The PageRank Citation Ranking: Bring Order to the Web

presented by:
Xiaoxi Pang

25.Nov 2010

Outline

- Introduction
- A ranking for every page on the Web
- Implementation
- Convergence Properties
- Personalized PageRank
- Conclusion

Introduction

- Information Retrieval encounter many new challenges in the large and heterogeneous World Wide Web.
- Traditional IR:
  importance of academic papers $\Rightarrow$ academic citation analysis.
- Web pages vs. academical publications:
  - Quality of academical publications: strictly controlled
    Web pages: free of quality control or publishing costs
  - $\hookrightarrow$ Huge number of pages can be created easily and inflating citation counts can be artificially manipulated
  - Academic papers: well defined units of work (quality & citation)
    Web pages vary on a wider scale than academic papers in quality, usage, citation and length

$\Longrightarrow$ Relative importance of web pages: **PageRank**

Link Structure of the Web

Link structure of the Web:

- pages = nodes & links = edges
- forward links = outedges
- backlinks = inedges
- A and B are Backlinks of C

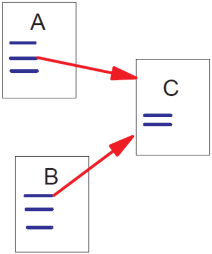


Figure: Link structure of page A, B and C

Link Structure of the Web

Standard citation analysis for PageRank:

- a link from page A to B is a vote from A to B
- highly linked pages are more "important" than pages with few links
- backlinks from "important" pages are more significant than links from average pages

$\Longrightarrow$ PageRank: how good an approximation to "importance" can be obtained just from the link structure of web

First Version PageRank: Simplified Ranking Function

Summarize: a page can get a high rank if it has high sum of the ranks of its backlinks.

First version of PageRank (simplified ranking function):

$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v}$$

- u is a web page
- F_v: set of pages v points to
- B_u: set of pages pointing to u
- $N_v = |F_v|$
- c: normalization factor

First Version PageRank: Simplified Ranking Function

Assumptions of simplified ranking function:

- Every page muss be reachable by some other page(s) in the web graph through link structure

Page ranking in the random surfer model:
more backlinks (from important pages) $\Rightarrow$ more chance to be reached by random click

First Version PageRank: Simplified Ranking Function

Another way to state simplified ranking function of PageRank:

- A: a square matrix (columns and rows corresponding web pages)
- $A_{v,u}$: link structure between page u and v
  $A_{v,u} = \frac{1}{N_u}$ if there is edge from u to v
  $A_{v,u} = 0$ otherwise
- R: vector of PageRank over all pages

$$R = cAR$$

$\hookrightarrow$ R is an eigenvector of matrix A with eigenvalue c.

Rank Sink

Simplified ranking function is constructed according to the intuitive basis: amount and importance of the backlinks
But: the web graph in real world is more complicated!
$*$ A special situation:

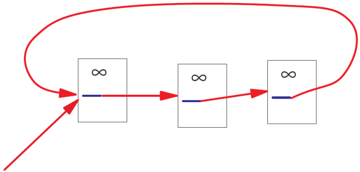


Figure: Loop which acts as a "rank sink"

- Loop only with inedge (no outedges)
- Iteration of PageRank: Loop accumulate rank (no distribution further)
- Consequence: pages in loop with high ranking (ranks of pages outside the loop are zero)

$\Rightarrow$ **rank sink**

Rank Sink

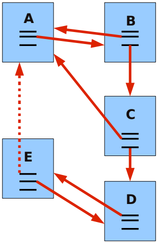


Figure: Loop accumulate rank (if without arrow from page E to A)

Loop (pages E and D) accumulate rank in calculation iteration
Consequently page E and D: PageRank 0.5 respectively
Other pages (A, B, C): zero
⇒ Distortion of the real importance ranking of pages

Developed Version PageRank

Definition

Let E be some vector over the Web Pages that corresponds to a source of rank. Then, the PageRank of a set of Web pages is an assignment, R', to the Web pages which satisfies

$$R'(u) = c \sum_{v \in B_u} \frac{R'(v)}{N_v} + cE(u)$$

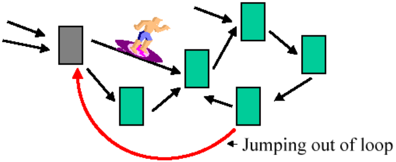
such that c is maximized and $||R'||_1 = 1$ ($||R'||_1 = 1$ denotes the L_1 norm of R').

