Application of Eigenvalues & Eigenvectors: PageRank Algorithm

MATH301: Linear Algebra

Computer Science Department

Nile University

Giza, Egypt

Under supervision:

Dr. Marwa Aref Sorour

Application of Eigenvalues & Eigenvectors: PageRank Algorithm

Contents

Application of Eigenvalues & Eigenvectors: PageRank Algorithm

Abstract

This study aims to show how PageRank algorithm, which is the basis for most search engines work, how it evolved with deep dive into mathematical equations and Advanced algorithms, and how it relates to linear Algebraic concepts like Eigenvalues & Eigenvectors.

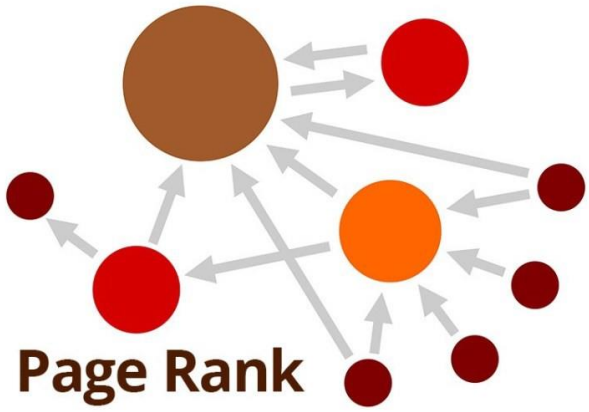

Figure 1

Keywords—Web Mining, Web Structure, Web Graph, Link Analysis, PageRank, Weighted PageRank, Distance Rank.

Introduction

In the past, people used to use content-based retrieval technique to determine the relevance of a keyword to a document by counting number of occurrences of the keyword in the document. This algorithm proved to be handy in normal text retrieval but in the world of World Wide Web (WWW) where we have billions of pages; the algorithm is extremely inefficient for searching. Let's imagine that we put the keyword "volatile" as many as we can in a web page the search engine would recommend this page which is misleading because we can manipulate the search results for our favour this way. That led people to start thinking about new algorithms for searching network as big as World Wide Web also, to make sure that algorithms are unbiased and time efficient. One of the most important techniques is called PageRank which Google firstly invented. In short, this algorithm rates the importance of webpages using hyperlink analysis [1] then returns to the user search results from high importance to low importance. Unlike content-based information retrieval, this algorithm does not consider content of the webpage, instead, it considers inter-connectivity of the web. In other words, in searching process connection between webpages is more important than its content. And this algorithm proved to be very successful judging by the success of Google search engine. That is why it is called the trillion-dollar algorithm.

During the last two decades, Google has been the most widely utilized search engine and still it is. This because they use techniques that get more accurate results comparing to other search engines like Bing or Duckduckgo. PageRank was a research project of Larry Page (thus the name PageRank [2]) and Sergey Brin in 1996 at Stanford University [3]. Sergey believed in link popularity method for searching the Web which means the webpage of many webpages link to it should be ranked higher as a search result than its peers [4]. This method was co-authored by Terry Winograd and Rajeev Motwani [5]. In 1998, the initial prototype of Google search engine was made, and PageRank algorithm was published. Shortly after, Lawrence and

Sergey founded Google Inc [6]. Since then, PageRank is the basis for Google's web search tools [7].Google's PageRank research paper about PageRank algorithm in 1998 cited three important resources. Firstly, Eugene Garfield, he early developed citation analysis in the 1950s at University of Pennsylvania. Secondly, Massimo Marchiori, he developed Hyper Search at the University of Padua. Thirdly, Jon Kleinberg, he published a paper about HITS in the same year 1998. Finally, Google original paper depended on those three research papers [6]. Notably, another small search engine named RankDex made by Robin Li from IDD Information Services in 1996 uses similar work of Google's paper of 1998 [8]. Though Li's RankDex is older than Google it wasn't patented until 1999 [9] to be used in the famous Google competitor in China today Baidu [10]. However, U.S. patents reference Robin Li's work in Lawrence Page's Google search methods.

