MATHEMATICS BEHIND GOOGLE'S PAGERANK ALGORITHM

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE MASTERS OF SCIENCE IN MATHEMATICS

IN TEXAS WOMAN'S UNIVERSITY

MATHEMATICS

TEXAS WOMAN'S UNIVERSITY

BY

BRIAN MOOR

DENTON, TEXAS

AUGUST 2018

ABSTRACT

MATHEMATICS BEHIND GOOGLE'S PAGERANK ALGORITHM

BRIAN MOOR

AUGUST 2018

Google has become a name synonymous with web searching, to the point where Merriam Webster defines google as "verb: to use the Google search engine to obtain information about (someone or something) on the World Wide Web." Google, Inc. was founded in 1998, and quickly became the leader of Internet search engines. With its use of the algorithm called PageRank, it retrieved search results that were much more relevant to the interests of its users. The PageRank algorithm works on the basic theory that the more important and useful a page is, the more other pages will link to it. Therefore, a page that has many other pages linking to it is more important, and will appear higher in the search results. This proved much more effective than competing search engines at the time, many of which were still searching primarily by keywords, something that was very easily abused. Since the introduction of the PageRank algorithm, there has been many efforts towards the improvement and optimization of the mathematics behind it. There have also been much work to apply the PageRank algorithm to other fields and areas of research. These include determining the relative importance of authors in published in scientific journals, finding potential interactions in proteins, and determining the relative importance of various species in a food web. This thesis will explore the history and mathematics behind the PageRank algorithm, and the optimizations and expanded uses it has found over the years.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

After Google was founded in 1999, it quickly became the leader of Internet search engines. With the introduction of the PageRank algorithm, it changed how web search providers found the search results for their users. The PageRank algorithm works by giving individual web pages a rank, determined by the number of links that are pointed towards the page. The more websites that link to a site, the more valuable the content of that site is considered, and the higher its rank. Some pages are more important than others, so the value of the links to other websites is determined by the rank of the page it comes from. Google included a number representing the PageRank of a webpage on its Google Toolbar when it launched in 2000. This put the PageRank in the public eye and caused many to focus on how to manipulate the number to increase their web traffic. While the PageRank algorithm was a major innovation for the search engines of the Internet, it also has found use in other areas. This paper will take a deep look into the PageRank algorithm, the methods it contains, and the other places where it can be used. It will examine the background mathematics necessary for the understanding of the algorithm, and walk through an example system building up to the PageRank algorithm.

CHAPTER II

BACKGROUND MATHEMATICS

2.1 Graph Theory

One of the essential foundations necessary for representing the Internet network is graph theory. Graph theory is the study of graphs, mathematical structures that model pairwise relations between objects. The first paper that shows graph theory was the ”Seven Bridges of Königsberg” by Euler (1759). The Königsberg problem involved trying to find a path that traveled each of the seven bridges connecting four landmasses, crossing each bridge only once. There Euler (1759) laid the groundwork for the field by reducing the bridge problem to a graph, where the landmasses were vertices and the bridges were edges. Graphs may be made up of vertices and edges, where vertices are points on the graph and the edges are the lines that connect them. An undirected graph makes no distinctions of the direction of an edge between two vertices; while in a directed graph, edges have directions.

A graph is made up of two sets, a set of vertices, V, and a set of pairs of vertices, the edges, E. The undirected graph in Figure 2.1 can be written as:

$$V(G) = \{A, B, C, D, E, F\}$$
$$E(G) = \{(A, B), (A, C), (A, D), (B, D), (C, D), (C, E), (D, E), (D, F), (E, F)\} \quad \text{(II.1)}$$

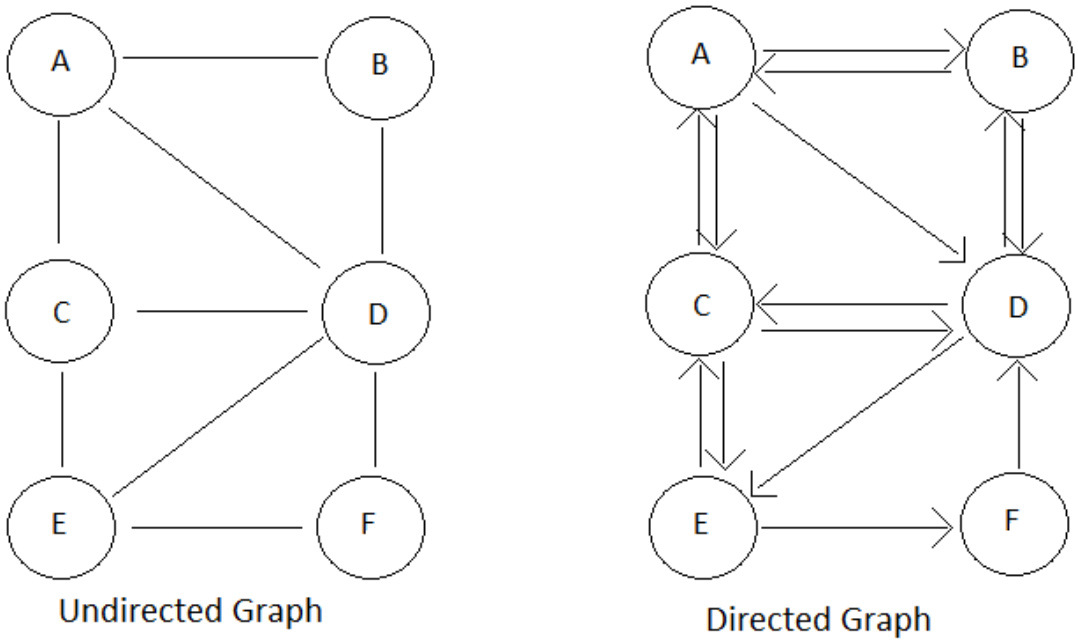

Figure 2.1: Undirected and Directed Graph

The directed graph model in Figure 2.1 can be written as:

$$\begin{aligned} V(G) =& \{A, B, C, D, E, F\} \\ E(G) =& \{(A, B), (A, C), (A, D), (B, A), (B, D), (C, A), (C, D), \\ & (C, E), (D, B), (D, C), (D, E), (E, C), (E, F), (F, D)\} \end{aligned} \tag{II.2}$$

Since the edges are directed in this graph, the edges such as (A, B) and (B, A) are distinct and separate. Graphs have been used in many fields and for many purposes, from modeling atomic structures to modeling traffic networks. They have also been useful in computer science, and are an effective method of modeling the structure of the Internet network.

