How does a search engine work?

inverted index

apple | ey|

pie | er|
data | er|

important order
At least 280 Lycos Info seek

Web page $\rightarrow$ html text $\rightarrow$ bag-of-words

$\mathbf{P}_i \rightarrow \mathbf{v}_i \in \mathbb{R}^{10000}$

$v_i(0, 0, 8, 0, 0, 0, \ldots, 0, 5, 0$

Search "apple pie" $\downarrow$

$\mathbf{g} \in \mathbb{R}^{10000}$

$g_i = (0, \ldots, 0, 0, 0, 0, \ldots, 0, 1, 0, 0, 0, 0, \ldots, 0) \in \mathbb{R}^{10000}$

highest cosine-sim($\mathbf{v}_i, \mathbf{g}$) $\rightarrow$ top of list.

... best? "apple pie", "apple pie", ..." white
Battle: Search engine vs. Spammers

- Modify dist ave (cos, Iaccord)
- Emphasize certain words
- Cap word count

query: (000 capacitance)

modify

copy top pages into bottom of your page
Index

Yahoo! / Look Smart

business model: paid placement

1. Google
2. Youtube
3. Facebook
4. Baidu (China)
5. Weibo
6. Tencent QQ
7. Taobao
8. Taohu
9. Yahoo!
10. Amazon
11. Twitter
12. Instagram
13. Netflix
Crawlers
automated bot
move around the web
follow links

...but
Page Rank

Idea 1: Important webpages link to important webpages.

Idea 2: Important webpages are visited often by a random surfer.

Markov Chain

Model web as graph \( G = (V, E) \)

Vertices = Pages
Edges = Hyperlinks

"crawler"
wch graph \( G = (V, E) \rightarrow \) Adjacency matrix \\

\[ \text{Probability} \xrightarrow{\text{Transition matrix} \ P} \]

\[ r : v \rightarrow \text{ers}(P, i) = \text{PageRank vector} \]

\( r(\text{Page}) : \text{larger better} \)

\[ \text{score (Page, query)} = \text{magic (} r(\text{Page}) \cos(\text{query}) \) \]
Is the web graph ergodic?

- cycles? No
- connected? No (but ok)
- transient?

Fix: Teleportation

$\exists = 15\%$ jump to random page
Compute

\[ y^* = P^* y_0 \]

**Option 1**

\[ \text{compute } P^n \text{ for any } x \in P^n \]

"Small world" \( P^2 \) dense

**Option 2**

\[ \text{for } i = 1 \text{ to } (n=50) \]

\[ y_{i+1} = P y_i \]

Return \( y^n = y_n \)

**Option 3**

\[ y_1 = (1-B)P + BQ \]

\[ Q = \frac{1}{m} \text{max} \]

\[ = (1-B)P y_1 + P^n \]
Run 2 versions of PageRank.
- Regular: teleportation is uniform $r(p)$.
- Trust: teleportation is more likely to jump to trusted pages.

$$S(p) = \frac{r(p) - t(p)}{r(p)}$$ (larger $\rightarrow$ more likely spam)