COMP348 — Document Processing and the Semantic Web

Week 02 Lecture 2: Web Search

Diego Mollá

COMP348 2019H1

Abstract

In this lecture we will briefly look at technology related to web search. We will pay special attention to Google's PageRank algorithm. This was the initial algorithm that Google used to rank the results of the Google searches.

Update February 22, 2019

Contents

Reading

- Lecture notes
- Tanase & Radu's Lecture on PageRank algorithm:
  http://www.math.cornell.edu/~mec/ Winter2009/RalucaRemus/Lecture3/lecture3.html

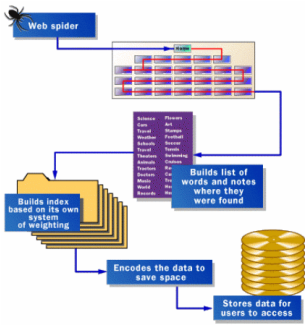
Additional Resources

- Brin and Page (1998): http://infolab.stanford.edu/~backrub/google.html — a seminal paper by the founders of Google.

1 Crawling and Indexing the Web

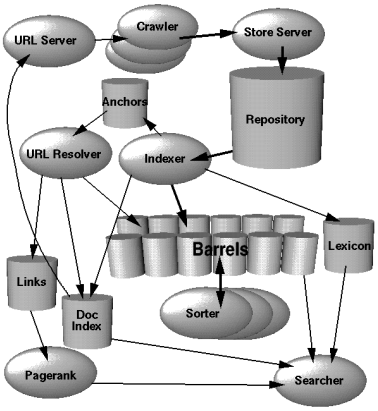
Crawling the Web

- Web search engines keep an off-line snapshot of the Web.
- This snapshop is created and maintained by crawlers.
- Crawlers (spiders, ants, ...) are programs that fetch Web pages.



Scaling to the Web

- Crawling and indexing a collection of documents on itself is not difficult.
- The challenge is to scale to the entire Web.
- Main Web search engines employ massive parallel processes for distributed crawling and indexing.



2 Ranking the Search Results

Ranking Search Results

- Google's original search method was a simple Boolean search: "find all documents that contain all the query terms".
- A key innovation from Google was how to rank the results.
  - Another key innovation was how to scale up to the entire Web.
  - (Google's real success was how to monetise Web search but we won't go into that.)
- The PageRank algorithm determines the importance of a page regardless of the query.
- The importance of a page is determined by how well linked it is.

PageRank

- The Web can be seen as a graph.
- The nodes are the HTML pages, the edges are the hyperlinks.
- PageRank can be defined recursively.

Formula of PageRank

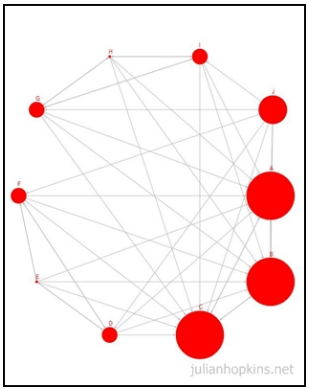
$$PR(A) = \frac{1-d}{N} + d(\frac{PR(T_1)}{C(T_1)} + \cdots + \frac{PR(T_n)}{C(T_n)})$$

N = total number of documents;
T_i = page that links to A;
$C(T_i)$ = outgoing links from page T_i.

PageRank and Random Walks

- The PageRank of a page is the probability of arriving at that page after a long sequence of random clicks.
- This "random surfer" will follow a link from a page with probability d: the "damping factor" (usually $d = 0.85$).

The formula is a variant of the eigenvector centrality measure used in network analysis.



Computing PageRank

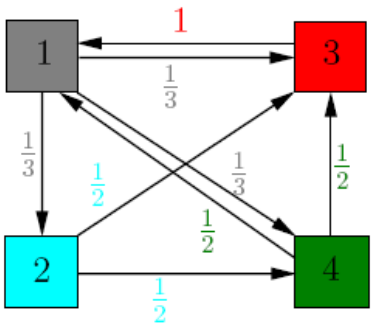
- PageRank can be computed iteratively.
- At first iteration, we assume all nodes have same weight $1/N$.
- We then apply the formula to spread the weights to the neighbours.
- Given that the inherent graph is very sparse, only a few iterations are needed to converge.

The Mathematics of Google Search
http://www.math.cornell.edu/~mec/ Winter2009/RalucaRemus/Lecture3/lecture3.html

The Transition Matrix of a Graph

We can express a graph as a matrix.

- Columns, rows: nodes.
- Cell(j,i): weight of the edge from node i to node j.



$$A = \begin{pmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{pmatrix}$$

$A(j,i) = \frac{1}{C(i)}$ if there's a link from i to j.

Note that, by the nature of the transition matrix, the sum of all elements in every column is 1.

Adding the Damping Factor

Adjusted Transition Matrix

$$M = d \cdot (A) + (1-d) \cdot B$$

$$B = \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

$(1-d) \cdot B$ corresponds to the term $\frac{1-d}{N}$ in the PageRank formula:

$$PR(A) = \frac{1-d}{N} + d(\frac{PR(T_1)}{C(T_1)} + \cdots + \frac{PR(T_n)}{C(T_n)})$$

Spreading the Weights

Using the transition matrix and an initial set of weights, we only need to iteratively multiply the matrix with the weights.

$$PR = M \cdot PR$$

Proof

$$\begin{aligned} PR(i) &= M(i,1)PR(1) + M(i,2)PR(2) + \cdots \\ &= (\tfrac{1-d}{N} + d \cdot A(i,1))PR(1) + \\ &\quad (\tfrac{1-d}{N} + d \cdot A(i,2))PR(2) + \cdots \\ &= \tfrac{1-d}{N} + d(\tfrac{PR(1)}{C(1)} + \tfrac{PR(2)}{C(2)} + \cdots) \end{aligned}$$

(note that $PR(1) + PR(2) + \cdots = 1$)

A more detailed proof follows. The proof will assume that the network has three nodes only but it can be generalised to any network size.