2.1.1 Notation

The notation and mathematics we will use in this paper are explained here. All matrices are $n \times n$, and vectors are $n \times 1$, and all matrix entries are real numbers. The transpose of a vector v is the $1 \times n$ vector v^T. The vector **e** is a

column vector of all ones. **PR**(A) is the PageRank of the page A, the damping factor is d, conventionally defined by Brin and Page (1998) as 0.85, and the number of pages linked is N.

2.2 Application to the Internet

The importance of graph theory to the Internet is in how it is modeled. The Internet is made up of many webpages, each containing information on its page. Each page may also contain links to other pages. This can be modeled using graph theory, where each page is a vertex, and the edges between each vertex are equivalent to the links from one page to another. This can be modeled as such:

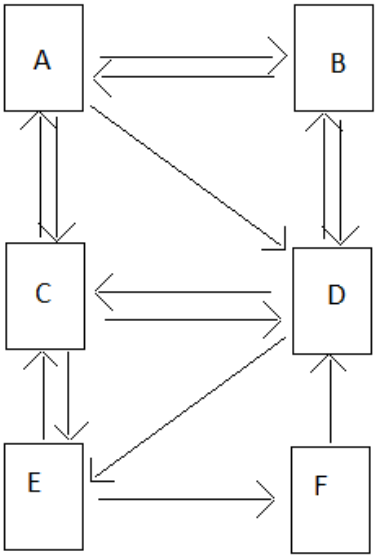

Figure 2.2: Model Internet

This model of a simplified set of webpages shows a graph of directed edges, representing pages and the outgoing links between them. This simplified model of the Internet is easily expandable and is useful for describing the methods used in the PageRank algorithm. This model can be represented in matrix form where:

$$S_{ij} = \begin{cases} 1 & \text{if } p_j \text{ links to } p_i \\ 0 & \text{otherwise} \end{cases} \tag{II.3}$$

For the model Internet network in Figure 2.2, the matrix is:

$$\mathbf{S} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{II.4}$$

It is also useful to model it as:

$$S_{ij} = \begin{cases} 1/l_{p_j} & \text{if } p_j \text{ links to } p_i \\ 0 & \text{otherwise} \end{cases}, \tag{II.5}$$

Where l_{p_j} is equal to the number of outgoing links on page p_j. The matrix for the model Internet network in Figure 2.2 is then:

$$\mathbf{S} = \begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 & 0 \\ 0 & 1/3 & 0 & 0 & 1/3 & 1/3 \\ 0 & 0 & 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{II.6}$$

This is also a row-stochastic matrix, which is shown in Section 2.3.

2.3 Markov Chains

A row-stochastic matrix is a non-negative $n \times n$ matrix where each row sums to 1. It is important here for its use in Markov chains. A *Markov chain* is a stochastic process that satisfies the Markov property, a memoryless property where the probability of future behavior is independent of past behavior (Meyer, 2000). The Internet network matrix can also be defined by using the *transition probability*:

$$S_{ij} = P(X_t = p_j | X_{t-1} = p_i) \tag{II.7}$$

where each entry can be described as the odds that a web surfer would follow a link to page p_j, given that they were on page p_i. Since: $\mathbf{S} = [S_{ij}]$, then **S** is a *transition matrix*.

With a transition matrix **S** we can then find a *probability distribution vector* **p**, a non-negative vector where:

$$\mathbf{p}^T = (p_1 \ p_2 \ \ldots \ p_n) \text{ such that } \sum_k p_k = 1 \tag{II.8}$$

In a Markov chain, the kth step probability distribution vector is:

$$\mathbf{p}^T(k) = (p_1(k) \ p_2(k) \ \ldots \ p_n(k)), \ k = 1, 2, ..., \text{ where } p_j(k) = P(X_k = S_j) \tag{II.9}$$

Where $p_j(k)$ is the probability of being on page p_j on the kth step, and the *intitial distribution vector*:

$$\mathbf{p}^T(0) = (p_1(0) \ p_2(0) \ \ldots \ p_n(0)), \text{ where } p_j(0) = P(X_0 = S_j) \tag{II.10}$$

is the probability that the Markov chain starts in S_j. These vectors can be used with the Internet network matrix to show the likelihood of being on an individual page p_j.

It is important to be able to describe the k^{th} step distribution for the given initial distribution vector $\mathbf{p}^T(0)$. Using elementary probability laws, it is easy to determine that $\mathbf{p}^T(1) = \mathbf{p}^T(0)\mathbf{S}$. Because a Markov chain has no memory, then $\mathbf{p}^T(2) = \mathbf{p}^T(1)\mathbf{S}$, which acts as if $\mathbf{p}^T(1)$ was the initial distribution. Continuing on in this manner with substitution reveals that:

$$\begin{aligned} &\mathbf{p}^T(k) = \mathbf{p}^T(k-1)\mathbf{S} = \mathbf{p}^T(k-2)\mathbf{S}^2 = ... = \mathbf{p}^T(0)\mathbf{S}^k \\ &\mathbf{p}^T(k) = \mathbf{p}^T(0)\mathbf{S}^k \end{aligned} \tag{II.11}$$

Where p_{ij} in $\mathbf{S}$ is the probability of moving from page p_i to page p_j in k steps.

Because the matrix $\mathbf{S}$ is a stochastic matrix and $\lambda_1 = 1$ is the dominant eigenvalue of $\mathbf{S}$, the eigensystem:

$$\pi^T\mathbf{S} = \pi^T, \quad \pi \geq \mathbf{0}, \quad \pi^T\mathbf{e} = 1 \tag{II.12}$$

has a unique solution π, called the *stationary distribution vector*. The vector π is the dominant left eigenvector corresponding to λ_1. The i^{th} component π_i represents the percentage of being on page i. With the Internet network matrix, this is useful for determining the likelihood of being on a webpage by following links, and is an essential part of the PageRank algorithm, discussed more in Section 3.2.

