_a, $\tilde{G}$ by connecting y to x_b
- $\Omega(n^2)$ node pairs flip Pagerank order HW
- I.e., L_1 score stability does not guarantee rank stability
- Can "natural" social networks lead often to such tie-breaking?

Effect of graph perturbation on HITS

- Focus on a, dominant eigenvector of $E^\top E = S$, say
- G changes to $\tilde{G}$, E to $\tilde{E}$, S to $\tilde{S}$, a to $\tilde{a}$
- Goal is to keep $\tilde{S}$ close to S but push $\tilde{a}$ far from a (in L_1 sense, say)
- Let $\lambda_1 \geq \lambda_2$ be the two largest eigenvalues of S
- Let $\delta = \lambda_1 - \lambda_2 > 0$, also called eigengap of S
- S has factorization
$$S = U \begin{pmatrix} \lambda_1 & 0 & \mathbf{0} \\ 0 & \lambda_2 & \mathbf{0} \\ 0 & 0 & \Lambda \end{pmatrix} U',$$
- Columns of U are eigenvectors of S
- Λ is a diagonal matrix containing the remaining eigenvalues

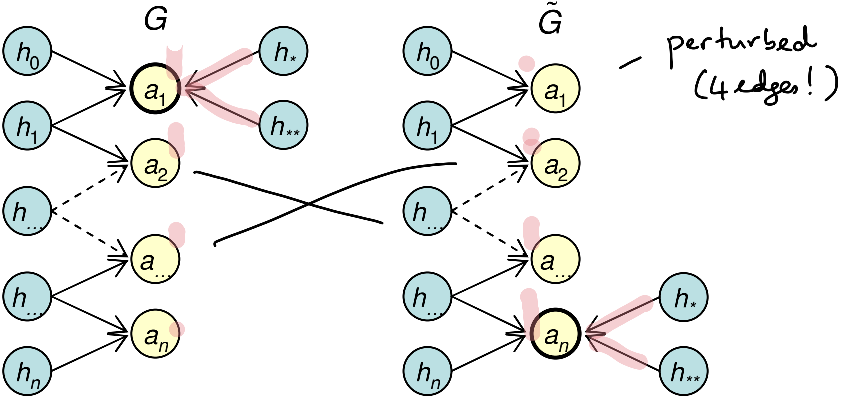
Effect of graph perturbation on HITS (2)

- The new matrix is defined as
$$\tilde{S} = S + 2\delta U_{\cdot 2} U'_{\cdot 2} = U \begin{pmatrix} \lambda_1 & 0 & \mathbf{0} \\ 0 & \lambda_2 + 2\delta & \mathbf{0} \\ 0 & 0 & \Lambda \end{pmatrix} U'.$$
- Because $\|U_{\cdot 2}\|_2 = 1$, the L_2 norm of the perturbation, $\|\tilde{S} - S\|_2$, is 2δ
- Given $\tilde{S}$ instead of S, how will λ_1 and λ_2 change to $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$?
- By construction $\tilde{\lambda}_1 = \lambda_1$
- But for the second eigenvalue,
$$\tilde{\lambda}_2 = \lambda_2 + 2\delta > \lambda_2 + \delta = \lambda_1 = \tilde{\lambda}_1.$$
- I.e., $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$ switch roles
- $\tilde{\lambda}_2$ is now the largest eigenvalue
- $\therefore \quad a = U_{\cdot 1}$ while $\tilde{a} = U_{\cdot 2}$

Effect of graph perturbation on HITS (3)

- $\tilde{a}$ is perpendicular to a, both have unit norm
- $\therefore \|a - \tilde{a}\|_2 = \|U_{\cdot 1} - U_{\cdot 2}\| = \sqrt{2}$
- HITS is completely at the mercy of eigengap $\lambda_1 - \lambda_2$, while PageRank is stabilized by parameter $1 - \alpha$
- How bad is the instability in practice?
- Can we add something like $1 - \alpha$ to HITS? Two approaches
  - PHITS
  - SALSA

HITS rank stability, adversarial



- Number of edges changed is $O(1)$
- $\Omega(n^2)$ node pairs swapped in authority order ▸ HW

Made with Xodo PDF Reader and Editor

HITS instability in practice

query = "gen. algo"

Perturbed graph 5 times

HITS – rank unstable

Full graph

rank(q)

	Title	Auth.					
1	Genetic algorithms in search optimization	Goldberg	1	3	1	1	1
2	Adaptation in natural and artificial systems	Holland	2	5	3	3	2
3	Genetic programming: On the programming of...	Koza	3	12	6	6	3
4	Analysis of the behavior of a class of genetic...	De Jong	4	52	20	23	4
5	Uniform crossover in genetic algorithms	Syswerda	5	171	119	99	5
6	Artificial intelligence through simulated...	Fogel	6	135	56	40	8
7	A survey of evolution strategies	Back+	10	179	159	100	7
8	Optimization of control parameters for genetic...	Grefenstette	8	316	141	170	6
9	The GENITOR algorithm and selection pressure	Whitley	9	257	107	72	9
10	Genetic algorithms + Data Structures = ...	Michalewicz	13	170	80	69	18
11	Genetic programming II: Automatic discovey...	Koza	7	-	-	-	10
2060	Learning internal representations by error...	Rumelhart+	-	1	2	2	-
2061	Learning to predict by the method of temporal...	Sutton	-	9	4	5	-
2063	Some studies in machine learning using checkers	Samuel	-	-	10	10	-
2065	Neuronlike elements that can solve difficult...	Barto+Sutton	-	-	8	-	-
2066	Practical issues in TD learning	Tesauro	-	-	9	9	-
2071	Pattern classification and scene analysis	Duda+Hart	-	4	7	7	-
2075	Classification and regression trees	Breiman+	-	2	5	4	-
2117	UCI repository of machine learning databases	Murphy+Aha	-	7	-	8	-
2174	Irrelevant features and the subset selection...	John+	-	8	-	-	-
2184	The CN2 induction algorithm	Clark+Niblett	-	6	-	-	-
2222	Probabilistic reasoning in intelligent systems	Pearl	-	10	-	-	-