E: Supplement the page rank for every page (so PageRanks not only rely on the link structure);
Defined uniform over all web pages (in most test) or user define

E vector in Random Surfer Model

Another intuitive basis of new Version in Random Surfer Model:
Simplified version supposes that the random surfer will keep clicking on links as long as links are valid, but in the real world:
if surfer get into a loop $\Rightarrow$ jump out to visit another random page

* Which page will be random chosen to jump to based on the distribution in E
* $||E||_1 = 0.15$



Dangling Links

Dangling links: links point to any page without outgoing links
$\hookrightarrow$ "dead ends" with no further distribution of rank

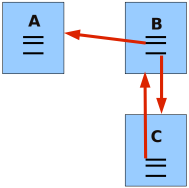


Figure: Danglink Links (page A is "dead end")

Solution:remove them from system until PageRanks for all of other pages are calculted, then put them back into system and recompute PageRanks.

Computing PageRank

Algorithm:

$R_0 \leftarrow S$ *# initialize vector over web pages*

loop:

$R_{i+1} \leftarrow AR_i$ *# new ranks sum of normalized backlink ranks*

$d \leftarrow ||R_i||_1 - ||R_{i+1}||_1$ *# compute normalizing factor*

$R_{i+1} \leftarrow R_{i+1} + dE$ *# add escape term*

$\delta \leftarrow ||R_{i+1} - R_i||$ *# control parameter*

while $\delta > \varepsilon$ *# stop when converged*

Implementation

Computing resources: database with 24 million web pages references 75 million unique URLs

Memory and disk:

* Weight vector: each URL 4 byte float (75 million URLs = 300MB)

* Weights from current time step stored in Memory, previous Weights accessed linearly on disk

* Link database Matrix A also in Disk (linear access)

Implementation:

1. Unique integer ID for each URL
2. Sort and Remove dangling links
3. Rank initial assignment
4. Iteration until convergence
5. Add back dangling links and Re-compute
6. Took ca.5 hours

Convergence Properties

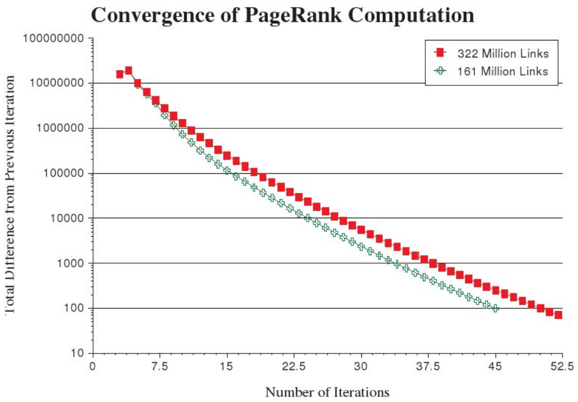


Figure: Rates of Convergence for Full Size and Half Size Link Databases; Scaling factor: linear in log n

Convergence Properties

- PageRanks computation in logarithmic time
- Web Graph is rapidly mixing: an expander
- Properties of expander graphs can be utilized in future computation involving the Web graph

Personalized PageRank

E **Vector:**

- Rank source to make up for the rank sinks
- The distribution of web pages that a random surfer periodically jumps to
- Customize page ranks

Test: the total weight E on a single web page
$*$ Two home pages: Netscape & John McCathy
$*$ Result: two home pages get highest PageRank respectively, higher PageRanks belong to its immediate links
– e.g pages related to computer science at Stanford University get much higher PageRank in John McCarthy-rank than Netscape-rank

$\Longrightarrow$ Setting vector E can generate personalized PageRanks

Conclusion

PageRank:

- Based just on Web's graph structure
- Citation analysis: backlinks (amount and importance)
- Developed PageRanks: rank source for a good approximation of "importance" ranking
- High quality search results

Thank you for your attention!