

Page Rank

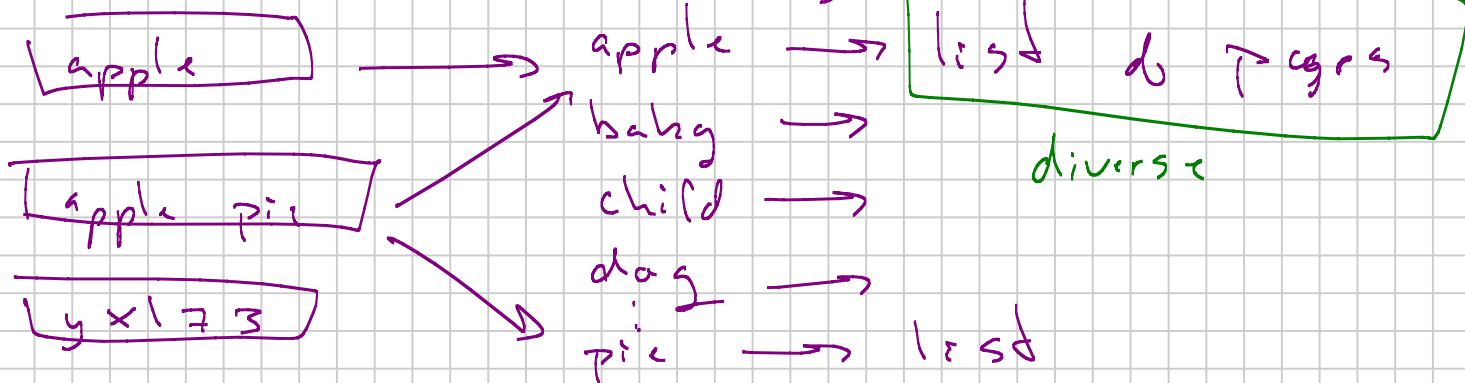
Note Title

4/11/2016

To build a robust search engine.

Efficient SE

↳ inverted index
search query



Search apple how to rank pages in importance?

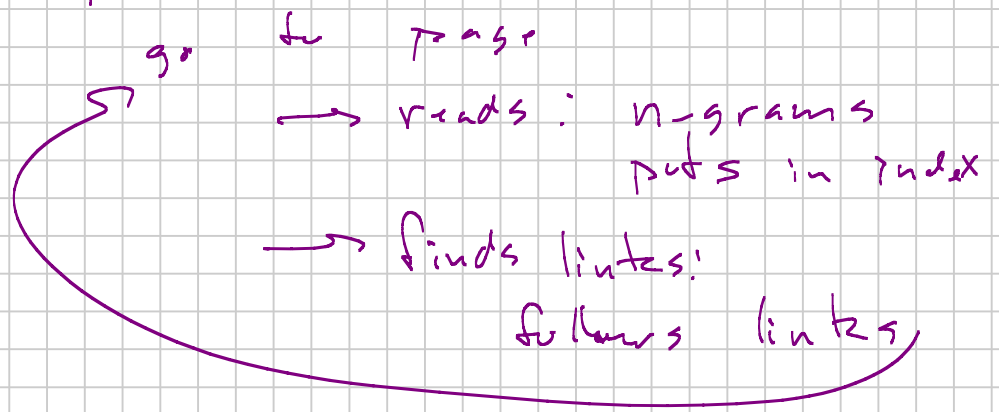
• Jaccard $J_S(\{\text{apple}\}, \{\text{words on page}\})$
→ cosine similarity

Spam pages: [apple, apple, ...]
hidden cause

• Jaccard $J_S(\{\text{words on page}\}, \{\text{apple, fruit, pie, sauce, delicious}\})$
cosine

Spam pages: copy highly ranked pages

Crawlers: program, that surf the web.



$\langle a \text{ href} = \text{"..."} \rangle \text{pic} \langle a \rangle$

Spam page: fake pages point to forged page.

Index Pages: curated list of links.

• Yahoo! / LookSmart

Page Rank

IDEA 1.

Important pages are linked to by other important pages.

IDEA 2.