Literature Review

Background:

In order to find pages that match a user's query, many modern search engines employ a two-step procedure. Traditional text processing is used in the first step to locate all documents that contain the search terms or are related to the search terms semantically. A look-up into an inverted file, the use of vector space, or a query expander that makes use of a thesaurus are all possible ways to accomplish this. This initial step may return thousands of pages relevant to the query due to the vast size of the web. Many search engines sort this list according to some sort of ranking criterion to make it easier for a user to manage. Utilizing the extra information the web naturally contains due to its hyperlinking structure is a common method for producing this ranking. Link analysis is now the method for ranking, as a result. The Google search engine's PageRank ranking system is one popular and effective link-based ranking system. On the East Coast, a young scientist named Jon Kleinberg was working on a web search engine project called HITS as an assistant professor at Cornell University. His algorithm, which was novel at the time because most search engines only used text content to return relevant documents, improved search engine results by utilizing the web's hyperlink structure. In January 1998, he presented his work [Kleinberg 99], which he had started a year earlier at IBM, at the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, which was held in San Francisco, California. Nearby, two PhD candidates at Stanford University were spending late nights working on a project similar to PageRank. Working together on their web search engine since 1995, Larry Page and Sergey Brin were both computer science students. Things were really picking up speed for these two scientists by 1998. For the fledgling company, which later grew to be the enormous Google, they were using their dorm rooms as offices. By August 1998, Page and Brin had both taken a leave of absence from Stanford to concentrate on their expanding company, since that eventful year, PageRank has become the most popular link analysis model, in part because of its query independence, virtual immunity to spamming, and Google's enormous commercial success. Contrary to Brin and Page, Kleinberg did not attempt to turn HITS into a business because he was already establishing himself as an inventive academic. Later businesspeople did, though; the search engine Teoma bases its technology on an extension of the HITS algorithm [Sherman 02]. As a side note, Google kept Brin and Page busy and prosperous enough to continue their leave from Stanford. Following their well-cited

original 1998 paper, this study explores the numerous improvements that have been made to the fundamental PageRank model, allowing readers to gain a deeper understanding of PageRank. We should note that the techniques described in this paper were developed by Brin and Page and later incorporated into their search engine Google. Since the information from the 1998 papers [Brin et al. Page 98, Brin et al., 98a 98b]. We do, however, know that PageRank continues to be "the brains of [Google's] software dot. and still serves as the foundation for all of [their] web search tools," according to information directly taken from the Google website at http://www. google . com/technology/index . html 3.

How it works:

The Basic PageRank Model The original Brin and Page model for PageRank constructs a Markov chain using the web's hyperlink structure and a simple1 transition probability matrix P. The existence of the long-run stationary vector T, also known as the PageRank vector, is ensured by the irreducibility of the chain. It is common knowledge that the stationary vector will be reached when the power method is applied to a primitive matrix. The size of the subdominant eigenvalue of the transition rate matrix also affects how quickly the power method converges [Stewart 94]. If a matrix's graph demonstrates that each node can be reached from every other node, then the matrix is irreducible. When there is only one eigenvalue on the spectral circle of a nonnegative, irreducible matrix, it is referred to as primitive. Aperiodic chains are irreducible Markov chains with a simple transition matrix. A straightforward test for primitivity was developed by Frobenius: the matrix A 0 is primitive if and only if A m > 0 for some m > 0 [Meyer 00]. The power method applied to a matrix can be tested to see if it will converge. The Markov Model of the Web We started by demonstrating how Brin and Page, the creators of the PageRank model, forced the transition probability matrix, which is derived from the web's hyperlink structure, to be random and ad hoc. A directed graph can be used to represent the web's hyperlink structure. Websites' goal in general is to provide users with information and needs that will help them succeed in the highly competitive world. Analyzing user behavior can help you understand how they interact with your site or application. Web mining is used to discover the content of the Web, the users' behavior in the past, and the webpages that the users want to view in the future. Web mining is a form of search engine optimization (SEO) that involves finding and analyzing web pages and pages of related websites for content that can be used to improve the visibility of a websites. Web content management (WCM) helps you find useful information from web content. WCM discovers relationships between web pages by analyzing web page structure. WCM ascertains user profiles and the users' behavior on the web logfile. Many researchers have found that WCM is effective method to improve performance. Links analysis and WCM can classify web pages into different types, based on their hyperlink structure. This allows for more accurate pattern recognition, as well as better understanding of related web pages. focuses mainly on the structure within a document (the inner document level), while it tries to discover the link structure of the hyperlinks between documents (the interdocument level). The number of inlinks (links to a page) and the number of outlinks (links from a page) are important information in web mining. This is because a popular website is often mentioned by other websites and because an "important" website has a high number of links. Link analysis is seen as an effective approach for web mining. This paper discusses how PageRank can be used to improve the quality of web pages. A new weighted pageRank algorithm is provided. The purpose of this paper is to introduce the reader to the concept of stress, and to discuss the effects

