LINEAR ALGEBRA APPLICATION: GOOGLE PAGERANK ALGORITHM.

JONATHAN MACHADO

Abstract. Google's PageRank algorithm is what makes Google such a strong search engine. The pioneering PageRank algorithm redefined how a search engine operates and executes. In this paper, the underlying mathematical basics for understanding how the algorithm functions are provided. A basic analysis of hyperlinks with its association to the algorithm and the PageRank algorithm is studied. Ultimately, this paper shines light on a neat application of linear algebra coupled with graph theory.

1. Introduction

Revolutionizing how the modern world operates, the Internet is a powerful medium in which anyone around the world, regardless of location, can access endless information about any subject and communicate with one another without bounds. All that is needed is a computer and the World Wide Web. One of the greatest results of the Internet was the establishment of hyperlinks. The World Wide Web is an extensive computer network consisting of billions of web pages holding documents of information. Hyperlinks are the pathways from one web page to another, initiating the capability of communication between these pages. Interactions between documents are performed by referencing one another via links. Here lies the foundation on how the most dominant search engine, Google, does its magic.

So, how does Google do it? Initially, Google breaks the web into sections, crawls through these segments, and adds it to their main index kept across thousands of different machines [2]. This process is done daily to keep Google's index of the web up-to-date. Now, a user visits Google, types in a query, and off the Google search engine goes to find the most relevant and important web pages to be shown in regards to what was searched. First, the query is decomposed into the individual words typed in the search engine [3]. Google then deploys programs known as spiders that crawl in Google's index in search for pages that include the words, across many machines [1]. These spiders start off on a few pages. They follow the links on the current page to other pages on a continuous search; and so on, until every page regarding the query is indexed [3]. All of these pages are combined together for Google to now apply over a hundred different ranking factors such as the quality of the page (authoritative, low quality, or spam), the location of the words (in the title, url, etc.), the proximity of the words (if the words are next to each in a sentence or not), time users have spent on the pages before, etc., to sort the resulting pages based on overall rank [1].

Notably, the famous PageRank algorithm created by Google's founders is the most critical component in determining the overall rank of a page. Throughout the searching process, the PageRank algorithm is main factor used to evaluate the pages that are most reputable and authoritative across the index. The derivation of the PageRank algorithm was what set Google apart from the rest early on and made it the successful, most powerful search engine to date. The PageRank algorithm revolutionized how search engines retrieved pages from

the web and truly displayed these pages in order of significance. In essence, the algorithm proposes that the relevance or importance of a web page is dictated by the number of quality hyperlinks linking to it. It is useful to represent these networks of hyperlinks linking web pages to each other as directed graphs. It turns out that linear algebra coupled with graph theory are the tools needed to calculate web page rankings by notion of the PageRank algorithm. The focus of this paper is to explain the underlying mathematics behind the Google's PageRank algorithm. We dive into fundamentals of the Google's PageRank algorithm, providing an overview of important linear algebra and graph theory concepts that apply to this process. In the end, the reader should have a basic understanding of the how Google's PageRank algorithm computes the ranks of web pages and how to interpret the results.

2. Mathematics behind the PageRank algorithm

2.1. **Markov Chains.** We begin by introducing Markov chains. We define a Markov chain as a mathematical model that describes an experiment or measurement that is performed many times in the same way, where the outcome of a given experiment can affect the outcome of the next experiment. The process starts at an initial state, namely x_0, and transitions successively from one state to another, say x_1, x_2,...,x_k. The outcome of a given state depends only on the immediately preceding state.

Definition 2.1. *A* **probability vector** *is a vector with nonnegative entries that add up to 1.*

We note probability vectors are the states in a Markov chain, hence these vectors are often referred to as *state vector*.

Definition 2.2. *A* **column-stochastic matrix** *is a square matrix in which all entries are greater than or equal to zero (nonnegative) and whose columns are probability vectors.*

Definition 2.3. *A matrix is* **positive** *if all its entries are positive (greater than zero) real numbers.*

Ultimately, we are interested in analyze the chain's long-term behavior after starting at some initial state. Thus, a Markov Chain can be expressed as the first-order difference equation or also referred to as a dynamical system:

$$x_{k+1} = Ax_k \text{ for } k = 0, 1, 2, \dots \tag{1}$$

where A is a column-stochastic matrix.

Note to compute x_k in general, we can use

$$x_k = A^k x_0 \text{ for } k = 0, 1, 2, \dots \tag{2}$$

So, we ask ourselves this question: what is the outcome at state x_k as time goes on? When studying these Markov Chains, usually as the system passes through time, the state vectors seems to approach an equilibrium. This special long-term outcome leads to the concepts of eigenvalues and eigenvectors.

Definition 2.4. *A* **eigenvector** *of a square matrix A is a nonzero vector $\vec{x}$ such that $A\vec{x} = \lambda\vec{x}$ for some scalar λ, where λ is an* **eigenvalue**.

Such an $\vec{x}$ is an eigenvector corresponding to λ. Additionally, in dynamical systems, if A is a column-stochastic matrix, there exists an eigenvalue $\lambda = 1$.

