Introduction to Data Mining

Lecture9 Pagerank

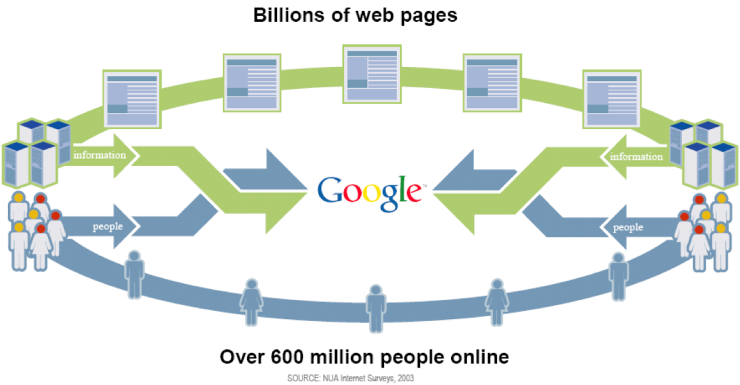
Jun Huang

Anhui University of Technology

Spring 2018

huangjun_cs@163.com

Motivation


Billions of web pages
information
information
Google
people
people
Over 600 million people online
SOURCE: NUA Internet Surveys, 2003

Why ranking pages?

- **There are thousands of Michael Jacksons in the world**
- Michael Johnson $\cdots$. Michael Smith ... Just Michael .... Even thousands of Michael Jackson
- Taxidriver Michael Jackson, Teacher Michael Jackson ..... Why does the popsinger only make it to the top results
- Because of the algorithm google uses to rank its results – Page Rank

Founders

- **Larry Page**
  - B.S.E in C.E. (U of Michigan)
  - Ph.D. candidate in C.S. at Stanford
- **Sergey Brin**
  - B.S. in Math's and CS (1993) –U of Maryland
  - M.S. in CS (1995) –Stanford University
  - Recipient of NSF Grad Fellowship
  - Ph. D candidate in C.S. at Stanford

Abstract

- To prototype a large scale search engine
- Use existing structure of hypertext
- Crawl and index web efficient
- Produce better search results

Introduction

- Amount of info in web is increasing
- New users inexperienced with the "art" of web research is increasing
- Before google search engines were
- Human Maintained
  - Only cover popular topics effectively
  - Subjective and biased
  - Expensive to build and maintain
  - Cannot cover esoteric topics
- Automated:
  - Keyword matching only
  - Low quality search results
  - Easy to mislead

Challenges

- Indexing millions of web pages
- Answering millions of queries
- Very little academic research done in the field
- Advances in technology
- Insufficiency of existing search techniques
- Exploiting already present information
- Internet is like the wild wild west, anyone can publish anything

What is Google?

- Started out as a search engine!
- Common word for googol –1 followed by 100 zeros
- Google.com registered in Sep 15, 1997
- Mission –to organize seemingly infinite information in the web

Other Search Engines

- Archie (1990) was the first search engine
- Vlib (92), WWWW (1993), Altavista(1994) , Webcrawler(1994), Yahoo (1994)
- Altavista = Purchased by yahoo in 2003, shut it down in 2013
- Vlib = Virtual Link Library setup by Tim Berners Lee
- Microsoft = MSN (1998), Live Search (2007), Bing (2009)
- Even previous google employees started a search engine in 2008 called Cuil
- Baidu (2000)
- Sogo (2004)
- 360 (2012)

Google's scaling problems analysis

- Fast crawling technology to get and update indexes
- Storage space
- Efficient data processing
- Effective query handling
- Using better Data Structures
- Role of hardware :
  - Good –Performance improving, price decreasing
  - Bad –Disk time, O.S. Robustness will stay bottleneck factors

Design goals I

- **Improving Search quality**
  - 1994 –a complete search index will be enough
  - 1997 –washing junk results. (Only 1 of 4 search engines came up when searched in top 10 results)
  - Number of documents is increasing but peoples are not interested on all
  - Maintaining precision with increasing size of web
- Plan: Use Link Structure and Anchor text to analyze importance of pages

Design goals II

- **Academic Search Engine Research**
  - Web is becoming increasingly commercial
  - 1.3 % .com domains in 1993 => 60% .com domains in 1997
  - Search engines are also commercial, no publications in them
  - Search engines becoming very closed and advertisement oriented
- plan: push development and understanding to academic world

Design goals III

- **To Make Search engines USABLE for common people**
- **To build architecture that supports research on large scale web data**
    - Set up environment to let researchers experiment on large chunks of Text

Intuitions I

- **The number of links pointing to a page matters**
  - If a random walker is at page X which has a link to page P, it might end up going to page P since there is a path between P and X
  - So, more websites pointing to a site increases the possibility of a random walker reaching that page