2.3.1 Stochastic Matrices

There are several properties that can be known from a stochastic matrix. The *spectral radius* of a matrix is defined as:

$$\rho(A) = \max_{\lambda \epsilon \sigma(A)} |\lambda|, \tag{II.13}$$

Where $\sigma(A)$ is the set of distinct eigenvalues of the matrix. The *infinite norm* of a matrix is defined as:

$$||A||_\infty = \max_i \sum_j |a_{ij}|, \tag{II.14}$$

The norm creates an upper bound on $\rho(A)$, so it is also true that:

$$\rho(A) \leq ||A|| \tag{II.15}$$

All stochastic matrices have a row sum of 1, therefore $||A||_\infty = 1$. Equivalently, **Ae** $=$ **e**, where **e** is a vector of all ones. So for all stochastic matrices, there is the associated eigenpair $(1, \mathbf{e})$, and therefore:

$$1 \leq \rho(A) \leq ||A||_\infty = 1 \Rightarrow \rho(A) = 1. \tag{II.16}$$

We can show this to be true for the matrix for the model network given in II.6. The eigenvalues for **S** are:

$$\lambda_1 = 1, \quad \lambda_2 = 1/3, \quad \lambda_3 = \frac{-\sqrt{7}-1}{6}, \quad \lambda_4 = \frac{\sqrt{7}-1}{6}. \tag{II.17}$$

The *dominant eigenvalue*, where $|\lambda_i| > |\lambda_j|$ for all j, is $\lambda_1 = 1$. The eigenvectors for **S** are:

$$v_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad v_2 = \begin{bmatrix} \frac{-17}{9} \\ \frac{-7}{3} \\ \frac{1}{9} \\ \frac{1}{3} \\ \frac{5}{3} \\ 1 \end{bmatrix} \quad v_3 = \begin{bmatrix} \frac{\sqrt{7}+1}{3} \\ -1 \\ \frac{-\sqrt{7}-1}{3} \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad v_4 = \begin{bmatrix} \frac{-\sqrt{7}+1}{3} \\ -1 \\ \frac{\sqrt{7}-1}{3} \\ 0 \\ 1 \\ 0 \end{bmatrix} \tag{II.18}$$

The eigenvector corresponding to the dominant eigenvalue, v_1, is the *dominant eigenvector*, which is shown to be positive. Since $\lambda_1 = 1$ and $v_1 = \mathbf{e}$, it can be seen that the stochastic matrix **S** has the eigenpair $(1, \mathbf{e})$.

CHAPTER III

GOOGLE'S PAGERANK ALGORITHM

3.1 Background

In 1995, Brin and Page met at Stanford University, where Brin showed Page around when he was considering attending graduate school there (How we started and where we are today, n.d.). By the next year, Brin and Page had formed a partnership working on a program called BackRub, a search engine that used links to determine the importance of pages on the World Wide Web. This was later renamed as Google™, and in 1998, Brin and Page founded the company Google, Inc. The original algorithm and method for the Google search engine was named PageRank. PageRank was described in a paper published by Brin and Page (Brin & Page, 1998; Brin, Page, Motwani, & Winograd, 1999). There, Brin and Page detailed the algorithm and justified the process by hypothesizing a web surfer who would start on a random page and click on links until they were bored and started on another random page. Over the years since Brin and Page's first publication, there has been much exploration and optimization of their work. This paper will look at that and the history of Google and the algorithm it uses.

3.2 PageRank Algorithm

In their 1998 paper, Brin and Page gave the algorithm for finding the PageRank as:

$$\mathbf{PR}(A) = (1-d) + d\left[\frac{\mathbf{PR}(T_1)}{\mathbf{L}(T_1)} + ... + \frac{\mathbf{PR}(T_n)}{\mathbf{L}(T_n)}\right], \qquad \text{(III.1)}$$

where they find the PageRank (**PR**) of page A, by taking the the PageRank of all pages that link to A, defined here as T, divided by the number of outgoing links on each page, defined as L. The parameter d is a dampening factor, which can be set from between 0 and 1, given as 0.85 in their paper. It is meant to simulate the number of links that a random surfer will follow before they go to a random, unlinked page.

Brin and Page (1998) also claimed that the PageRanks formed a probability distribution over all web pages, so that the sum of all of them would be 1. This is not the case with the algorithm they gave, so it must be modified as such:

$$\mathbf{PR}(A) = \frac{(1-d)}{N} + d\left[\frac{\mathbf{PR}(T_1)}{\mathbf{L}(T_1)} + ... + \frac{\mathbf{PR}(T_n)}{\mathbf{L}(T_n)}\right], \tag{III.2}$$

Where N is the total number of pages in the network. With this modification it then forms a probability distribution. By performing a number of iterations of the algorithm, the PageRanks of all pages in the network can be determined. This can be written generally as:

$$\mathbf{PR}(a_i) = \frac{(1-d)}{N} + d\sum_{a_j \in G(a_i)} \frac{\mathbf{PR}(a_j)}{\mathbf{L}(a_j)}, \tag{III.3}$$

Where a_i is a webpage, and a_j is a page with an outgoing link to a_i.

As shown in Section 2.2, the link matrix can be defined by:

$$\mathbf{H}_{ij} = \begin{cases} 1/l_{p_j} & \text{if } p_j \text{ links to } p_i \\ 0 & \text{otherwise} \end{cases}, \tag{III.4}$$

Where l_{p_j} is equal to the number of outgoing links on page p_j.

The PageRanks for the model network are shown in Figure 2.2. Giving each page a starting PageRank value of $\frac{1}{N}$, then iterating the algorithm until the dif-

ference between two iterations is less than an arbitrary small value, e:

$$\mathbf{PR} = \begin{bmatrix} 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 & 0 \\ 0 & 1/3 & 0 & 0 & 1/3 & 1/3 \\ 0 & 0 & 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \tag{III.5}$$

After the differences are less than e:

$$\mathbf{PR} = \begin{bmatrix} 0.1203 \\ 0.1441 \\ 0.1203 \\ 0.3000 \\ 0.1441 \\ 0.1712 \end{bmatrix} \tag{III.6}$$

Here, the sum of all PageRanks equals 1. However, there are instances where this approach does not work.

3.2.1 Dangling Nodes

The structure of the Internet network is not as neat as the example we have given. One major issue for the process we have shown so far is when a webpage has no outgoing links. If the outgoing link from webpage F is removed,

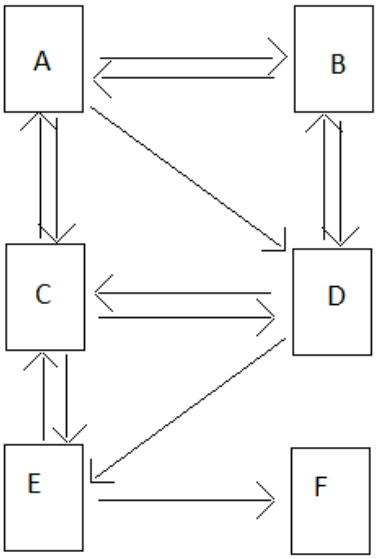

Figure 3.2: Dangling Node

then the matrix for the model becomes:

$$\mathbf{H} = \begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 & 0 \\ 0 & 1/3 & 0 & 0 & 1/3 & 1/3 \\ 0 & 0 & 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{III.7}$$

$$\mathbf{PR} = \begin{bmatrix} 0.0807 \\ 0.0772 \\ 0.0807 \\ 0.1035 \\ 0.0772 \\ 0.0871 \end{bmatrix} \tag{III.8}$$

It is clear that the sum of the PageRanks does not equal 1. This is because the webpage F is a *dangling node*, a webpage with incoming links but no outgo-

ing links. In a follow up paper in 1999, Brin, Page, Motwani, and Winograd stated that they removed all dangling nodes before calculating PageRank. There have been several fixes for dangling nodes since then, which is explored more in Section 4.2. One simple one was replacing the zero row for the dangling node in the link matrix with a row where each entry is $\frac{1}{N}$, effectively creating a link from the dangling node to every other page in the network.