Theorem 2.1. *If A is a column-stochastic matrix, then it has an eigenvalue $\lambda = 1$.*

Theorem 2.2. *If A is a positive column-stochastic matrix, then there is a unique eigenvector corresponding to the eigenvalue $\lambda = 1$ such that it has only positive entries and the sum of its entries equals 1.*

Definition 2.5. *A* **steady-state vector** *or* **equilibrium vector**, *q, is a probability vector with eigenvalue $\lambda = 1$ such that*

$$Aq = q, \tag{3}$$

where A is a positive column-stochastic matrix.

Definition 2.6. *A column-stochastic matric, A, is* **regular** *or* **primitive** *if for some positive integer k, A^k results strictly in a positive matrix.*

Theorem 2.3. *If A is square regular column-stochastic matrix, then A has a unique steady-state vector, q. Furthermore, if x_0 is an initial state and $x_k = A^k x_0$ for k = 0,1,2,..., then the Markov Chain x_k converges to q as $k \to \infty$.*

2.2. **Graph Theory.** Now, it will be helpful to have a conception on graph theory basics.

Definition 2.7. *A* **graph** *is an object that consists of a non-empty set of vertices and another set of edges.*

In this case, we can refer to graphs as a *network*, vertices as *nodes*, and edges as *links* connecting the nodes.

Definition 2.8. *A* **directed graph** *or* **digraph** *is a set of nodes and a collection of directed edges that each connects an ordered pair of vertices.*

Definition 2.9. *For any two vertices i and j of a directed graph, if there is an edge from i to j or from j and i, the two vertices are* **adjacent**.

Definition 2.10. *A graph is* **connected** *if for distinct nodes i and j, there is a directed path either from i to j or from j to i.*

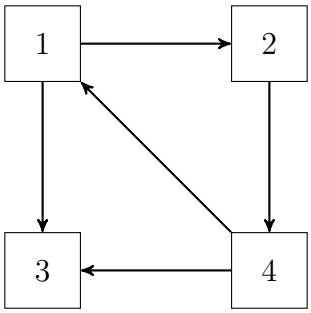

FIGURE 1. A connected graph.

Definition 2.11. *A graph is* **strongly connected** *if there is a directed path from every vertex to every other vertex.*

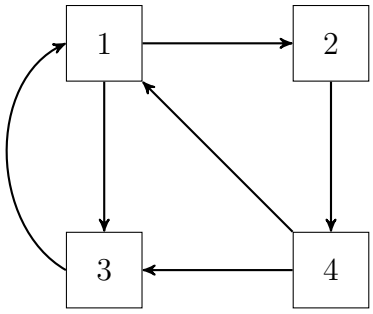

FIGURE 2. A strongly connected graph.

3. Google PageRank Algorithm

3.1. **Hyperlink Analysis.** Important properties and interesting outcomes of networks or graphs can be drawn out through matrix representation. Matrix representation of graphs successfully captures the characteristics of a given network and allows for the opportunity to deeply analyze its behavior, thus enabling many applications to arise.

The entire web can be viewed as a network of graphs with nodes representing webpages and edges representing the hyperlinks connecting them.

Definition 3.1. *An* **adjacency matrix** *is an $n \times n$ matrix containing 1's in its entries on row i, column j of the matrix if there is an edge from node i to node j and 0's otherwise.*

It follows that the web or a portion of the web in which one is interested in can be illustrated by an adjacency matrix. Any network has n finite nodes or webpages. Each webpage is indexed by an distinct integer p for $1 \leq p \leq n$. Now consider the web graph as shown in Figure 3. This network can represented as the adjacency matrix A:

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

Since we are ultimately interested in how the webpages are connected throughout networks to hopefully reach a conclusion of its long term behavior, lets take matrix A and multiply it by itself:

$$A^2 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 0 & 2 \\ 1 & 1 & 1 & 1 \\ 2 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \tag{5}$$

As it turns out, the resulting matrix from A^2 reveals the number of different paths having a distance of 2 units from webpage i to j. For instance, there are 2 paths from webpage 3 to 1 with a distance of 2: page 3 to page 2 to page 1 and page 3 to page 4 to page 1. On the other hand, there is no path of distance 2 between page 4 to 1.

Additionally, A^3 will inform the number of different paths having a distance of 3 units from webpage i to j and so on.

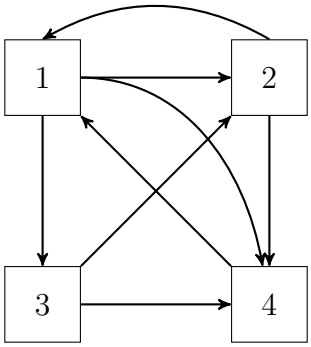

FIGURE 3. A strongly connected web graph representing hyperlinks linking four different websites. Regarding pages 1 and 2, they both have a backlink to each other.