- Each numeric column corresponds to randomly wiping out say 20% of the nodes of the reference graph
- HITS authority ranks severely affected

PageRank stability in practice

Stable because of teleport

≈ dropouts

1	Genetic Algorithms in Search, Optimization and...	Goldberg	1	1	1	1	1
2	Learning internal representations by error...	Rumelhart+	2	2	2	2	2
3	Adaptation in Natural and Artificial Systems	Holland	3	5	6	4	5
4	Classification and Regression Trees	Breiman+	4	3	5	5	4
5	Probabilistic Reasoning in Intelligent Systems	Pearl	5	6	3	6	3
6	Genetic Programming: On the Programming of...	Koza	6	4	4	3	6
7	Learning to Predict by the Methods of Temporal...	Sutton	7	7	7	7	7
8	Pattern classification and scene analysis	Duda+Hart	8	8	8	8	9
9	Maximum likelihood from incomplete data via...	Dempster+	10	9	9	11	8
10	UCI repository of machine learning databases	Murphy+Aha	9	11	10	9	10
11	Parallel Distributed Processing	Rumelhart+	-	-	-	10	-
12	Introduction to the Theory of Neural Computation	Hertz+	-	10	-	-	-

- PageRank rankings more robust to uniform random erasure of nodes
- How about choosing victims non-uniformly?

Stable HITS variant: SALSA

- PageRank distributes probabilities, whereas HITS replicates scores
- Obvious bipartite random walk on bipartite graph
  - From hub u, walk forward to one of $\mathrm{OutNbr}(u)$ auth nodes uar
  - From auth a, walk backward to one of $\mathrm{InNbr}(v)$ hub nodes uar
- "Stochastic algorithm for link structure analysis"
- For simplicity assume alternating path between any pair of hub or auth nodes
- Focus on the auth to hub to auth walk
- Transition probability from v to w is

$$p(v,w) = \frac{1}{\mathrm{InDegree}(v)} \sum_{(u,v),(u,w)\in E} \frac{1}{\mathrm{OutDegree}(u)}$$

Stable HITS variant: SALSA (2)

- Similarly a hub to auth to hub walk
- What is the steady-state node visit probability $\pi(v)$ using $p(\cdot, \cdot)$ as transition probabilities?
- Claim: $\pi(v) \propto \mathrm{InDegree}(v)$ (!)

Probabilistic HITS aka PHITS

- Make HITS closer to PageRank
- Instead of ‘copying’ random surfers, ‘distribute’ them
- Add teleport both half-steps
- Prevents bipartite cliques from trapping random surfers

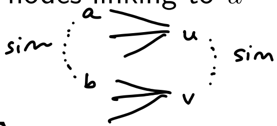
PHITS stability on node erasure

1	Learning internal representations by error...	Rumelhart+	1	3	3	2	1
2	Probabilistic Reasoning in Intelligent Systems	Pearl	4	1	1	1	2
3	Classification and Regression Trees	Breiman+	2	2	2	3	4
4	Pattern classification and scene analysis	Duda+Hart	3	4	4	4	3
5	Maximum likelihood from incomplete data via...	Dempster+	5	6	6	6	5
6	A robust layered control system for a mobile robot	Brook+	6	5	5	5	6
7	Numerical Recipes in C	Press+al	7	7	7	7	7
8	Learning to Predict by the Method of Temporal...	Sutton	8	8	8	8	8
9	STRIPS: A New Approach to ... Theorem Proving	Fikes+	9	10	10	10	15
10	Introduction To The Theory Of Neural Computation	Hertz+	11	11	9	9	9
11	Stochastic relaxation, gibbs distributions, ...	Geman+	10	9	-	-	-
12	Introduction to Algorithms	Cormen+	-	-	-	-	10

- Random jump stabilizes PHITS
- How much stability is best?

SimRank [13]

- Given directed, unweighted graph and two nodes u, v
- Node u is linked from in-neighbors $I(u)$
- How similar are u, v based only on graph neighborhood?
- Recursive definition: to the extent that nodes linking to u and v are similar:



< 1

$$\mathrm{sim}(u,u) = 1$$

$$\mathrm{sim}(u,v) = \frac{\beta}{|I(u)|\,|I(v)|} \sum_{a \in I(u), b \in I(v)} \mathrm{sim}(a,b)$$

where $\beta \in (0,1)$ is a damping constant, much like α in PageRank

- sim well-defined?
- computed efficiently?
  Monte Carlo est

Node pair graph G^2

- From $G = (V, E)$ construct G^2 as follows
- In G^2, add node (x, y) for every node pair $x, y \in V(G)$
- In G^2, node (a, b) links to node (c, d) iff $(a, c), (b, d) \in E(G)$
- Similarity propagates from node to node in G^2 much like PageRank
- Direct implementation may be impractical