of stress on the body. The different types of stress are also discussed, and the impact of stress on the individual is also discussed. A brief overview of web structure mining is given, including the PageRank algorithm, which is a widely used algorithm in web ranking. Since the growth of the Web, it has become increasingly difficult to provide high-quality pages that are relevant to the users based on their queries. The web pages that are not self-descriptive can be because the author didn't spend enough time writing them, or because some of the links on the page are just for convenience, not meant to be considered a lead to the page. There is no easy way to find relevant pages through a search engine relying on web contents or using hyperlink information. Some algorithms have been proposed to address the problems mentioned above. There are PageRank and Hypertext Induced Topic Selection (HITS) algorithms. Pages that are found in a web document are given a rank based on how often they are found. The site evaluation measures the importance of the pages by analysing the links. PageRank is a ranking algorithm developed by Google and named after Larry Page. PageRank ranks pages based on their web structure. The Google search engine retrieves pages that are relevant to the given query. Then it adjusts the results to provide more important pages at the top of the page list.

The PageRank algorithm is a ranking algorithm that looks at how important pages are connected to each other. The more important links a website has, the more important its links are. Therefore, PageRank takes the backlinks into account and propagates the ranking through links: a page has a high rank if the sum of the ranks of its backlinks is high.

Methodology:

Markov chains:

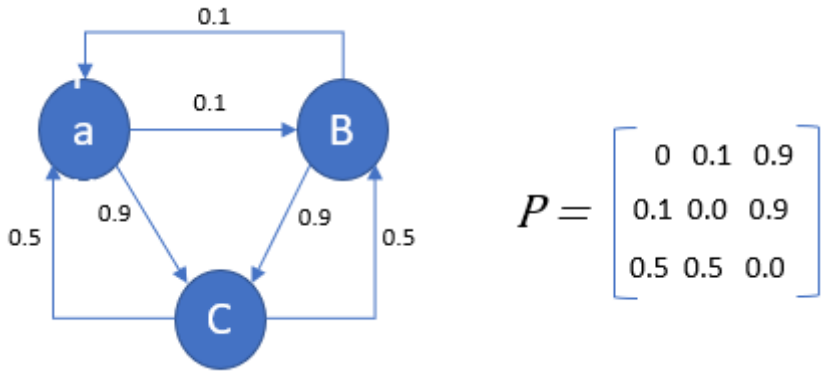

Figure 2

Markov chain in the context of web pages is defined by a few specific properties in the context of a web network individual web pages correspond to states of the Markov chain and going from state to state involves a transition probability in this particular network any user can go from state 3 to state 1 with transition probability 0.5 if a user is at state 2 they have a 100 chance of transitioning to state 3. in the context of web networks it's common for transition probabilities to be assumed based on the number of outgoing links for example if a web page has three outgoing links we assume the user is equally likely to transition to each of the states with probability 1 /3. it is possible to also alter these probabilities based on some prior knowledge or data so if we change the distribution as follows it would assume that users are

much more likely to click on an outgoing link to web page 2. now the problem we care about today is ranking the importance of states in the simplified real world use case here is to imagine a user got four web pages from a search query these web pages will often reference each other so they form a Markov chain how should we rank the web pages to show an end user the first question here is how we should define the importance of a state in a Markov chain if we have a variety of users surfing web pages what might be a good way to rank the relative importance of each web page while one natural way to think about the importance of web pages is to imagine a single user continuously going from page to page according to the Markov chain model at each point in time the user is going to be at a particular state of the Markov chain and we could maintain a counter for the number of times they visit a state over a long period of time the states that end up being visited more often can naturally be thought of as having a higher rank in fact rather than counts we can think of this long-term behaviour as a distribution on the proportion of time a single user spends in each state in this particular Markov chain if we run this experiment for a long period of time the distribution stabilizes and this distribution is formally called a here the stationary distribution indicates that the most important state is state 2 where the interpretation of the probability is that a random user who serves this network for a long period of time will spend about 26 percent of their time at state 2. what we're going to dive into in this video is how exactly might we find a stationary distribution of a Markov in what conditions do these stationary distributions even exist how do we efficiently compute them to get into how we calculate stationary distributions we have to first understand how we can model Markov chains mathematically at its core a markup chain is a random process involving steps in time a user might start at some initial state and at each step according to the model they randomly transition to one of the possible next locations at every step in time there's a probability of a user being in a particular state a natural way to model the state distribution at a particular step n in a Markov chain is with a vector of probabilities where the sum of the elements in the vector must equal one usually Markov chains are initialized with the initial state distribution at step zero it's pretty common for the initial distribution across states to be assumed as uniform which intuitively just means we expect the user to start on any web page with equal probability one of the key aspects of analysing Markov chains is understanding how the distribution across states change at each step or point in time this leads to a discussion on how we model transitions in a Markov chain. what is nice about Markov chains is we can find the probabilities of landing in each state at the next step fairly easily it's just a matrix vector product of the probabilities at the previous state times the transition matrix. the problem of finding the stationary distribution can be formalized a stationary distribution is now the next natural question of course is how we calculate the stationary distribution with Markov chains.