Intuitions II

- **If a page X has two outgoing links, then the chance of a random walker reaching P is reduced by half**
  - If the number of outgoing pages is high, the probability of reaching them is also reduced

Intuitions III

- **Page P is likely reached if the page rank of a page pointing to it is also high**

Intuitions IV

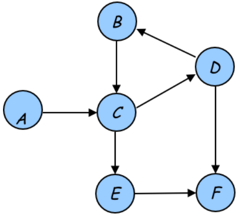
- **A random walker can jump to any page not in the graph as well**
  - You are using twitter for a while now, you suddenly decide to check your email instead
  - This "jumping" behavior of a random walker is modeled by a dumping factor

Page rank

- Uses the link structure of web
- Access a page' s value
- If A has a link to B, its like A voted for B

Background

- Directed Graphs
  - Set of nodes with directed edges
  - In our case, nodes = websites
  - Edges = links
  - Each website has in-coming links and outgoing links



- Hypertexts –interactive text with destination address to redirect user

Basic Concepts

- **What is PageRank?**
  - A method for rating the importance of web pages objectively and mechanically using the link structure of the web
- **In-links** of page i: I_i
  - The number of pages from other websites that point to page i
- **Out-links** of page i: O_i
  - The numer of pages from other websites that page i point to
- **Rank prestige**
  - The larger the in-links I_i, the bigger the rank of page i
  - The bigger score of the in-links pages, the bigger the rank of page i

Score of Page Rank

- Define a directed graph $G = (V, E)$ to indicates the relationships between webpages
- V is the set of webpages, and $|V| = n$
- E is the set of directed edges between different webpages
- $p(i)$ is the score of pagerank for the i-th page

$$p(i) = \sum_{(j,i)\in E} \frac{p(j)}{O_j}$$

- Define n-dimension vector $\mathbf{p}$

$$\mathbf{p} = (p(1), p(2), ..., p(n))^T$$

Score of Page Rank

- Define a adjacent matrix $\mathbf{A}$ of graph $G = (V, E)$

$$A_{ij} = \begin{cases} \frac{1}{O_i}, & \text{if } (i, j) \in E \\ 0, & \text{otherwise} \end{cases}$$

- Then, we can obtain:

$$\mathbf{p} = \mathbf{A}^T\mathbf{p}$$

- Thus, the rank vector $\mathbf{p}$ is an eigenvector of the stochastic web matrix $\mathbf{A}$
- In fact, its first or principal eigenvector, with corresponding eigenvalue 1

Constraint on $\mathbf{A}$

- $\mathbf{p}$ can be derived by markov theory, while $\mathbf{A}$ should satisfy three conditions:
  - **stochastic**: each element $A_{ij} \geq 0$, and $\sum_i A_{ij} = 1$. Thus, each web page must have one out-link
  - **strongly-connected graph**: $\forall u, v \in V$, thers exist one path from u to v
  - **nonperiodic**:
    - A graph is peridodic, for any state i is peridodic with period $k > 1$, that the length of path that starting from state i and back to it is integral multiple of k

Solution

- **stochastic**: each element $A_{ij} \geq 0$, and $\sum_i A_{ij} = 1$. Thus, each web page must have one out-link
- **Solution**:
- add links to all the other webpages for the pages without any out-links
- set the transition probabilities to other pages to $1/n$

Solution (cont.)

- **strongly-connected graph**: $\forall u, v \in V$, thers exist one path from u to v
- **nonperiodic**:
  - A graph is peridodic, for any state i is peridodic with period $k > 1$, that the length of path that starting from state i and back to it is integral multiple of k
  - A graph is nonperiodic when $k = 1$
- **Solution**:
- add links to all the other webpages for every page
- set parameter d to control the transition probabilities to other pages
  - $d \in [0, 1]$ is the damping factor

Score of PageRank

- After the operations, the adjacent matrix $\mathbf{A}$ satisfies the three constraints, i.e., **stochastic**, **strongly-connected graph**, **nonperiodic**
- The improved model pagerank is:

$$\mathbf{p} = (1 - d)\mathbf{e} + d\mathbf{A}^T\mathbf{p}$$

- For the i-th page, its pagerank

$$p(i) = (1 - d) + d\sum_{j=1}^{n} A_{ji}p(j)$$

- $d \in [0, 1]$ is the damping factor, d is set to be 0.85 in the original paper

Pagerank Iteration

Power iteration method

PageRank-Iterate (G)

$\mathbf{p}_0 \leftarrow \frac{\mathbf{e}}{n}$

$k \leftarrow 1$

repeat

$\mathbf{p} = (1-d)\mathbf{e} + d\mathbf{A}^T\mathbf{p}$

$k \leftarrow k+1$

untill $\|\mathbf{p}_k - \mathbf{p}_{k-1}\|_1 < \epsilon$

return $\mathbf{p}_k$