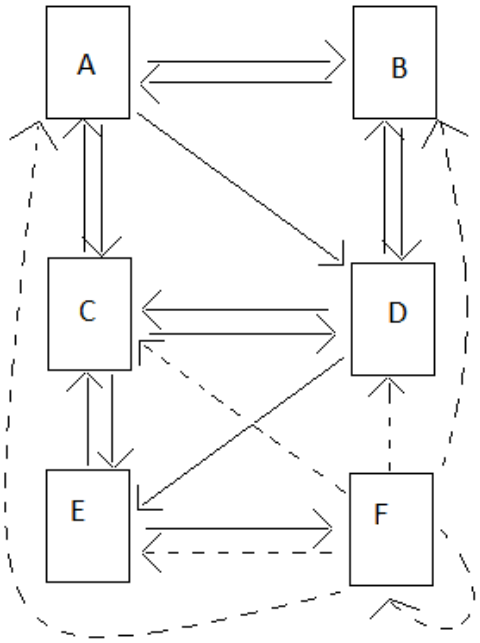

Figure 3.2: Dangling Node Fix

Taking **D** as a column vector that identifies dangling nodes, and a uniform row vector $\mathbf{w} = (\frac{1}{N}\frac{1}{N}...\frac{1}{N})$,

$$\mathbf{M} = \mathbf{H} + \mathbf{Dw}, \tag{III.9}$$

Recalculating III.5,

$$\mathbf{PR} = \begin{bmatrix} 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 & 0 \\ 0 & 1/3 & 0 & 0 & 1/3 & 1/3 \\ 0 & 0 & 1/2 & 0 & 0 & 1/2 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{bmatrix}, \tag{III.10}$$

And calculating the PageRank from that,

$$\mathbf{PR} = \begin{bmatrix} 0.1593 \\ 0.1524 \\ 0.1593 \\ 0.2044 \\ 0.1524 \\ 0.1721 \end{bmatrix} \tag{III.11}$$

Here, the PageRanks sum to 1.

In matrix notation (Wills, 2007), the Google matrix can be created with the equation:

$$\mathbf{G} = d * \mathbf{M} + (1 - d)\mathbf{e} * \mathbf{v} \tag{III.12}$$

Where **e** is a column vector of ones, and **v** is the *personalization vector*, a row probability distribution vector which is normally given as $\mathbf{v} = (\frac{1}{N}\frac{1}{N}...\frac{1}{N})$. The PageRank algorithm can then be written in matrix notation as:

$$[\mathbf{r}^k]^T = [\mathbf{r}^{k-1}]^T * \mathbf{G} \quad \text{where } k = 1, 2, ..., \tag{III.13}$$

Which is calculated where $[\mathbf{r}^0]^T = \mathbf{v}$, and continued until $||\mathbf{r}^k - \mathbf{r}^{k-1}|| < e$, where e is an arbitrarily small number. In Section 2.3, it is shown how to find π, the stationary distribution vector, which would be the same when $||\mathbf{r}^k - \mathbf{r}^{k-1}|| = 0$. For the PageRank algorithm, an exact solution is not necessary, and can be prohibitively expensive to compute. These are the algorithms and methods used by Brin and Page in their original papers. Other approaches and optimizations of the methods are explored in Chapter IV.

CHAPTER IV

USES IN WEB SEARCHES

4.1 Google's Changes

Google has launched many updates over the years in a constant attempt to combat spam and make sure that their search results show the most relevant sites (Wall, 2009-2017). On December 1, 2000, Google introduced the PageRank Toolbar, an update that added several search features to a toolbar on the browser. One important part of the update was the addition of a display of the PageRank of the site, which brought the PageRank number into the public eye.

Google has done many minor updates over the years that were not documented for the public, but major ones have been. After a series of unconfirmed updates, Google put out the Florida update, which drastically changed the PageRanks of many webpages, mostly cracking down on some deceptive tactics to adjust PageRank. After that came the Brandy update, with a massive index expansion, synonym expansion, and increased attention to anchor text. The next major update was actually a collective effort with Google, Yahoo!, and Microsoft introducing the nofollow attribute, which could be added to links in webpages to show they were not approved by the page and not to be counted in the PageRanking. This helped deal with spamming of links in comments and forums. In June of 2005, Google introduced the Personalized Search, which integrated a user's search history into their search results.

Over the next few years, Google put out updates Jagger, Big Daddy, Vince, and Mayday, with mostly minor changes. Caffeine, an update in June of 2010, introduced major changes and integration of crawling and indexing, resulting in a fresher index. Several more updates rolled out until February 2010 with

the Panda update, which cracked down on content farms and pages with high amounts of ads. More updates followed, with Venice integrating more local search data, Penguin adjusting more spam factors, Exact-Match Domain Update which changed how it handled exact match domains, and the Payday Loan update, which focused on spammy sites such as payday loans and porn. The update Hummingbird in August 2013 was a core algorithm update, and then Pigeon updated how it handled local results. In April 2013, Google released the Mobile-Friendly Update, which separated mobile rankings for websites with mobile friendly sites, and then the Quality update was a core algorithm change. In October 2015, Google announced that they had had an update called RankBrain, which incorporated machine learning into the algorithm. There have been many other minor updates to Google as it continues to improve its service.

Google has put much work into improving their search engine, a large part of which was combating spam and other methods of search engine manipulation. Other changes have addressed improving the precision of the results in regards to the search terms used, or in regards to the location of the searcher. Throughout all of these updates, the PageRank algorithm has remained an important part of the search engine.