Pair of random surfers

- One starting at a, the other at b
- Walking back along directed edges
- Until they meet at some node
- Expected meeting distance (EMD) $m(a, b)$: expected distance from $(a, b) \in V(G^2)$ to some singleton node $(x, x) \in V(G^2)$
- Problem: could be infinite
- Fix: expected f-meeting distance (with decay):

$$s'(a, b) = \sum_{t:(a,b) \rightsquigarrow (x,x)} \Pr(t)\beta^{\ell(t)}$$

Centerpiece subgraph

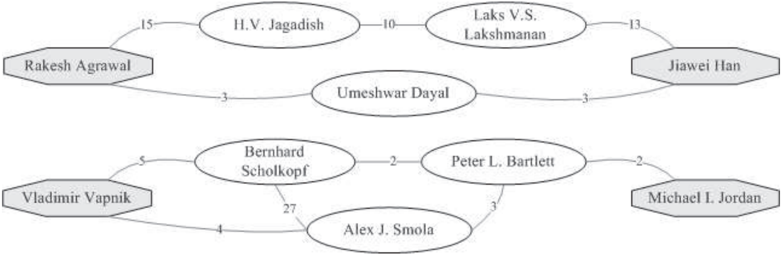
Given

- Given weighted undirected graph G
- Source node set $Q \subset V$
- Parameters k, b

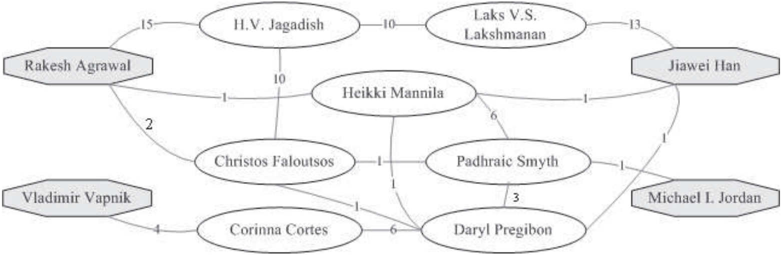
Find a suitably connected subgraph H that

- Contains all query nodes in Q
- And at most b other nodes
- Each of which has connetions to at least k query nodes
- And satisfies some “goodness” properties

Centerpiece example



(a) “K_softANDquery”: $k = 2$



(b) “AND query”

Goodness

- $r(i,j)$ is the goodness of j wrt query i
- E.g., $r(i,j) = p_{\delta_i}(j) = \mathrm{PPV}_i(j)$
- $r(Q,j)$ is the goodness of j wrt query set Q
- $r(Q,j;k)$ is the meeting probability: the steady state probability that at least k of $|Q|$ random walkers will all find themselves at node j
- $r(Q,j;|Q|) = \prod_{q\in Q} r(q,j)$
- $r(Q,j;1) = 1 - \prod_{q\in Q}(1-r(q,j))$
- In general, "K_softAND":
  $r(Q,j;k) = r(Q^{\setminus 1},k;k-1)r(Q,j) + r(Q^{\setminus 1},j;k)(1-r(Q,j))$
  where $Q^{\setminus 1}$ is the first $|Q|-1$ elems of Q
- Goodness for answer subgraph H is
  $g(H) = \sum_{j\in V(H)} r(Q,j)$
- But $\arg\max_H g(H)$ does not guarantee connectivity
- Heuristics to extract subgraph

Ranking nodes in graphs

- May not be natural or possible to represent items to be ranked using feature vectors
- Essential to model relations, graphs a natural device
- HITS and PageRank showed the way
- Like BM25, were hand-crafted
- How to incorporate machine learning into ranking in graphs?

Undirected graph Laplacian

- Simple unweighted undirected graph $G = (V, E)$ with $|V| = n$, $|E| = m$, no self-loops or parallel edges
- Node-node adjacency matrix $A \in \{0, 1\}^{n \times n}$ with $A(u, v) = 1$ if $(u, v) \in E$ and 0 otherwise
- Node-edge incidence matrix $N \in \{-1, 0, 1\}^{n \times m}$ with

$$N(v, e) = \begin{cases} -1 & \text{if } e = (v, \cdot) \\ 1 & \text{if } e = (\cdot, v) \\ 0 & \text{if } v \text{ is not either endpoint of } e \end{cases}$$

- Consider the graph Laplacian matrix $L_G = NN^\top \in \mathbb{R}^{n \times n}$
- $(NN^\top)(u, u)$ is the degree of node u
- $(NN^\top)(u, v)$ is -1 if $(u, v) \in E$, 0 otherwise
- Let D be a diagonal matrix with $D(u, u) =$ degree of u
- $NN^\top = D - A$ HW is a symmetric positive semidefinite matrix

Extending to weighted undirected graphs

- A is not boolean; $A(u,v)$ is the weight of edge (u,v) if any, 0 otherwise
- Modify N to
$$N(v,e) = \begin{cases} -\sqrt{A(e)} & \text{if } e = (v,\cdot) \\ \sqrt{A(e)} & \text{if } e = (\cdot,v) \\ 0 & \text{if } v \text{ is not either endpoint of } e \end{cases}$$
- Modify L_G to
$$L_G(u,v) = \begin{cases} \sum_w A(u,w), & u = v \\ -A(u,v), & u \neq v, (u,v) \in E \\ 0 & \text{otherwise} \end{cases}$$
- Modify "degree" matrix D to $D(u,u) = \sum_v A(u,v)$
- Still have $L_G = NN^\top = D - A$

Laplacian and node score smoothness

- For any vector $x \in \mathbb{R}^n$, ▸ HW
$$x^\top L x = \sum_{(u,v)\in E} A(u,v)\big(x_u - x_v\big)^2$$
- $x^\top L x$ penalizes node scores that are very different across "heavy" edges
- If $u \prec v$, we want $x_u + 1 \le x_v$
- Therefore define the ranking loss of score vector x as $\max\{0, 1 + x_u - x_v\}$
- The complete optimization problem is to $\min_x x^\top L x + B \sum_{u \prec v} \max\{0, 1 + x_u - x_v\}$
- B balances between roughness and data fit
- Because L is positive semidefinite, this is a convex quadratic program with linear constraints ▸ HW

