Internet Search Engine

Allan F. Emmanuel
Mentor: Prof. Ariane Masuda

Department of Mathematics
New York City College of Technology

Abstract

Search engines are designed to carry out web searches from the World Wide Web (WWW). The first internet search engine software was created in 1990. Today some of the most popular ones are Google search, Yahoo! Search, and Bing. They are some of the most visited websites on the internet and are useful at recommending links based on some of the keywords that a user types. This project will focus on using Linear Algebra to rank websites from the most visited ones to the least visited ones.

WebGraph

Alice designs a prototype that helps her understand how six pages are connected to each other. She uses this network of links to rank the pages from the most important ones to the least important ones. Her idea is to create a graph, where the vertices represent the pages and the directed edges represent the existence of links connecting the pages. This graph is called a **WebGraph**. She also considers the following assumptions.

- The six pages are not linked to or from any outside page.
- There are no links from a page to itself, i.e. there are no loops.
- There are no duplicate links, i.e. there are no multiple edges.
- There are no pages with no outgoing links; in particular, there are no isolated points.

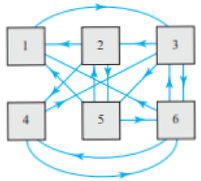

Figure 1. Alice's WebGraph (picture obtained from [1]).

Web surfing

Alice's strategy can be described in the following way. Suppose that she clicks on Page 3. Then she clicks on one of its links to either Page 2, Page 5 or Page 6. She counts the number of times she hits each page after 10, 100, 1000, 10000 and 20000 mouse clicks. She also computes each proportion, using four decimal places.

	Number of Mouse Clicks						Fraction of Mouse Clicks					
Page	0	10	100	1000	10000	20000	0	10	100	1000	10000	20000
1	0	2	19	181	1671	3192	0	0.1818	0.1881	0.1808	0.1671	0.1596
2	0	1	13	153	1340	2785	0	0.0909	0.1287	0.1528	0.134	0.1392
3	1	3	11	168	2426	4957	1.0	0.2727	0.1089	0.1678	0.2426	0.2478
4	0	0	20	187	1428	2382	0	0	0.1980	0.1868	0.1428	0.1191
5	0	3	14	99	1294	2549	0	0.2727	0.1386	0.0989	0.1294	0.1274
6	0	2	24	213	1842	4136	0	0.1818	0.2376	0.2128	0.1842	0.2068

Table 1. Numbers and Fractions of Visits.

The fractions in Table 1 stabilize and converge to a limiting value that depends on the structure of the graph. These values are used to measure the importance of a page. Based on 20000 mouse clicks, Table 1 reveals that Page 3 is the most important one followed by Pages 6, 1, 2, 5 and 4.

Markov Matrix Approach

We describe a procedure based on Markov chains by first introducing some terminologies.

Definition 1

The **adjacency matrix** of a WebGraph with n pages is an $n \times n$ matrix A such that a_{ij} is 1 if the j^{th} page has an outgoing link to the i^{th} page and 0, otherwise.

Definition 2

Suppose that a WebGraph has n pages. The **state vector** $x^{(k)}$ is the $n \times 1$ vector whose i^{th} entry is the probability that the surfer is on the i^{th} page after k random mouse clicks.

The adjacency matrix corresponding to the WebGraph in Figure 1 and the state vector $x^{(0)}$ corresponding to Table 1 are

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \quad \text{and} \quad x^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

- The sum of the entries of the i^{th} row of an adjacency matrix is the number of incoming links to the i^{th} page from the other pages.
- The sum of the entries in the j^{th} column is the number of outgoing links on the j^{th} page to other pages.
- In our case, the third entry of $x^{(0)}$ is 1. This means that the product $Ax^{(0)}$ is the third column vector of A.
- In general, if one is on the j^{th} page after k mouse clicks, then the j^{th} entry of $x^{(k)}$ is 1 and all other entries are 0. In addition, the product $Ax^{(k)}$ is the j^{th} column of A.

Definition 3

The **probability transition matrix** $B = [b_{ij}]$ associated with an adjacency matrix $A = [A_{ij}]$ is obtained by dividing each entry of A by the sum of the entries in the same column:

$$b_{ij} = \frac{a_{ij}}{\sum_{k=1}^{n} a_{kj}}.$$

A Computational Experiment

- We have $B = \begin{bmatrix} 0 & 1/3 & 0 & 0 & 1/3 & 1/3 \\ 0 & 0 & 1/3 & 0 & 1/3 & 0 \\ 1 & 0 & 0 & 1/2 & 0 & 1/3 \\ 0 & 1/3 & 0 & 0 & 0 & 1/3 \\ 0 & 1/3 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 1/2 & 1/3 & 0 \end{bmatrix}$ and $x^{(k)} = Bx^{(k-1)}$ for $k = 1, 2, 3, \ldots$
- The formula for $x^{(k)}$ is recursive. We experimented computing $x^{(20)}$ using $x^{(0)} = e_i$, the i^{th} standard basis vector, to see if the results would change.

$x^{(0)}$	$x^{(20)}$
e_1	$[0.15457815269\ 0.13637223372\ 0.27266380458\ 0.10911245162\ 0.13637223372\ 0.19090112366]^T$
e_2	$[0.15453467554\ 0.13636777477\ 0.27273111553\ 0.10908447084\ 0.13636777486\ 0.19091418847]^T$
e_3	$[0.15454853037\ 0.13634534610\ 0.27276808639\ 0.10909111913\ 0.13634534610\ 0.19090157191]^T$
e_4	$[0.15455086180\ 0.13636089722\ 0.27272696935\ 0.10909407584\ 0.13636089722\ 0.19090629857]^T$
e_5	$[0.15452072360\ 0.13636575945\ 0.27275414742\ 0.10907543578\ 0.13636575935\ 0.19091817440]^T$
e_6	$[0.15453686498\ 0.13637989828\ 0.27269857877\ 0.10908701140\ 0.13637989828\ 0.19091774830]^T$

- Based on these computations, the initial page $x^{(0)}$ does not seem to have any influence on the rank of the pages.

Eigenvector of the Probability Transition Matrix B

- We assume that $\lim_{k\to\infty} x^{(k)} = x$. Since $x^{(k)} = Bx^{(k-1)}$, we have $x = Bx$. In this case, x is an eigenvector of B associated to the eigenvalue 1.
- In order for the entries of x to sum 1, we can normalize x by dividing each entry by the sum of the entries. We have

$$x = \frac{1}{110}\begin{bmatrix}17 & 15 & 30 & 12 & 15 & 21\end{bmatrix}^T \approx \begin{bmatrix}0.1545 & 0.1364 & 0.2727 & 0.1091 & 0.1364 & 0.1909\end{bmatrix}^T$$

We conclude that the rank of important pages is the same one obtained by Alice: 3, 6, 1, 2, 5, 4.

Final Remarks

We used SageMath in our computations. The results in both methods for the WebGraph in Figure 1 are comparable. Alice's procedure works well if the graph is small. For larger graphs, the use of the matrix B is more efficient. However, for the use in WWW, the calculations become very expensive as the size of the matrix is expected to be very large. It would be interesting to explore ways of reducing this computational cost.

References

[1] Chris Rorres Howard Anton.
Elementary linear algebra: Applications.
Wiley, 11th edition, 2014.

[2] Emille Davie Lawrence.
How does google do it? the pagerank algorithm.
National Math Festival, https://vimeo.com/216762443, 2017.

allan.emmanuel@mail.citytech.cuny.edu