4.2 Optimization of PageRank

The PageRank algorithm is an effective method for ranking importance in a link-based structure. However, it can have many issues, including a high computational cost in large structures, and its vulnerabilities to manipulations. There has been much work investigating these issues.

4.2.1 Performance Improvements

In their 2007 paper, Wills explored the use of the power method to compute the PageRank vector. The convergence rate for the power method depends only on the damping factor, d. While the power method is easily used in smaller systems, the Google matrix for the entire Internet network contains more than 25 billion rows and columns. Therefore, it is not feasible to compute an exact solution, but to approximate the PageRank.

As shown in equation III.13, the PageRank **r** can be calculated with the power method. However, with the large size of the Google matrix **G**, it becomes computationally expensive to calculate it this way. Wills shows that the equation can be broken down in a way that is cheaper to calculate. By taking III.12 and III.9, and replacing it into III.13, we obtain:

$$\begin{aligned}[\mathbf{r}^k]^T &= [\mathbf{r}^{k-1}]^T(d(\mathbf{H}+\mathbf{Dw})+(1-d)\mathbf{ev}) \\ &= d[\mathbf{r}^{k-1}]^T\mathbf{H}+d([\mathbf{r}^{k-1}]^T\mathbf{D})\mathbf{w}+(1-d)([\mathbf{r}^{k-1}]^T\mathbf{e})\mathbf{v} \\ &= d[\mathbf{r}^{k-1}]^T\mathbf{H}+d([\mathbf{r}^{k-1}]^T\mathbf{D})\mathbf{w}+(1-d)\mathbf{v}\end{aligned} \tag{IV.1}$$

Where $[\mathbf{r}^{k-1}]^T * \mathbf{e} = 1$, because $[\mathbf{r}^{k-1}]^T$ is a probability vector, which sums to 1. The formula is the sum of three vectors, with the only vector-matrix multiplication being with **H**. Since the **H** matrix has more zero elements than **G**, it is cheaper to calculate.

Wills also looked at the termination criteria of the power method. The method we have shown so far is to use the power method until $||\mathbf{r}^k - \mathbf{r}^{k-1}|| < e$. While this is effective for finding the PageRanks, if the only interest is in providing a ranking to the webpages, it may be easier to solve only until a useful ranking can be obtained. For this, instead of a termination criteria being when $||\mathbf{r}^k - \mathbf{r}^{k-1}|| < e$, they instead measure the correlation between the rankings in successive iteration, measured as the Kendall's τ coefficient.

The Kendall's τ coefficient is a way to measure the correlation between two list rankings. τ is defined as:

$$\tau = \frac{1 - 2s}{\frac{1}{2}n(n-1)} \tag{IV.2}$$

Where s, Kendall's distance, is the number of different elements in the lists of ordered pairs in two rankings of n items. When $\tau = 0$, then the two lists are identical.

There are issues with the different ranked approximations. Wills listed several problems, where correct ranking could occur in one iteration and be destroyed in the next, instances where small residual norm does not guarantee a correct ranking, instances where $\tau = 0$ does not guarantee a correct ranking, and instances where the correct ranking occurs much earlier than the termination criteria. Wills created their own criteria for the ranking of the elements, described by them in theorem 4.5 and 4.7 (2007). Wills used these theorems to produce a computationally efficient criterion for ranking PageRank.

4.2.2 Combating Manipulation

Since the original PageRank algorithm, dealing with manipulation by webpages has been a major issue. This has been a problem mostly through *spam*, or the insertion of large numbers of links to point to a page in an attempt to inflate their PageRank number artificially.

In their 2003 paper, Haveliwala and Kamvar showed how the second eigenvalue of the Google matrix could be used to detect spam, as well as speed up PageRank computation. They show that for the web hyperlink matrix that the second eigenvalue $\lambda_2 = c$, where c is the damping number used in the PageRank algorithm. This can be useful to make PageRank computations faster by not having to compute λ_2. It can also help identify spam sites. The eigenvec-

tors that correspond to $\lambda_2 = c$ are associated with subgraphs of the web link graph which may have incoming edges but no outgoing edges, a structure often generated by link spammers to increase their rank.

In their 2015 paper, Sangers and van Gijzen built on that to show how one could deal with link spamming using the eigenvectors related to the second eigenvalue of the Google Matrix. They found that a set of states S is a closed subset of the Google matrix G only if $i \in S$ and $j \notin S$, then that implies that $p_{ij} = 0$. In other words, that there are no outgoing connections from the subset to the rest of the web. S is an irreducible closed subset if there is no proper subset of S that is a closed subset. Irreducible closed subsets correspond to the structures built by link spammers to hoard PageRank. The eigenvectors corresponding to the second eigenvalues of the Google matrix have a non-zero value corresponding to the irreducible closed subsets and a zero in other nodes. One solution proposed by them was to utilize the personalization vector v to lower the PageRank of suspected pages, by giving a small value in the corresponding node. This personalization vector allows for adjustments to the PageRank values without changing the basic formula.

4.3 Other Search Engines

Google was built using the PageRank to weigh web pages in relevance to searches. Google uses PageRank in combination with weighing the text and contents of the page, link text, and even capitalization of words to determine relevance of the page to the search terms. Other search services also started using similar methods to rank web pages for searches. This paper explores more of the variations that other services may use, and how Google may have changed what they have done in the past 20 years (Facts about Google and Competition, n.d.).

One of the earliest search engines was Archie, created in 1990 by Emtage. This was basically a database of web file names with a regular expression matcher to serve user queries. In 1993 came Jumpstation, a search engine that crawled the Internet for page titles and headers, and returned them for simple search matches. Webcrawler was released in April of 1994, the first engine to allow users to search for any word in the webpage. Lycos soon followed in July of 1994, with ranked relevance retrieval, prefix matching, and word proximity bonuses in their search, as well as a large catalog. Altavista came on December of 1995, one of the first to allow natural language queries. May of 1996 saw the release of the HotBot search site. In April of 1997, Ask Jeeves was launched as a natural language search engine; using human editors to match search queries, and ranked results based on popularity. At this time is when Google entered the scene, becoming one of the major search engines. Yahoo! used Google as its search engine provider until 2003, when they started using their own search engine, from all of the companies they had acquired. MSN was launched in 1998, owned by Microsoft, which used the Hotbot search engine until 2005, when they developed their own. In 2009, they changed its name to Bing. In July of 2009, Microsoft and Yahoo! announced that the Yahoo! search engine would be powered by Bing. Many of the search sites have shut down or been bought out by others, leaving it now with Google at the lead, with 1,800,000,000 monthly visitors, followed by Bing at 500,000,000, Yahoo! at 490,000,000, and Ask Jeeves at 300,000,000 as of July 2017 ”Top 15 Most Popular Search Engines June 2017”.