Directed graph Laplacian

- Assume each row of A has at least one nonzero element
- Let $D(u,u)$ be the sum of the uth row of A
- Define Markovian transition probability matrix $Q \in [0,1]^{n \times n}$ with $Q(u,v) = \Pr(v|u) = A(u,v)/D(u,u)$
- Assume the Markov random walk is irreducible and aperiodic
- Let $\pi \in \mathbb{R}^n$ be the steady-state probability vector for the random walk, and $\Pi = \mathrm{diag}(\pi)$
- The directed graph Laplacian is defined as

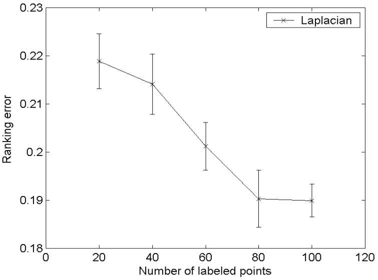
$$L = \mathbb{I} - \frac{\Pi^{1/2} Q \Pi^{-1/2} + \Pi^{-1/2} Q^* \Pi^{1/2}}{2}$$

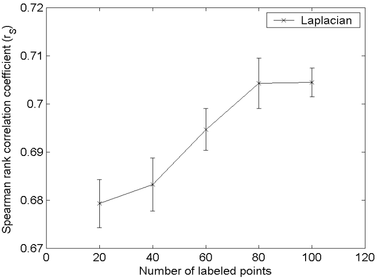
- Use in optimization in place of undirected graph Laplacian

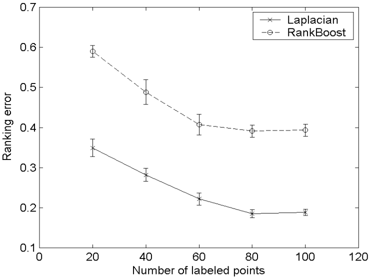
Smoothing properties

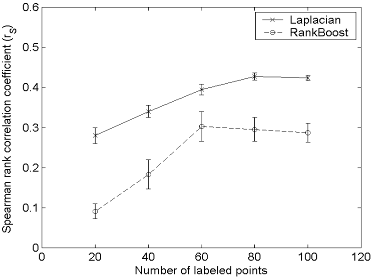
- We can show that
$$x^\top L x = \sum_{(u,v)\in E} \pi(u) Q(u,v) \left(\frac{x_u}{\sqrt{\pi(u)}} - \frac{x_v}{\sqrt{\pi(v)}} \right)^2$$
- In $\min_x x^\top L x + B \sum_{u \prec v} \max\{0, 1 + x_u - x_v\}$, suppose we set $B = 0$ (i.e., only smoothness matters)
- Clearly, $x_u \propto \sqrt{\pi(u)}$ will minimize $x^\top L x$
- I.e., in the absence of training preferences, a directed Laplacian smoother will lead to ordering nodes by decreasing Pagerank

Laplacian smoothing results









Limitations of the graph Laplacian approach

- The "link as hint of score smoothness" view is not universally applicable: millions of obscure pages u link to $v =$http://yahoo.com, with $x_u \ll x_v$
- While $\pi(u)$ is a probability, $x_u \in \mathbb{R}$ is an arbitrary score that need not satisfy Markov balance constraints (coming soon) and may even be negative
- Dual optimization involves computing the pseudoinverse L^+ of the Laplacian matrix
- Unlike L, L^+ is usually not sparse, and most packages need to hold it in RAM
- The generalization power of the learner depends on $\kappa = \max_{u \in V} L^+(u, u)$, a quantity hard to interpret

Learning r from $\prec$

- Recall $p = \alpha C p + (1-\alpha) r$, i.e., $(\mathbb{I} - \alpha C) p = (1-\alpha) r$, or $p = (1-\alpha)(\mathbb{I} - \alpha C)^{-1} r = M r$, say
- $\prec$ can be encoded as matrix $\Pi \in \{-1, 0, 1\}^{|\prec| \times |V|}$ and written as $\Pi p \geq \mathbf{0}^{|\prec| \times 1}$ (each row expresses one pair preference)
- "Parsimonious teleport" is uniform $r_0 = \mathbf{1}_{|V| \times 1} / |V|$; that gives us standard Pagerank vector $p_0 = M r_0$
- Want to deviate from p_0 as little as possible while satisfying $\prec$

$$\min_{r \in \mathbb{R}^{|V|}} (Mr - p_0)'(Mr - p_0) \quad \text{subject to}$$

$$\Pi M r \geq \mathbf{0}, \quad r \geq \mathbf{0}, \quad \mathbf{1}' r = 1$$

(quadratic objective with linear inequalities)

Pagerank as network circulation

- Can use Q and π to define a reference circulation $\{q_{uv} : (u, v) \in E\}$ as follows:
$$q_{uv} = \pi(u)Q(u, v)$$
- Idea: directly search for a circulation $\{p_{uv} : (u, v) \in E\}$
- Pagerank of node v will fall out naturally as $\sum_{(u,v)\in E} p_{uv}$

What properties must $\{p_{uv}\}$ satisfy?

- $p_{uv} \geq 0$ for all $(u, v) \in E$
- $\sum_{(u,v)\in E} p_{uv} = 1$
- Flow balance at each node v:

$$\sum_{u\in V} p_{uv} = \sum_{w\in V} p_{vw}$$

What roughness penalty should we assess?