Eigenvectors:

Eigenvector of a square matrix is defined as a non-vector in which when a given matrix is multiplied, it is equal to a scalar multiple of that vector. Let us suppose that A is an n x n square matrix, and if v be a non-zero vector, then the product of matrix A, and vector v is defined as the product of a scalar quantity λ and the given vector, such that:

$$Av = \lambda v \quad (1)$$

Where v = Eigenvector and λ be the scalar quantity that is termed as eigenvalue associated with given matrix A

Application of Eigenvalues & Eigenvectors: PageRank Algorithm

Eigenvalue of Matrix:

Eigenvalues are generally associated with eigenvectors in Linear algebra. Both of these terms are used in the interpretation of linear transformations. As we know that, eigenvalues are the particular set of scalar values related to linear equations, most probably in the matrix equations.

To define eigenvalues, first, we have to determine eigenvectors. Almost all vectors change their direction when they are multiplied by A. Some rare vectors say x is in the same direction as Ax. These are the "eigenvectors". Multiply an eigenvector by A, and the vector Ax is the number time of the original x. The basic equation is given by:

$$Ax = \lambda x \quad (2)$$

Here, the number λ is an eigenvalue of matrix A.

The basic idea of PageRank is that the importance of web page depends on the pages that link to it. On the one hand, if there are many web pages linking to page (u), then we consider page (u) to be important on the web. On the other hand, if the page (u) has only a few pages linking to it, but those pages are authoritative ones, we also consider page (u) to be an important web page. In the first case, page (u) accumulates importance by massive collection of its incoming links. In the second case, page (u′s) importance is transferred from those linking to it. If we use r(u) to denote the PageRank score of pages (u_i), then the above description could be expressed as the following formula:

$$r(u_i) = \sum_{uj \in B(ui)} \frac{r(uj)}{Nj} \quad (3)$$

where $B(u_i)$ is the set of web pages that points to (u_i) (i.e. the set of backward links) and Nj is the number of outgoing links on page (uj). The matrix representation of the simplified PageRank Algorithm can be written as:

$$R = AR \quad (4)$$

where $R = [r(u1), r(u2), \cdots, r(u_N)]^T$ and the terms of the matrix A are usually,

$$aij == \begin{cases} \frac{1}{Nj} & \text{if page (uj) links to page(ui)} \\ 0 & \text{otherwise} \end{cases}$$

On condition that A is left a stochastic matrix, the iterative computation of R represents the evolution of a Markov Chain and the solution to R is the steady state probability of the Markov Chain. The following power method is generally used to solve the problem:

$$R_{m+1} = AR_m \quad (5)$$

Application of Eigenvalues & Eigenvectors: PageRank Algorithm

Results:

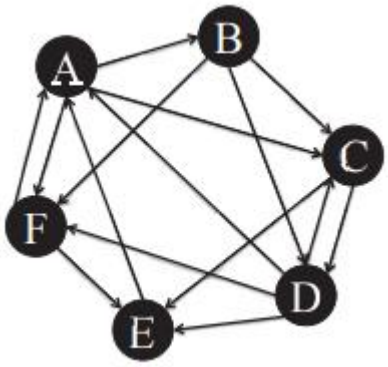

Figure 3: A tiny web graph with 6 nodes

To illustrate, consider the directed graph shown in Fig. 3 based on a tiny web. Based on the definition transformation matrix A, we can easily write it as

$$A = \begin{bmatrix} 0 & 0 & 0 & 1/4 & 1 & 1/3 \\ 1/3 & 0 & 0 & 0 & 0 & 1/3 \\ 1/3 & 1/4 & 0 & 1/4 & 0 & 0 \\ 0 & 1/4 & 1/2 & 0 & 0 & 0 \\ 0 & 1/4 & 1/2 & 1/4 & 0 & 1/3 \\ 1/3 & 1/4 & 0 & 1/4 & 0 & 0 \end{bmatrix}$$