Theorem 3.1. *Consider a directed graph and a positive integer k. Then the number of directed walks from node i to node j of length k is the entry on row i and column j of the matrix A^k, where A is the adjacency matrix.*

This neat result for adjacency matrices leads to insight on how a user starting on a particular webpage can transition to other pages. Consequently, in time, as the user surfs the web in relation to his/her query, he/she will eventually visit the webpages with the most hyperlinks since many other pages lead to it. Google's PageRank algorithm ultimately utilizes this information of hyperlink connections to conjure up the ranks of the pages.

3.2. **PageRank Algorithm Analysis.** Google's PageRank algorithm takes the hyperlink analysis slightly further. In addition to the number of hyperlinks a particular webpage has, the PageRank algorithm pays close attention to how reputable and authoritative those pages from the incoming hyperlinks are. To incorporate this factor into a web graph, weights are given to each hyperlink.

Definition 3.2. *The* **indegree** *of a node is the number of edges pointing to it.*

Definition 3.3. *The* **outdegree** *of a node is the number of edges pointing away from it.*

Weights are computed as follows: If there is an edge from i to j and the outdegree of node i is d_i, then the weight for that edge is $\frac{1}{d_i}$. The application of weights brings forth fairness in this ranking system. Think of it this way: Weights are motions. A page that links to another is a vote that the other page is important and therefore makes a motion to raise the page's rank. The incorporation of weights attempts to not allow pages that link to many others, commonly referred to as *hubs*, to unreasonably effect the ranks, essentially treating each link with equal value. Additionally, a page that has many links pointing to it from these hubs will not receive an overwhelming influence that results in an unfair rankings.

Definition 3.4. *A* **transition matrix***, corresponding to an adjacency graph, incorporates weights to better model the behavior the network.*

We note that the weight corresponding from edge i to j is placed on column i and row j. Returning to our web graph shown in Figure 3, we now consider constructing a transition

matrix to perform the PageRank algorithm:

$$T = \begin{bmatrix} 0 & \frac{1}{2} & 0 & 1 \\ \frac{1}{3} & 0 & \frac{1}{2} & 0 \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix} \tag{6}$$

We note that we have nonnegative column-stochastic matrix.

Theorem 3.2. *For any strongly connected graph, the transition matrix is column-stochastic.*

Google's PageRank algorithm views this network as a dynamical system to conclude its long term behavior. Using equation 2, this network at hand can be expressed as

$$x_k = T^k x_0 \text{ for } k = 0, 1, 2, \ldots, \tag{7}$$

where T is our transition matrix and x_0 is the vector

$$x_0 = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \end{bmatrix}, \tag{8}$$

encompassing the fact initially the four webpages start out with equal rank.

The dynamical system above models a random user's movement through the web with respect to time. In this case, a user's movement through the network illustrated in Figure 3. At first, a user decides on visiting one of the four pages randomly with equal probability, hence x_0. After some time, the user notices the hyperlinks on the page and transitions to another webpage. At each transition, the probability of moving to a particular website is the weight corresponding from the current page to that page. As time passes by, the user would have visited every website in the network. Every link to specific webpages encountered in a currently visited page increases the probability of that page's chance to get visited. This is all equivalent to a page's rank. Google's PageRank algorithm considers these probabilities of each page as their rank. Ultimately, this is what the PageRank algorithm calculates to determine a page's rank. The notion of a user surfing the web of interest and each transition increases or decreases a page's probability or rank is equivalent to repeatedly multiplying the transition matrix over and over again. At some point, as time passes and the user continuously surfs the web, the probabilities or rank of the webpages reach an equilibrium. Seeing as this is network is a dynamical system transitioning through time, we can apply the properties of Markov Chains. Since we have a column-stochastic matrix in our system, applying theorem 2.1, we are guaranteed an eigenvalue of 1. Furthermore, by definition 2.5, a steady-state vector exists. Repeatedly computing the state vectors until a steady vector is reached or deriving the probabilistic eigenvector corresponding to eigenvalue $\lambda = 1$ is what Google's PageRank algorithm does. So, in our case, our transition matrix does in fact have an eigenvalue of 1. Solving for the steady-state vector q, we arrive at

$$q = \begin{bmatrix} .387 \\ .194 \\ .129 \\ .290 \end{bmatrix}. \tag{9}$$

Thus, the ranks of the four webpages are given above. Page 1 is ranked highest with .387; page 4 is ranked second highest with .290; followed by page 2 with .194; and lastly page 3 with .129.

References

[1] Amy N Langville and Carl D Meyer. *Google's PageRank and beyond: The science of search engine rankings*. Princeton University Press, 2011.

[2] Raluca Tanase, Remus Radu. The Mathematics of Web Search. `http://www.math.cornell.edu/~mec/Winter2009/RalucaRemus/index.html`, 2017. [Accessed 10 September, 2017].

[3] Danny Sullivan. How search engines work. *SEARCH ENGINE WATCH, at http://www. searchenginewatch. com/webmasters/work. html (last updated June 26, 2001)(on file with the New York University Journal of Legislation and Public Policy)*, 2002.

Department of Mathematics and Statistics, University of North Carolina at Greensboro, Greensboro, NC 27402, USA

E-mail address: `jomachad@uncg.edu`