- May want to maximize the entropy of $\{p_{uv} : (u, v) \in E\}$, i.e., $-\sum_{u,v} p_{uv} \log p_{uv}$
- May want to propose flow $\{q_{uv} : (u, v) \in E\}$ as a parsimonious belief and minimize $\mathrm{KL}(p||q) = \sum_{u,v} p_{uv} \log \frac{p_{uv}}{q_{uv}}$
- Can show that staying close to q is good for learning

Unconstrained maximum entropy flows

- Associate dual variable β_v for every flow balance constraint
$$\sum_{u \in V} p_{uv} = \sum_{w \in V} p_{vw}$$
- By dualizing the optimization, we see that HW flows have the form
$$p_{uv} \propto q_{uv} \exp(\beta_v - \beta_u)$$
- Dual objective is $\min_\beta Z$ where
$Z = \sum_{(u,v) \in E} q_{uv} \exp(\beta_v - \beta_u)$

Optimizing $\{p_{uv}\}$ with teleports

- The Markov walk specified by Q need not be irreducible and aperiodic
- As in Pagerank, we can make it so using teleports
- Walk probability $\alpha \in (0,1)$, teleport probability $1-\alpha$
- Implement teleport using transition from every v to dummy node d and back
- This leads to additional primal constraints

$$\frac{p_{vd}}{1-\alpha} = \frac{\sum_{(v,w)\in E} p_{vw}}{\alpha} \qquad \forall v \in V$$

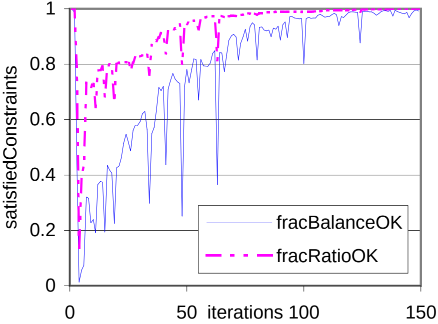
- And dual variables τ_v, leading to the solution

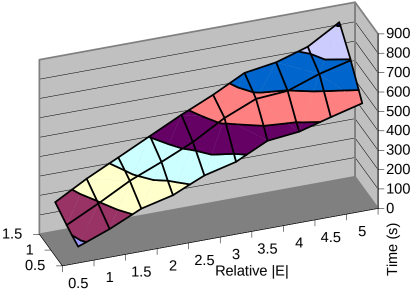
$$\begin{aligned} p_{dv} &\propto q_{dv} \exp(\beta_v - \beta_d) \\ p_{vd} &\propto q_{dv} \exp(\beta_d - \beta_v + \alpha\tau_v) \\ p_{uv} &\propto q_{uv} \exp(\beta_v - \beta_u - (1-\alpha)\tau_u) \end{aligned}$$

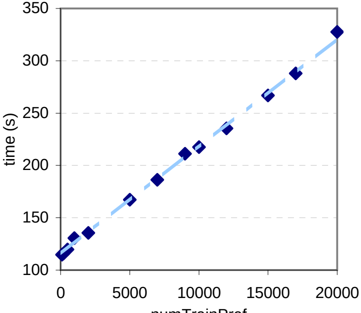
Preference constraints

- Preference $u \prec v$ leads to constraint
$$\sum_{(w,u)\in\hat{E}} p_{wu} \leq \sum_{(w,v)\in\hat{E}} p_{wv},$$
where $\hat{E} = E \cup \{(v,d) : v \in V\} \cup \{(d,v) : v \in V\}$
- Note, no margin (yet)
- Corresponding dual variables $\{\pi_{uv} : u \prec v\}$
- Define $\mathrm{bias}(v) = \sum_{r \prec v} \pi_{rv} - \sum_{v \prec s} \pi_{vs}$
- Modified solution has form
$$p_{dv} \propto q_{dv} \exp(\beta_v - \beta_d + \mathrm{bias}(v))$$
$$p_{vd} \propto q_{dv} \exp(\beta_d - \beta_v + \alpha\tau_v)$$
$$p_{uv} \propto q_{uv} \exp(\beta_v - \beta_u - (1-\alpha)\tau_u + \mathrm{bias}(v))$$

Performance of constrained circulation approach







- Must check primal constraints before terminating dual
- Scales linearly with $|V|$, $|E|$ and $|\prec|$

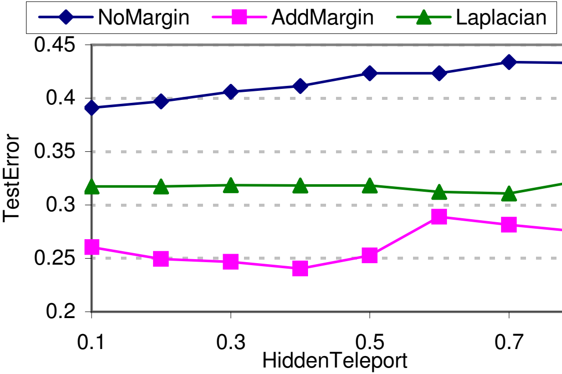
Incorporating an additive margin

- Preference constraints were expressed as $\sum_{(w,u)\in\hat{E}} p_{wu} \leq \sum_{(w,v)\in\hat{E}} p_{wv}$, not $1 + \sum_{(w,u)\in\hat{E}} p_{wu} \leq s_{uv} + \sum_{(w,v)\in\hat{E}} p_{wv}$
- $s_{uv} \geq 0$ is a primal slack variable
- Because $\sum_{u,v} p_{uv} = 1$, 1 is "too aggressive" as a margin
- ...unless we scale up $\{p_{uv}\}$
- Let q be a probability distribution and p an unnormalized distribution such that $\sum_x p(x) = F$
  - $\mathrm{KL}(p\|q) \geq 0$ if $F \geq 1$
  - For a fixed $F \geq 1$, $\arg\min_p \mathrm{KL}(p\|q) = Fq$
- New objective

$$\min_{\{p_{uv}\},\{s_{uv}\geq 0\},F\geq 1} \mathrm{KL}(p\|q) + C\sum_{u\prec v} s_{uv} + C_1 F^2$$