CHAPTER V

USES IN CITATION NETWORKS

5.1 Measurement of Author Impact

One of the areas that the PageRank algorithm has been most extensively studied is in citation networks. The measurements of a researcher's productivity for many years have mostly been by the number of publications that they have released. In 1972, Garfield proposed for scientific journals a measurement of their impact on the scientific community. Their *impact factor* was measured by taking the average of the number of citations that a journal received in the past two years. This was done in an attempt to curtail the large number of what was seen as unnecessary journals at the time. It has later been adapted to use for individuals to measure their scientific worth, taking the average number of citations they received across all of their published work.

In order to determine the impact of those publications, it is common to measure the number of times that each work has been cited. This has been used for various procedures such as hiring, promoting, or awarding grants, but it also has faced criticism. While a paper may have a large number of citations, it may be only due to how well-known the work is, while another work may have a significant influence in the field but is only referenced by a handful of other papers. By applying the PageRank algorithm to the process, it can be used to measure the relative importance of papers and their authors, where even almost forgotten papers that influenced later important works can be shown for their value.

Another method was proposed by Hirsch in his 2005 paper. The *Hirsch index* or *h-index* is a number to characterize the scientific impact of a researcher. The *h-index* is the maximum value of the author's h number of publications that

have *h* citations. With this method, the *h-index* only changes if the number of citations a new publication receives is higher than the previous *h-index* value of the author. Hirsch showed that with this method, the physicist Nobel prize winners of the last 20 years had larger than average *h-index* numbers.

5.2 Potential Improvements

In their paper, Chen, Xie, Maslov, and Redner (2007) used the PageRank algorithm to assess the relative importance of papers beyond citation count, measured as their Google number. They attempt to take into account the effect of an important paper citing another, and lessen the effect of papers with large citation counts. With a simple replacement of each paper for a webpage, and each citation for an outgoing link, the PageRank algorithm is used to ranks the importance of papers. They took the damping number $d = 0.5$, with the justification that by the second reference, you would be likely to find a paper that was referenced by both the original and referenced papers.

In Chen et al.'s (2007) findings, they found that using the PageRank algorithm resulted in similar results compared to citation count for most papers, but that in several cases, papers appeared with high Google numbers that had relatively low citation counts. Many of those papers were easily recognized as seminal publications. One in particular was noteworthy for having an extremely low citation rank for its Google number, *The Theory of Complex Spectra* by J. C. Slater. This paper introduced the determinant form for the many-body spectra, which is so well known that few papers bother to cite the original work. This is one of the cases where the referenced paper was brought up in the rankings by the relative importance of papers that cite it. It is a case for the use of the PageRank algorithm to give importance beyond a simple citation count.

Yan and Ding (2010) explored more with the algorithm by applying it to co-authorship network analysis in the informetrics community. Yan and Ding (2010) worked to measure both the authors' academic impact via citation count, as well as their community impact, via co-authorship of papers. By measuring their community impact, Yan and Ding sought to quantify an author's social capital. Yan and Ding extended PageRank by integrating both the community impact through co-authorship and their academic impact through citations.

For a co-authorship network, a node represents an author, edges represent the coauthor relation, and the weight of each edge is the number of co-authorships between each author. Through Yan and Ding's modification of the PageRank algorithm, they obtained the formula:

$$PR\_W(p) = (1-d)\frac{CC(p)}{\sum_{j=1}^{N} CC(p_j)} + d\sum_{i=1}^{k}\frac{PR\_W(p_i)}{C(p_i)} \tag{V.1}$$

The main adjustment in this equation is the substitution of $\frac{1}{N}$, where N is the number of nodes in the network, with $\frac{CC(p)}{\sum_{j=1}^{N} CC(p_j)}$, where $CC(p)$ is the number of citations pointing to the author p, and $\sum_{j=1}^{N} CC(p_j)$ is the citation count of all nodes in the network. This weighs the algorithm toward authors with higher citation counts. By adjusting the damping number d, Yan and Ding adjust the weight given to the citation ranking versus the weight given to the coauthorship value.

Since Yan and Ding incorporated citation count into their algorithm, it was not a suitable criterion for evaluating their findings. Instead, Yan and Ding compared the results of their algorithm's rankings of authors to the program committee (PC) membership data for 12 International Society for Scientometrics and Informetrics (ISSI) conferences and by comparing how many had received the Derek de Solla Price Award. Yan and Ding found that the PR_W algorithm was

able to rank award winners at the top of the list accurately, and outperformed both the standard PR and the h-index.

5.3 Contrary Findings

In a 2014 study by Fiala, Šubelj, Žitnik, and Bajec (2015), they took three large datasets of papers in the computer science fields and analyzed them using a variety of different ranking methods. Fiala et al. (2015) compared 12 different methods, including citation count, in-degree, where they only count one citation from each author, and HITS, a method similar to PageRank. Fiala et al. (2015) also compared the standard PageRank algorithm and alterations of the algorithm, using various methods to weigh it, taking into account the number of citations, collaborations, and common publications between authors. Fiala et al. (2015) used a modified version of the PageRank algorithm:

$$PR(u) = \frac{(1-d)}{|V|} + d \sum_{(v,u)\in E} PR(v)\Omega \tag{V.2}$$

where $\Omega = \sigma_{v,k} / \sum_{(v,j)\epsilon E} w_{v,j}$, and

$$\sigma_{v,k} = \frac{w_{v,k}}{(c_{v,k}+1)/(b_{v,k}+1)\sum_{(v,j)\in E} w_{v,j}} \tag{V.3}$$

where V is the set of authors, E is the set of edges, and w, b, and c are coefficients used to produce a weight for each citation. The coefficients are defined where w is the number of citations between authors, c is the number of collaborations between authors, and b is the number of common publications between them. By adjusting the values of those coefficients, Fiala et al. (2015) obtained a variety of algorithms to work with. Fiala et al. (2015) compared the results from their rankings to the editorial board membership of prestigious journals. After analyzing their data sets for each variation, they concluded that the PageRank

method and all of its variations did not outperform citation counts. The method that they found to return the most accurate results was the in-degree citation count, which outperformed not just the PageRank algorithms, but also the normal citation count.

While many of the earlier studies on the application of the PageRank algorithm show potential in its use for citation networks, this study shows it to not be a clear improvement. Many different factors must be taken into account, and what may work for one field of study may not for another. Although there are certain situations where the PageRank algorithm produces results the citation count cannot, the work necessary to use PageRank for citation networks is not insubstantial, especially compared to the normal citation network.