If we start from uniform distribution, the initial PageRank of each node is 16. Let R denote the initial PageRank score vector, with all entries equal to 16. We iteratively compute the new PageRank score by multiplying the matrix A to the left. Numerical computation gives:

$$R = \begin{pmatrix} 0.167 \\ 0.167 \\ 0.167 \\ 0.167 \\ 0.167 \\ 0.167 \end{pmatrix}, \quad PR = \begin{pmatrix} 0.264 \\ 0.111 \\ 0.139 \\ 0.125 \\ 0.222 \\ 0.139 \end{pmatrix}, \quad P^2R = \begin{pmatrix} 0.300 \\ 0.134 \\ 0.147 \\ 0.097 \\ 0.175 \\ 0.147 \end{pmatrix}, \ldots\ldots, \quad P^{13}R = \begin{pmatrix} 0.265 \\ 0.138 \\ 0.150 \\ 0.110 \\ 0.187 \\ 0.150 \end{pmatrix}$$

We observe that the sequence of iterations R, PR, P^2R, $\cdots$, P^nR tends to converge e to the

value $R* = \begin{pmatrix} 0.265 \\ 0.138 \\ 0.150 \\ 0.110 \\ 0.187 \\ 0.150 \end{pmatrix}$, which is the solution to the PageRank of all web pages.

The convergence of the above method is guaranteed as long as the matrix A is a left stochastic matrix.

References

[1] Henzinger, M.R.," Hyperlink analysis for the Web",Internet Computing, IEEE, vol.5, no.1, pp.45-50, Jan/Feb 2001

[2] David Vise and Mark Malseed (2005). The Google Story. p. 37.

[3] Raphael Phan Chung Wei (2002-05-16). New Straits Times.

[4] Page, LawrenceandBrin, SergeyandMotwani, RajeevandWinograd, Terry(1999)The PageRank Citation Ranking: Bringing Order to the Web.Technical Report. Stanford InfoLab.

[5] 187-page study from Graz University, Austria, includes the note that also human brains are used when determining the page rank in Google

[6] Brin, S.; Page, L. (1998)." The anatomy of a large-scale hypertextual Web search engine". Computer Networks and ISDN Systems 30: 107117

[7] " Google Technology". Google.com. Retrieved 2011-05-27.

[8] Li, Yanhong (August 6, 2002). "Toward a qualitative search engine". Internet Computing, IEEE (IEEE Computer Society) 2 (4): 2429

[9] USPTO, "Hypertext Document Retrieval System and Method", U.S. Patent number: 5920859, Inventor: Yanhong Li, Filing date: Feb 5, 1997, Issue date: Jul 6, 1999

[10] Greenberg, Andy, "The Man Who's Beating Google", Forbes magazine, October05, 2009

[11] Bri, P. Zhou, G. Pinski, H. Herbertz, A. Anderson, Anon., J. Bollen, P. Bonacich, D. E. Chubin, R. Cohen, R. Dalpé, E. Garfield, W. Glänzel, J. C. Guan, and T. He, "Bringing PageRank to the citation analysis," Information Processing & Management, 20-Aug-2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457307001203. [Accessed: 21-Nov-2022].

[12] J. Tague-Sutcliffe, A. Sidiropoulos, G. Pinski, N. Ma, X. Liu, L. Egghe, R. P. Dellavalle, P. Chen, S. Brin, J. Bar-Ilan, J. Bollen, R. S. Burt, Y. Ding, and D. Fiala, "Discovering author impact: A pagerank perspective," Information Processing & Management, 17-Jun-2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457310000488. [Accessed: 21-Nov-2022].

[13] "Adaptive methods for the computation of PageRank," Linear Algebra and its Applications, 16-Apr-2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0024379504000023. [Accessed: 21-Nov-2022].

[14] "An analytical comparison of approaches to personalizing pagerank ..." [Online]. Available: http://dbpubs.stanford.edu/pub/2003-35. [Accessed: 21-Nov-2022].

[15] “Deeper inside PageRank,” Taylor & Francis. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/15427951.2004.10129091. [Accessed: 21-Nov-2022].