- New constraint $\sum_{u,v} p_{uv} = F$ replaces $\sum_{u,v} p_{uv} = 1$

Comparing Laplace vs. circulation



- In Laplace score smoothing, node scores can induce all possible permutations
- In case of network circulation, many node permutations may not be achievable for a given graph
- Smaller hypothesis space, more bias, more stable
- Seems to actually help; even better with additive margin

Typed edge conductance

- In the constrained circulation formulation, training input has very local effect owing to teleport
- Beyond a distance of about $1/(1-\alpha)$, training preferences cannot generalize
- A different, very common setting associates a type $t(u,v) \in \{1,\ldots,T\}$ with each edge (u,v)
- The weight of edge (u,v) is $\beta(t(u,v))$
- Given $\prec$ we want to estimate $\beta_1,\ldots,\beta_T$
- Assuming no dead-end nodes,

$$C(j,i) = \begin{cases} \alpha \frac{\beta(t(i,j))}{\sum_{(i,k)\in E} \beta(t(i,k))}, & i \neq d, j \neq d \\ 1-\alpha, & i \neq d, j = d \\ r_j, & i = d, j \neq d \\ 0, & i = j = d \end{cases}$$

- Here r_j is the teleport into node j, implemented using dummy node d

Constrained design of conductance

- Scaling all β by any positive factor keeps all $C(\cdot,\cdot)$ unchanged
- So we can arbitrarily scale $\beta_t \geq 1$
- C is a function of β, therefore sometimes written as $C(\beta)$
- Goal is to find $\beta \geq \vec{1}$ such that
  - $p = C(\beta)p$
  - $p_i \leq p_j$ for all $i \prec j$
- As before, we can change the constraint $p_i \leq p_j$ into a loss function $\mathrm{loss}(p_i - p_j)$
- Two problems to solve
  - Break recursion $p = C(\beta)p$ and express p directly in terms of β, so we can use a numerical optimizer
  - If there are many solutions β, which one should we prefer?

Choice of loss function

- Standard hinge $\mathrm{hinge}(y) = \max\{0, 1 + y\}$
- As before, enforcing additive margin 1 is tricky
- Scaling β has no effect on satisfying margin
- In practice, no margin or very small arbitrary margin makes no difference, both work well
- To make loss smooth and differentiable, could have picked $\mathrm{loss}(y) = \ln(1 + e^y)$
- But this does not work, experiments suggest that $\mathrm{loss}(0) = 0$ is essential
- Approximation of hinge with zero margin ($\mathrm{hinge}(y) = \max\{0, y\}$) with Huber loss:

$$\mathrm{huber}(y) = \begin{cases} 0, & y \leq 0 \\ y^2/(2W), & y \in (0, W] \\ y - W/2, & W < y \end{cases}$$

Parsimonious choice of β

- If $\beta = \vec{1}$, we get unweighted Pagerank
- Therefore the model cost can be taken as $\sum_t (\beta(t) - 1)^2$
- In fact, we get unweighted Pagerank if all $\beta(t)$ are equal, not necessarily all equal to one
- Model cost $\sum_{t,t'} (\beta(t) - \beta(t'))^2$ is another possibility
- Discourages large multiplicative factors ... ModelCost$(K\beta) = K^2$ModelCost(β)
- ... but not additive terms: ModelCost$(\beta + K\vec{1}) =$ ModelCost(β)
- In practice these work about equally well

Breaking the $p = C(\beta)p$ recursion

- Pagerank usually approximated using the Power Method $p \approx C^H p^0$ where
  - p^0 is an initial distribution over nodes, usually uniform
  - H is a suitably large horizon for convergence
- Overall optimization problem:

$$\min_{\beta \geq \vec{1}} \sum_t (\beta(t) - 1)^2 + B \sum_{i \prec j} \mathrm{huber}\big((C^H p^0)_i - (C^H p^0)_j\big)$$

- Unfortunately, not a convex optimization; need some grid plus local gradient search
- Next: computing gradient

Breaking the $p = C(\beta)p$ recursion (2)

- Compute alongsize Pagerank (using Chain Rule):

$$\forall i \forall t : \quad \frac{\partial}{\partial \beta(t)} (C^0 p^0)_i = 0$$

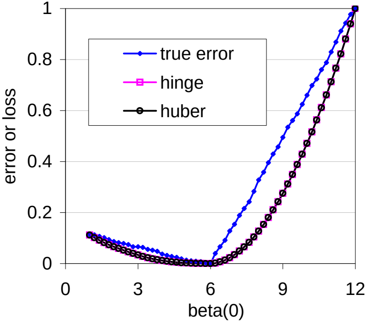
$$(C^h p^0)_i = \sum_j C(i,j)(C^{h-1} p^0)_j$$

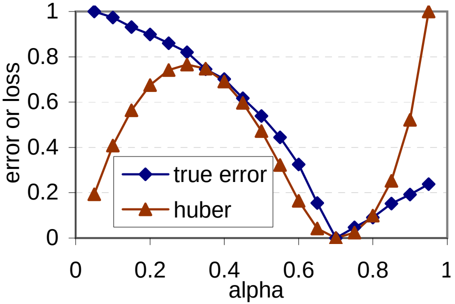
$$\frac{\partial (C^h p^0)_i}{\partial \beta(t)} = \sum_j \left[\frac{\partial C(i,j)}{\partial \beta(t)} (C^{h-1} p^0)_j + C(i,j) \frac{\partial}{\partial \beta(t)} (C^{h-1} p^0)_j \right]$$