CHAPTER VI

OTHER APPLICATIONS OF PAGERANK

6.1 NFL Rankings

The PageRank algorithm has been applied to other fields as well. Zack, Lamb, and Ball (2012) applied it in an effort to rank NFL teams accurately. Zack et al. (2012) applied what they refer to as the *GEM method*, which uses the margin of victory between two teams to weigh the link between them. If teams play multiple times, then the sum of points won by is weighted on the outgoing link from each victorious team. By using this method, Zack et al. (2012) build the hyperlink matrix:

$$M_{ij} = \begin{cases} 1/\sum_{m=1}^{n}(v_{i_m} - v_{j_m}) & \text{if } v_i \text{ is greater than } v_j \\ 0 & \text{otherwise, or if } n = 0 \end{cases} \quad \text{(VI.1)}$$

Where v is the score from each team in their games, and n is the number of games between teams. Zack et al. (2012) then did the same thing for total yardage, turnovers, and time of possession, with the score rankings noted as GEM 1 and the composite as GEM 2. They compared them using a Kendall rank correlation, defined as $\tau = (n_c - n_d)/(n(n-1)/2)$, where n_c is the number of concordant pairs and n_d is the number of discordant pairs in the rankings. After Zack et al. (2012) removed the least-correlated statistic, they found that they achieved a Kendall rating of $\tau = 0.60$. The method Zack et al. (2012) used had room for variability and could be adjusted for use with other sports as well.

6.2 Food Webs

Allesina and Pascual (2009) used the PageRank algorithm to measure species relative importance for co-extinction. In their 2009 paper, Allesina and Pascual sought to analyze complex ecological networks in an attempt to determine the importance of individual species in the food web. A food web can be created by taking nodes as different species in the environment and the directed edges as the transfer of energy from one species to another, most often by being eaten. One difference between this and the basic PageRank is that importance is flipped, so that a species is important if it points to, or in other words is eaten by, an important species. Food webs are also do not make an irreducible matrix, but a damping factor is unrealistic, since food cannot randomly ”jump” around a food web. Allesina and Pascual (2009) dealt with this by adding a special node, a ”root,” which stood for the food source of all of the primary producers in the food web. Every node also had a link from itself to the root, which signified the intrinsic loss of matter of a species, which builds into detritus and is recycled into the food web. By applying these modifications, the food web becomes irreducible.

For testing, Allesina and Pascual (2009) performed in-silico extinction experiments. Each step Allesina and Pascual (2009) removed a single species and recorded the number of secondary extinctions. There were several algorithms used to determine the species removed. The PageRank algorithm was used to remove the species with the highest PageRank at each step. Allesina and Pascual (2009) also measured the effects of the removal of the species with the highest number of connections, and the removal of species according to their closeness centrality, where nodes are considered highly central if they have a short distance to many nodes. Also measured was the betweenness centrality, where a node has high betweenness if it lies on the shortest path of

many couples of nodes, and removal according to dominators, where one node is dominated by another if every path from the root to itself contains the other, so that if the other goes extinct then it must as well.

Allesina and Pascual (2009) applied these algorithms to 12 published food webs. To compare these results, they programmed a genetic algorithm to find the best possible sequence with an evolutionary search, which has been shown to be effective although computationally expensive. Compared to the genetic algorithm, most of the methods did not perform well. The majority did not match the effectiveness of the genetic algorithm. Of the ones that were comparable, the removal procedure based on dominators only did as well 2/3 of the time, but the PageRank algorithm matched the genetic algorithm 11 out of 12 times.

In an attempt to further improve the PageRank method, Allesina and Pascual (2009) reclassified some of the links in the food web as either ”redundant” or ”functional” based upon their effects on secondary extinctions. By removing the redundant connections, Allesina and Pascual (2009) built a simplified food web that was just as robust as the original. When the PageRank algorithm was tested with this, the results matched the genetic algorithm. With this, Allesina and Pascual (2009) were able to provide an effective algorithm to rank species importance in the food web, and show that it was much more effective that counting the number of connections.

6.3 Protein Networks

Ivan and Gorlmusz (2011) applied the PageRank algorithm to protein interaction networks. Ivan and Gorlmusz (2011) demonstrated it on the metabolic network data of the tuberculosis bacterium and the proteomics analysis of the blood of melanoma patients. Ivan and Gorlmusz (2011) used the PageRank algorithm, stating one of its best attributes being its stability, because the pro-

tein interaction networks contained many false negative and positive interaction edges. Ivan and Gorlmusz (2011) gave the stability estimation of PageRank as:

$$||\mathbf{p} - \hat{\mathbf{p}}||_1 \leq \frac{2(1-c)}{c} \sum_{j \in U} p_j \tag{VI.2}$$

Where the ith coordinate of vector **p** gives the PageRank of vertex i, and vector $\hat{\mathbf{p}}$ gives the PageRank of the vertices after the edges with endpoints in set U are changed. Therefore, if only the edges between the less important nodes are changed, then the effects of the change on the PageRank remain low. This was important because of the often unreliable mapping of less important protein interactions.

Because the PageRank algorithm works with a directed graph, Ivan and Gorlmusz (2011) used it with a metabolic graph, where nodes represent chemical reactions and are connected with a directed edge if one reaction produces a product that is used by another reaction. After computing the PageRank for the metabolic graph of *Mycobacterium tuberculosis*, Ivan and Gorlmusz (2011) were able to identify nodes that were of special interest. These nodes were chemical reactions that had a PageRank that was larger than proportional to their degree, which meant that in a random walk they are hit more often than others with the same network degree. This means that, similar to how Chen et al. (2007) used it to find lesser known but still very important scientific papers, Ivan and Gorlmusz (2011) were able to identify important chemical reactions that did not have a large number of links to other reactions.

Ivan and Gorlmusz (2011) also wanted to investigate protein-protein interaction (PPI) networks, which are undirected graph networks where the nodes are proteins and the edges are interactions between them. Ivan and Gorlmusz (2011) were able to use the personalized PageRank also developed by Brin et al. (1999) to analyze the proteomics data of melanoma patients. By adjusting

the damping factor, Ivan and Gorlmusz (2011) were able to personalize the algorithm for 13 different proteins found in the study. Then Ivan and Gorlmusz (2011) applied the personalized algorithm to the human PPI graph HPRD (Human Protein Reference Database). Ivan and Gorlmusz (2011) found that of the proteins with the highest PageRank, 10 out of the 13 proteins that the algorithm was personalized to appeared in the top 22, and another 10 had clear connections to cancer.