Let $PR(i)$ refer to the element i of the vector PR. $PR(i)$ represents the PageRank value of node i.

Let $M(i,j)$ refer to row i and column j of the matrix M. This means that $M(i,j) = \frac{1-d}{N} + d \cdot A(i,j)$, where $A(i,j)$ has the value $\frac{1}{C(j)}$ if there is a link from node j to node i.

Thus, expanding $PR = M \cdot PR$ we have the equation:

$$\begin{pmatrix} PR(1) \\ PR(2) \\ PR(3) \end{pmatrix} = \begin{pmatrix} M(1,1) & M(1,2) & M(1,3) \\ M(2,1) & M(2,2) & M(2,3) \\ M(3,1) & M(3,2) & M(3,3) \end{pmatrix} \cdot \begin{pmatrix} PR(1) \\ PR(2) \\ PR(3) \end{pmatrix}$$

Then:

$$\begin{aligned} PR(i) &= M(i,1)PR(1) + M(i,2)PR(2) + M(i,3)PR(3) \\ &= (\tfrac{1-d}{N} + d \cdot A(i,1))PR(1) + (\tfrac{1-d}{N} + d \cdot A(i,2))PR(2) + (\tfrac{1-d}{N} + d \cdot A(i,3))PR(3) \\ &= \tfrac{1-d}{N}(PR(1) + PR(2) + PR(3)) + d(A(i,1)PR(1) + A(i,2)PR(2) + A(i,3)PR(3)) \end{aligned}$$

Since $PR(1) + PR(2) + PR(3) = 1$, then:

$$PR(i) = \tfrac{1-d}{N} + d(A(i,1)PR(1) + A(i,2)PR(2) + A(i,3)PR(3))$$

Since $A(i,j)$ has the value $\frac{1}{C(j)}$ if there is a link from node j to node i, we obtain the original formula of PageRank:

$$PR(A) = \frac{1-d}{N} + d(\frac{PR(T_1)}{C(T_1)} + \cdots + \frac{PR(T_n)}{C(T_n)})$$

An Iteration in Python

```
>>> import numpy as np
>>> A = np.array([[0.,     0.,     1., 1./2.],
                  [1./3., 0.,     0., 0.    ],
                  [1./3., 1./2., 0., 1./2.],
                  [1./3., 1./2., 0., 0.    ]])
>>> M = 0.85*A + 0.15*(1./4.*np.ones((4,4))
>>> PR = 1./4.*np.ones((4,1))
>>> PR = np.dot(M,PR)
>>> PR
array([[ 0.35625    ],
       [ 0.10833333],
       [ 0.32083333],
       [ 0.21458333]])
```

Iteration Until Convergence in Python

Code

```
epsylon = 0.01
iterations = 0
PR = 1./4.*np.ones((4,1))
oldPR = np.zeros((4,1))
```

```
while max(np.abs(oldPR-PR)) > epsylon:
    oldPR = PR
    PR = np.dot(M,PR)
    iterations +=1
print "PR after",iterations,"iterations:"
print PR
```

Output

```
PR after 5 iterations:
[[ 0.36966846]
 [ 0.14289417]
 [ 0.28643227]
 [ 0.2010051 ]]
```

3 Search Engine Optimisation

Search Engine Optimisation

Search Engine Optimisation
Try to ensure that our favoured URL is ranked top against relevant searches.

- Many businesses aim at getting the top hit in relevant searches.
- They try to reverse-engineer the indexing and ranking methods of search engines.
- Search engines incorporate secret algorithms to prevent spamming search results.

Indexing

What information is indexed?

1. Most relevant words.
    - e.g. those with high tf.idf.
    - Depends on the internals of the search engine.
2. Words from the title.
3. Words from headings may be given more importance.

Spamming Techniques 

1. Add words to the "keywords" and "description" meta tags.
2. Add words in hidden text of same colour as the background.

Ranking

How to improve ranking?

1. Ensure that your page is linked by others.
2. Ensure your page is listed by authoritative pages.

Link Farms 

- A common technique to artificially increase the rank of your page is to exchange links with others.
- Link farms are clusters of heavily linked webpages.
- Search engines use algorithms to detect link farms.

General Tips for Improving Visibility of your Page

From Google's Search Engine Optimization Starter Guide (2019):
https://support.google.com/webmasters/answer/7451184

1. Tell which pages shouldn't be crawled (e.g. using robots.txt).
2. Create unique, accurate page titles.
3. Use the "description" meta tag.
4. Use heading tags to emphasize important text.
5. Add structured data markup.
    - We'll cover some of this in this unit.
6. Organize your site hierarchy.
7. Create a simple navigational page for users.
8. Simple URLs convey content information.
9. Make your site interesting and useful.
10. Know what your readers want (and give it to them).
11. Write good link text.
12. Be careful who you link to.
13. Combat comment spam with "nofollow".
14. Use the "alt" attribute in your images.
15. Make your site mobile-friendly.
16. Promote your website.
17. Analyze your search performance and user behaviour.

Take-home Messages

1. What is crawling? indexing? ranking?
2. What is the PageRank formula?
3. Implement PageRank in Python.
4. What are the general methods of Search Engine Optimisation?

What's Next

Week 3

- Introduction to Statistical Classification.

Reading

- NLTK Chapter 6.
- Manning et al. Chapter 14.