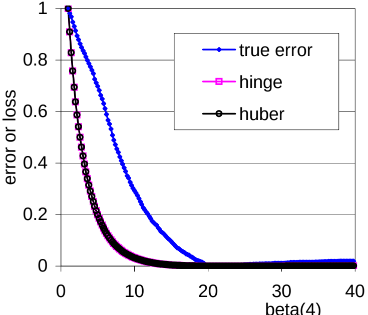
- Finally,

$$\frac{\partial C(i,j)}{\partial \beta(\tau)} = \begin{cases} -\alpha \frac{\beta(t(i,j)) \sum_w [\![\tau = t(i,w)]\!]}{(\sum_w \beta(t(i,w)))^2} & \tau \neq t(i,j) \\ \alpha \frac{\sum_w \beta(t(i,w)) - \beta(t(i,j)) \sum_w [\![\tau = t(i,w)]\!]}{(\sum_w \beta(t(i,w)))^2}, & \tau = t(i,j) \end{cases}$$

Exact loss and the approximations



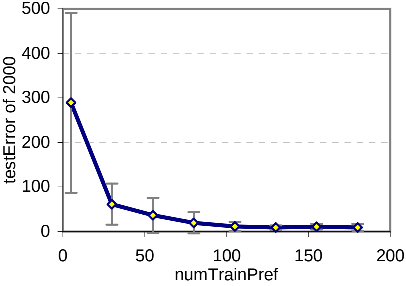


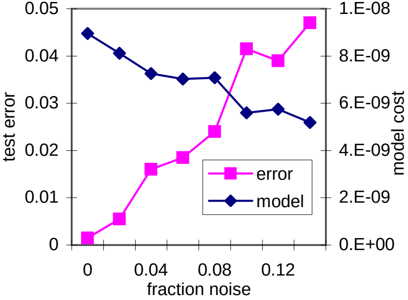


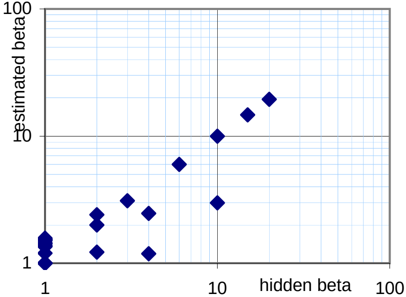
- Theoretically, the optimization surface has local minima
- Wrt β, the surface is very benign in practice
- If one also wanted to search for α, a little more care is needed

Made with Xodo PDF Reader and Editor

β estimation and learning performance







- Fast training rate
- Robust to training noise
- Reconstructs β reasonably

Link prediction: Problem setup

- Given historical snapshots of an evolving graph over some time interval through now: $G[t_0, t_1]$
- Predict edges likely to be added in the (near) future
- Often by scoring, then ranking node pairs that are not neighbors at present
- Let more time pass, observe which edges are added (like relevant docs) vs. not added (irrelevant docs): $G[t_1, t_2]$
- Success is usually measured as a ranking performance like AUC, MAP, etc.

Compromise setup

- In case graph snapshots along time are not available
- Only one snapshot $G = (V, E)$
- All edge slots E_0 ($|V|(|V|-1)$ for directed, $\binom{|V|}{2}$ for undirected)
- Remove some random edges E_t from E, leaving $E \setminus E_t$ for training
- System scores/ranks edge slots in $E_0 \setminus (E \setminus E_t)$
- Edges in E_t are "relevant", others "irrelevant"
- Now use standard ranking measures

Limitation: Test sample may not be representative of time-travel

Why is link prediction difficult?

- Social networks are sparse (constant average degree)
- When a node gets added,
  - E_0 grows by $|V|$
  - True edges grows by a constant
- DBLP in year 2000: fraction of possible citations materialized is 2×10^{-5}
- DBLP coauthorship graph from 1995 to 2004:
  - Number of authors increased from 22,000 to 286,000
  - Possible coauthorships increased $169\times$
  - Actual coauthorships incrased only $21\times$

Common predictors

Larger score $\implies$ more likely to link up

Negated shortest path length: Nodes x, y near each other more likely to link up. Frequent special case: triad completion.

Number of common neighbors: $|\Gamma(x) \cap \Gamma(y)|$

Preferential attachment: $|\Gamma(x)|\,|\Gamma(y)|$

Jaccard: $\dfrac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$

Adamic-Adar: $\displaystyle\sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log |\Gamma(z)|}$ — Celebrity common neighbors z have very little effect

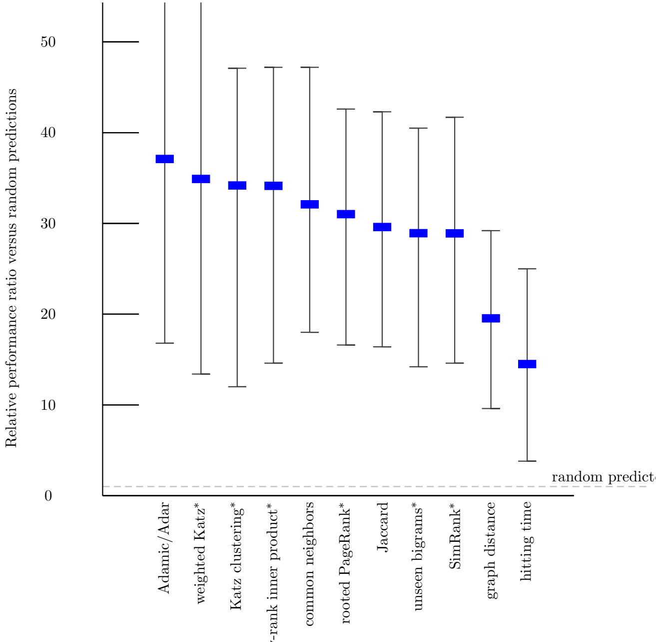
Katz: Let $P(x, y; \ell)$ be the number of paths from x to y of length exactly ℓ, and $\beta \in (0, 1)$ a tuned constant. Then the score is $\sum_{\ell \geq 1} \beta^{\ell} P(x, y; \ell)$.