In a 2013 paper, Bánky, Iván, and Grolmusz introduced a new method that would compensate for the weight that smaller degree nodes compared to larger degree nodes, which are sometimes simply removed from the analysis to prevent them from overwhelming the smaller nodes. Bánky et al. (2013) showed that in undirected graphs, the PageRank of a node is proportional to its degree if the personalization vector was proportional. The PageRank with a personalization vector **w** such that:

$$\mathbf{w} = \left(\frac{d(v_1)}{2|E|}, \frac{d(v_2)}{2|E|}, ..., \frac{d(v_n)}{2|E|} \right)^T, \tag{VI.3}$$

Where $d(v_i)$ is the degree of vertex v_i and $|E|$ is the number of edges in the graph, is equal to **w** for undirected graphs. This allowed Bánky et al. (2013) to factor out the relative degreeness from each vertex.

In order to use a similar method for directed graphs, Bánky et al. (2013) defined the personalization vector **w**:

$$\mathbf{w} = \left(\frac{d_r(v_1)}{|E|}, \frac{d_r(v_2)}{|E|}, ..., \frac{d_r(v_n)}{|E|} \right)^T, \tag{VI.4}$$

Where $d_r(v)$ is the number of directed edges pointed at vertex v. Then they defined a "revitalized personalized PageRank", **rPPR**, where:

$$\mathbf{rPPR}(\mathbf{v}) = \frac{\mathbf{PageRank(v)}}{\mathbf{d_r}(\mathbf{v})} \tag{VI.5}$$

Bánky et al. (2013) applied this method to several different metabolic networks, *Mycobacterium tuberculosis*, *Plasmodium falciparum*, and the MRSA *Staphylococcus aureus* SAA strain. By only looking at the **rPPR**, and not any of the known properties of the proteins aside from that, they were able to identify many protein targets that are also known to have proven biological interest. The results gave higher scores to nodes with relatively high PageRanks compared to their degrees. This can identify nodes of high importance and may be promising new drug targets that are not hubs of the network. Using these results, they hope to be able to identify new important targets for further investigation. With this method, Bánky et al. (2013) were confident that they would be able to use the PageRank algorithm to help find low-degree nodes with high intrinsic metabolic functionality. By using the personalized vector to factor out nodes with high degrees, Bánky et al. (2013) were able to find the non-hub nodes corresponding to essential reactions.

CHAPTER VII

CONCLUSION

The PageRank algorithm had a significant impact on the Internet, and that impact has extended to other fields as well. It has been shown to improve ranking methods for many areas, such as author impacts, protein networks, and food webs, and has the potential to improve many others. The PageRank algorithm is a versatile enough method to be applied to many different situations, and can be modified for more complex situations.

BIBLIOGRAPHY

Allesina, S., & Pascual, M. (2009). Googling Food Webs: Can an Eigenvector Measure Species' Importance for Coextinctions? *PLOS Computational Biology, 5*(9). DOI: 10.1371/journal.pcbi.1000494

Bánky, D., Iván, G., & Grolmusz, V. (2013). Equal Opportunity for Low-Degree Network Nodes: A PageRank-Based Method for Protein Target Identification in Metabolic Graphs. *PLoS ONE, 8*(1). DOI: 10.1371/journal.pone.0054204

Brin, S.,& Page, L. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems 3*(1-7), 107–117. DOI: 10.1016/S0169-7552(98)00110-X

Brin, S., Page, L., Motwani, R., & Winograd, T. (1999). The PageRank Citation Ranking: Bringing Order to the Web. *Stanford University Technical Report*. Retrieved from http://ilpubs.stanford.edu:8090/422/

Chen, P., Xie, H., Maslov, S., & Redner, S. (2007). Finding scientific gems with Google's PageRank algorithm. *Journal of Informetrics, 1*(1), 8-15. DOI: 10.1016/j.joi.2006.06.001

eBizMBA. (2017). *Top 15 Most Popular Search Engines June 2017*. Retrieved February 06, 2018, from http://www.ebizmba.com/articles/search-engines

Euler, L. (1759). Solutio problematis ad geometriam situs pertinentis. *Mmoires de l'Acadmie des sciences de Berlin, 8*, 128–140

Fiala, D., Šubelj, L., Žitnik, S., & Bajec, M. (2015). Do PageRank-based author rankings outperform simple citation counts? *Journal of Informetrics, 9*(2), 334-348. DOI: 10.1016/j.joi.2015.02.008

Garfield, E. (1972). Citation analysis as a tool in journal evaluation. *Science, 178*, 471–479.

Google. (n.d.). *How we started and where we are today.* Retrieved February 06, 2018, from https://www.google.com/about/our-story/

Google. (n.d.). *Facts about Google and Competition*. Retrieved February 06, 2018, from https://web.archive.org/web/20130702063520/https://www.google.com/competition/howgooglesearchworks.html

Haveliwala, T., & Kamvar, S. (2003). The Second Eigenvalue of the Google Matrix. *Stanford University Technical Report*. Retrieved from: http://ilpubs.stanford.edu:8090/582

Hirsch, J. (2005). An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences, 102* (46), 16569-16572; DOI: 10.1073/pnas.0507655102

Iván, G., & Grolmusz, V. (2011). When the Web meets the cell: using personalized PageRank for analyzing protein interaction networks. *Bioinformatics, 27*(3), 405-407. DOI: 10.1093/bioinformatics/btq680

Meyer, C. (Ed.) (2000). *Matrix Analysis and Applied Linear Algebra*. Philadelphia, PA: Soc. for Industrial and Applied Math.

Sangers, A. & van Gijzen, M. (2015). The eigenvectors corresponding to the second eigenvalue of the Google matrix and their relation to link spamming. *Journal of Computational and Applied Mathematics 277*, 192-201

Wall, A. (2017). *History of Search Engines: From 1945 to Google Today*. Retrieved from http://www.searchenginehistory.com/

Wills, R. S. (2007). *When Rank Trumps Precision: Using the Power Method to Compute Google's PageRank* (Doctoral Dissertation). Retrieved from https://repository.lib.ncsu.edu/handle/1840.16/4953

Yan, E., & Ding, Y. (2010). Discovering author impact: A PageRank perspective. *Information Processing and Management, 47*, 125-134. DOI: 10.1016/j.ipm.2010.05.002.

Zack, L., Lamb, R., & Ball, S. (2012). An application of Google's PageRank to NFL Rankings. *Involve, 5*(4), 463-471.