Web Search: Crawling the Web

CPS 296.1

Topics in Database Systems

Crawling the Web

- What pages should the crawler download?
- How should the crawler refresh downloaded pages?
  - How do Web pages change?

- Cho et al. "Efficient Crawling through URL Ordering." *WWW7*, 1998
- Cho and Garcia-Molina. "The Evolution of the Web and Implications for an Incremental Crawler." *VLDB*, 2000
- Cho and Garcia-Molina. "Synchronizing a Database to Improve Freshness." *SIGMOD*, 2000

Initial crawl

- Start with an initial set of URL's, and place them in a priority queue
- Repeat until some stopping condition

  Pick a URL from the queue

  Download the page

  Extract the URL's on the downloaded page

  Place newly discovered URL's in the queue

Review of importance metrics

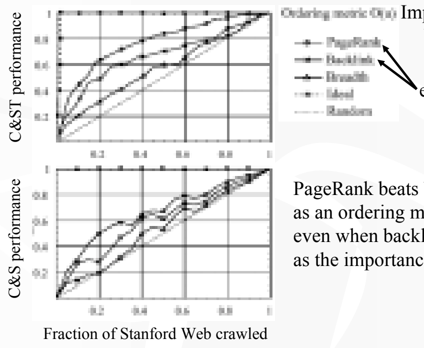
- Interest driven (useful for focused crawls)
  - Textual similarity to a driving query
  - Relevance to a topic
- Popularity driven
  - Backlink count
  - PageRank
- Location driven (based on URL)
  - Example: .com is more useful than .org
  - Example: …/home/… is more useful than …/tmp/…

➢ Combined importance value

Evaluating the crawler's performance

- Crawler crawls and stops after visiting K pages
- Crawl-and-stop model
  - A perfect crawler visits the K most important pages on the Web: performance = 100%
  - An imperfect crawler only visits M out of the K most important pages: performance = M / K
- Crawl-and-stop-with-threshold model
  - Given an importance target G, suppose there are a total of H pages with importance higher than G
  - A crawler visits M out of the H pages: performance = M / H, or 100% if $M \geq H$

Ordering URL's to improve performance



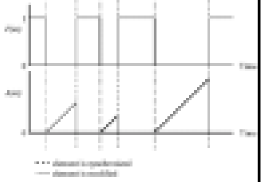
Importance metric is backlink count

estimated

PageRank beats backlink count as an ordering metric even when backlink count is used as the importance metric!

Refreshing pages

- Freshness of a page e_i at time t
  - $F(e_i; t) = 1$ if e_i is up-to-date at t
  - $F(e_i; t) = 0$ otherwise
- Age of a page e_i at time t
  - $A(e_i; t) = 0$ if e_i is up-to-date at t
  - $A(e_i; t) = (t$ – modification time of $e_i)$ otherwise
- Goal: improve freshness and/or age
  - Average over a collection
  - Average over time

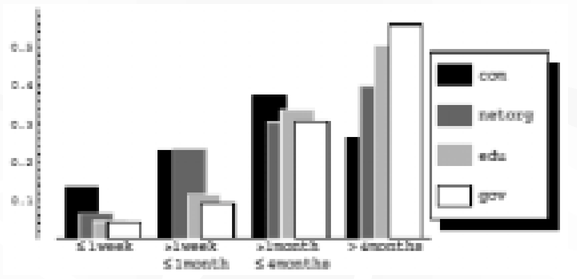


How do real Web pages change?

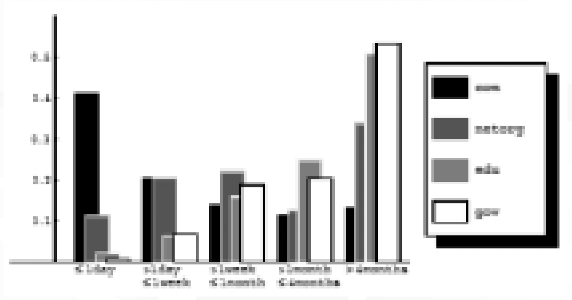
Experimental setup

- Over a four-month period
- Total of 720,000 pages from 270 sites with high PageRank's
- Everyday, revisit the root pages of these sites, and visit some number of pages that are reachable from the root pages

How long do Web pages live?