Random Surfer: important if a random surfer goes there a lot

$G: (V, E)$ V : pages

E : links between pages

$$M = (V, P, q) \xRightarrow{\text{ergodic}} q^*$$

page $v \in V$

$q^*(v) = \text{page rank value}$

$$\text{Score}(v, \text{"search"}) = f\left(q^*(v), \cosine(v, \text{"search"}), \cos\left(\frac{\text{links}}{v}, \text{"search"}\right), \dots\right)$$

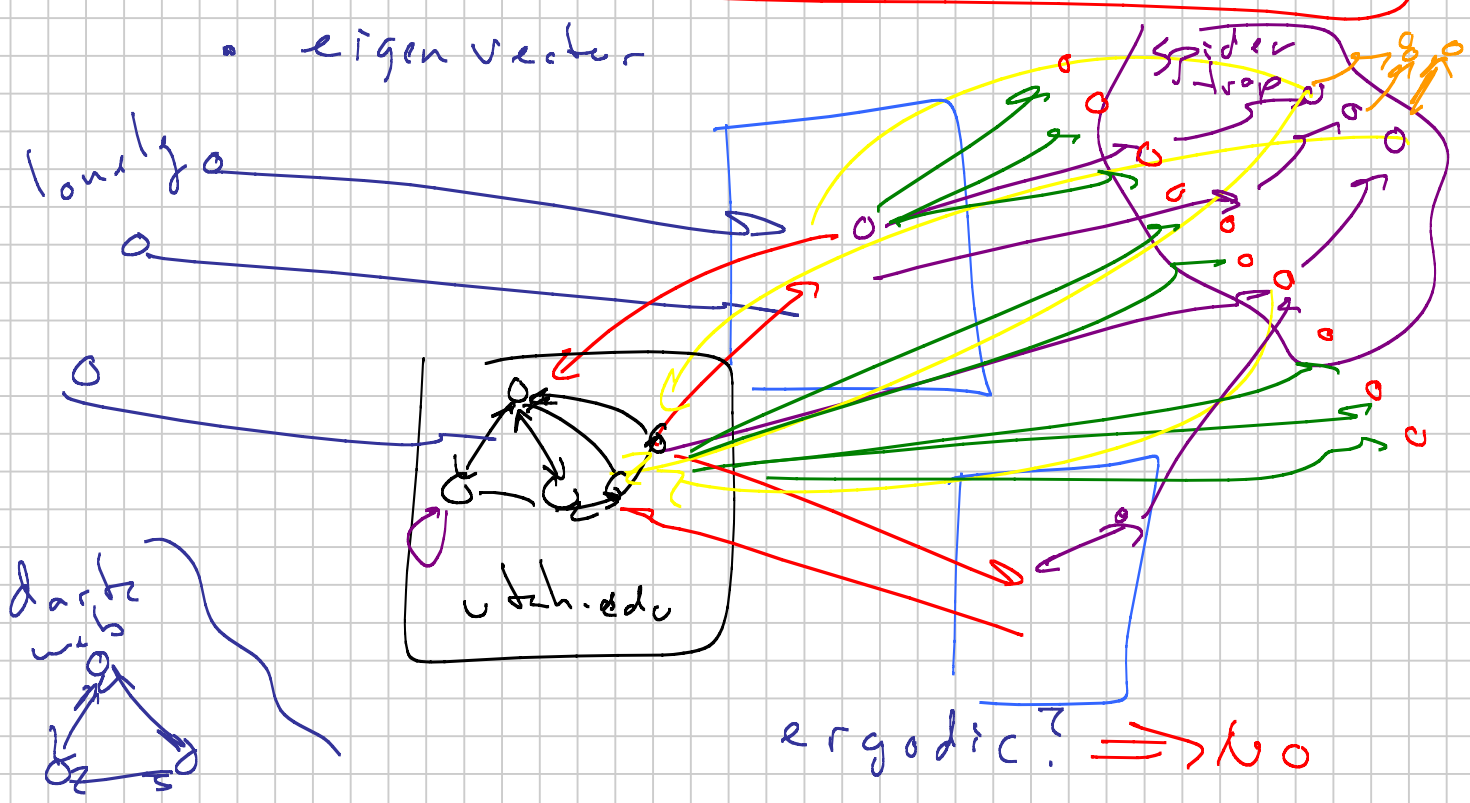
Compute q^*

- MCMC, use crawler counts.
 - ↳ slow convergence

- $P^* = P^n$ if q_0 then $q^* = P^* q_0$
 - ↳ P^* super dense

- $q_{i+1} = P q_i$ repeat n times so

- eigen vector



Teleportation

15% time \rightarrow jump to random page

$$q_{i+1} = \left(\underset{\substack{\uparrow \\ 0.15}}{(1-\beta)P} + \beta Q \right) q_i$$

$$Q = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

\rightarrow ergodic

Trust Rank

Teleportation \rightarrow only jump to trusted pages

$$p(v) \gg t(v)$$

"Trust"

Personalized Page Rank

id visits some pages regularly

\rightarrow teleport to these