Common predictors (2)

- Longer paths contribute less
- Parallel paths contribute additively
- Enumeration usually limited to small radius

Other random walk based: Hitting time, commute time, escape probability, personalized PageRank, ...

Baseline performance summary [21]



Possible enhancements

- Combine signals from many predictors
- Add local similarity or propensity to link
  - Why does (u, v_1) exist but not (u, v_2)?
- Add community/(co)cluster level link density
  - From collaborative filtering: kids tend to like adventure movies

Local (dis)similarity

References

[1] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu, “Automatic multimedia cross-modal correlation discovery,” in *SIGKDD Conference*, 2004, pp. 653–658. [Online]. Available: http://doi.acm.org/10.1145/1014052.1014135

[2] H. Tong and C. Faloutsos, “Center-piece subgraphs: Problem definition and fast solutions,” in *SIGKDD Conference*, 2006. [Online]. Available: http://www.cs.cmu.edu/~christos/PUBLICATIONS/kdd06CePS.pdf

[3] S. Brin and L. Page, “The anatomy of a large-scale hypertextual web search engine,” in *WWW Conference*, 1998. [Online]. Available: http://decweb.ethz.ch/WWW7/1921/com1921.htm

References (2)

[4] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the Web," 1998, manuscript, Stanford University.

[5] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub, "Extrapolation methods for accelerating PageRank computations," in *WWW Conference*, 2003, pp. 261–270. [Online]. Available: http://www2003.org/cdrom/papers/refereed/p270/kamvar-270-xhtml/index.html

[6] G. Jeh and J. Widom, "Scaling personalized web search," in *WWW Conference*, 2003, pp. 271–279. [Online]. Available: https://goo.gl/oAZZLq

References (3)

[7] T. H. Haveliwala, "Topic-sensitive PageRank," in *WWW Conference*, 2002, pp. 517–526. [Online]. Available: http://www2002.org/CDROM/refereed/127/index.html

[8] Z. Bar-Yossef, A. Z. Broder, R. Kumar, and A. Tomkins, "Sic Transit Gloria Telae: Towards an understanding of the Web's decay," in *WWW Conference*, 2004, pp. 328–337. [Online]. Available: http://www.www2004.org/proceedings/docs/1p328.pdf

[9] R. Baeza-Yates, P. Boldi, and C. Castillo, "Generalizing PageRank: Damping functions for link-based ranking algorithms," in *SIGIR Conference*, Seattle, Aug. 2006, pp. 308–315. [Online]. Available: http://www.dcc.uchile.cl/~ccastill/papers/baeza06_general_pagerank_damping_functions_link_ranking.pdf

References (4)

[10] A. Ng, A. Zheng, and M. Jordan, "Stable algorithms for link analysis," in *SIGIR Conference*. New Orleans: ACM, Sep. 2001, available from http://www.cs.berkeley.edu/~ang/.

[11] R. Lempel and S. Moran, "SALSA: The stochastic approach for link-structure analysis," *ACM Transactions on Information Systems (TOIS)*, vol. 19, no. 2, pp. 131–160, Apr. 2001. [Online]. Available: http://www.cs.technion.ac.il/~moran/r/PS/lm-feb01.ps

[12] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas, "Finding authorities and hubs from link structures on the world wide Web," in *WWW Conference*, Hong Kong, May 2001. [Online]. Available: http://www10.org/cdrom/papers/314

References (5)

[13] G. Jeh and J. Widom, "Simrank: a measure of structural-context similarity," in *SIGKDD Conference*. New York, NY, USA: ACM Press, 2002, pp. 538–543.

[14] S. Agarwal, "Ranking on graph data," in *ICML*, 2006, pp. 25–32. [Online]. Available: http://web.mit.edu/shivani/www/Papers/2006/icml06-graph-ranking.pdf

[15] F. Chung, "Laplacians and the Cheeger inequality for directed graphs," *Annals of Combinatorics*, vol. 9, pp. 1–19, 2005. [Online]. Available: http://www.math.ucsd.edu/~fan/wp/dichee.pdf

References (6)

[16] J. A. Tomlin, "A new paradigm for ranking pages on the world wide Web," in *WWW Conference*, 2003, pp. 350–355. [Online]. Available: http://www2003.org/cdrom/papers/refereed/p042/paper42_html/p42-tomlin.htm

[17] A. Agarwal, S. Chakrabarti, and S. Aggarwal, "Learning to rank networked entities," in *SIGKDD Conference*, 2006, pp. 14–23. [Online]. Available: http://www.cse.iitb.ac.in/~soumen/doc/netrank

[18] S. Chakrabarti and A. Agarwal, "Learning parameters in entity relationship graphs from ranking preferences," in *PKDD Conference*, ser. LNCS, vol. 4213, Berlin, 2006, pp. 91–102. [Online]. Available: http://www.cse.iitb.ac.in/~soumen/doc/netrank

References (7)

[19] Z. Nie, Y. Zhang, J.-R. Wen, and W.-Y. Ma, "Object-level ranking: Bringing order to Web objects," in *WWW Conference*, 2005, pp. 567–574. [Online]. Available: http://www2005.org/cdrom/docs/p567.pdf

[20] M. Diligenti, M. Gori, and M. Maggini, "Learning Web page scores by error back-propagation," in *IJCAI*, 2005. [Online]. Available: http://www.ijcai.org/papers/1205.pdf

[21] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1002/asi.20591

References (8)

[22] A. De, N. Ganguly, and S. Chakrabarti, "Discriminative link prediction using local links, node features and community structure," http://arxiv.org/abs/1310.4579, Oct. 2013. [Online]. Available: http://arxiv.org/abs/1310.4579