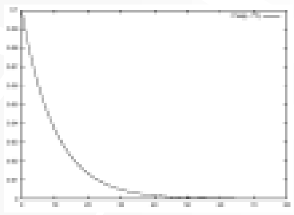


How often do Web pages change?

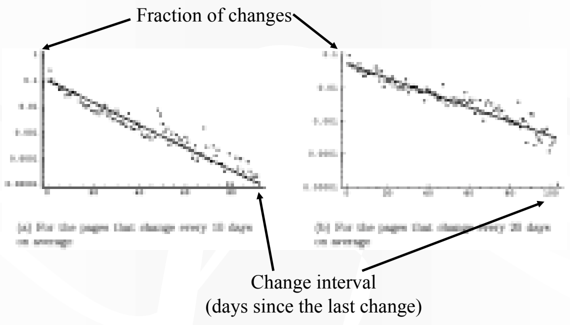


Modeling changes

- Poisson process: a sequence of random events that happen independently with fixed rate over time
- If changes to a page follow a Poisson process of rate λ, its change intervals follow the distribution $\lambda e^{-\lambda t}$
- Example: $\lambda = 0.1$ (once every 10 days on average)



Verification of Poisson process

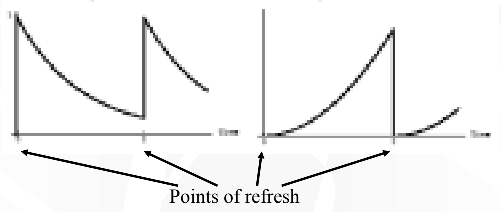


Expected freshness and age of a page

- Assuming changes follow a Poisson process

$$E[F(e;t)] = 1 - \int_0^t (\lambda e^{-\lambda s})ds = e^{-\lambda t}$$

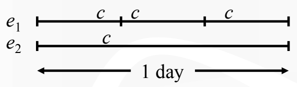
$$E[A(e;t)] = \int_0^t (t-s)(\lambda e^{-\lambda s})ds = t\left(1 - \frac{1-e^{-\lambda t}}{\lambda t}\right)$$



Refresh strategies

- Uniform: revisit all pages at the same frequency f, regardless of how often they change
- Proportional: revisit a page proportionally more often as it changes more often
- Which one is better (i.e., higher freshness and lower age over collection over time)?
  - Uniform always beats proportional for any distribution of change frequencies, as long as changes on each page follow a Poisson process

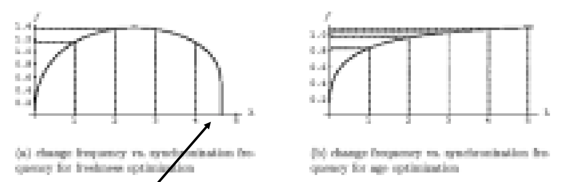
Example of uniform beating proportional



(Not Poisson)

- Say $f_1 + f_2 = 4$
  - $f_1 = 3, f_2 = 1$: expected freshness $1/2$ and $1/2$
  - $f_1 = 2, f_2 = 2$: expected freshness $3/8$ and $3/4$
  - Uniform is better!
- Note the difference from the example in paper

Optimal refresh strategy


(a) change frequency vs. synchronization frequency for freshness optimization

(b) change frequency vs. synchronization frequency for age optimization

It's not worthwhile trying to keep up with the pages that change too frequently (relative to the resources available)

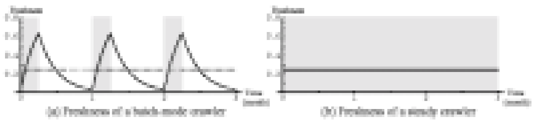
policy	Freshness	Age
Proportional	0.12	400 days
Uniform	0.57	5.6 days
Optimal	0.62	4.3 days

Freshness and age prediction based on the real web

Batch and steady crawlers

- Batch crawler: runs periodically
  - Typically uses shadowing
    - That is, newly downloaded pages are stored in a separate collection, which replaces the old collection when crawling completes
  - Typically refreshes all pages at a fixed frequency
- Steady crawler: runs continuously
  - Typically uses in-place updating
  - Can refresh pages at variable frequencies (e.g., using the optimal refresh strategy)

Batch versus steady crawlers



(a) Freshness of a batch-mode crawler

(b) Freshness of a steady crawler

- To get the same average freshness, the batch crawler places more load on network/server when it operates
- Shadowing further decreases freshness, but is easier to implement than in-place updating