IMPLEMENTATION OF THE PAGERANK ALGORITHM

JUSTIN J. WANG*

October 11, 2016

Abstract.
Centrality measures are useful tools in network analysis since they quantify a node's importance in a graph modeling a network. The definition of importance varies based on context, so choosing an appropriate centrality measure is essential. For larger networks in particular, we need to be able to find the most important nodes among hundreds or thousands of them. PageRank centrality assigns a measure to each node based on the number of incoming edges and their weights. It is a variant of the eigenvector centrality measure. The algorithm is defined recursively; as a consequence, a node that has a lot of incoming edges from nodes with high PageRank has high PageRank itself.

The method was designed to rate web pages objectively and mechanically, effectively measuring the human interest and attention devoted to them. To test the utility of PageRank for search, Brin and Page [5] built a web search engine called Google. In this paper, we apply the measure on various directed graphs from http://snap.stanford.edu/data/ - Stanford large network dataset collection. We implement the PageRank centrality measure to order the nodes, and to see how useful it is when applied to a road network, a Facebook social network, a Google webgraph, and Amazon co-purchasing network. [4] We describe the algorithm and implement it in python.

Key words. PageRank algorithm, eigenvector centrality, directed networks, graph theory

1. Introduction. The concept of centrality is investigated in order to answer the question, "What are the most important nodes in a network?" Centrality is an important measure used in social, biological, communication, and transportation networks since it helps analyze the relative structural prominence of nodes in the network. Especially in social network analysis it can measure the most influential or the most connected person depending on the context of the network. PageRank was designed to rank the crawlable Web, so that web pages (nodes) with high numbers of backlinks (inedges) are deemed more important. It is more sophisticated than simply counting number of links, which would correspond to the node degree.

2. Centrality Measures. Throughout the paper, we consider only the graphs $G = (V, E)$ that are simple, directed, and connected and that have $n \geq 2$ nodes.

2.1. Mathematical Formulation of Google PageRank. Given a list of edges, our first step is to construct the adjacency matrix such that each cell represents the proportion of outflow from a node. Consider the graph of a network $N = (G; c)$ with positive edge weights $c : E \to \mathbb{R}_{>0}$ indicating the proportion of the outflow.

To illustrate, given the following graph 2.1 :

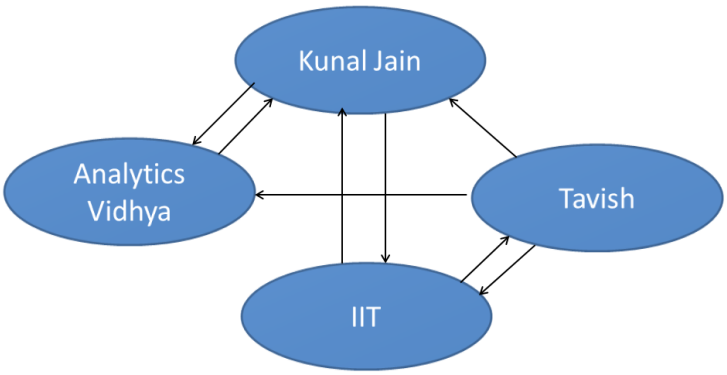

FIG. 2.1. *Directed Graph [6]*

DEFINITION 1. *We define the adjacency matrix $A : V \to V$ is based on the hyperlink structure of the graph. Specifically,*

$$A_{i,j} = \frac{1}{d_i}, \textit{if page } i \textit{ links to page } j \textit{ [1]} \tag{2.1}$$

[1]MS Data Science program, Galvanize, Inc., 543 Howard St., San Francisco, CA 94105, email: justwjr@ucla.edu
[2]galvanizeU, 44 Tehama St., San Francisco, CA 94105, linkedin.com/in/justw; github.com/justwjr

where d_i is the total number of unique outgoing links from page i.

Here is the corresponding adjacency matrix A for Figure 2.1 above:

		FROM			
		KJ	TS	IIT	AV
TO	KJ	0.00	0.33	0.50	1.00
	TS	0.00	0.00	0.50	0.00
	IIT	0.50	0.33	0.00	0.00
	AV	0.50	0.33	0.00	0.00

FIG. 2.2. *Normalized Adjacency Matrix A [6]*

For instance, Tavish (TS) has 3 outgoing links which makes each proportion as 1/3.
Since,

$$\sum_j \vec{v}_{ij} = 1 \quad \text{for all i} \tag{2.2}$$

and all entries are greater than or equal to 0, A is a stochastic matrix representing a transition matrix in a Markov-chain.
First, It is trivial to see that 1 is an eigenvalue of A. In fact, the principal eigenvalue of any stochastic matrix is 1, which can be proven by contradiction or using the Gershgorin circle theorem. The eigenvector $\boldsymbol{\pi}$ to the eigenvalue 1 is called the stable equilibrium distribution of A. It is also called Perron-Frobenius eigenvector. [3]
Next, we construct the Markov-chain and obtain the stable equilibrium vector, $\boldsymbol{\pi}$, which is the probability vector representing the total time spent on each node. This can be broken down into the equation:

$$A * \boldsymbol{\pi} = \boldsymbol{\pi} \tag{2.3}$$

where A is the transition matrix, and $\boldsymbol{\pi}$ is the probability of being at each transitory state.

2.2. Teleportation adjustments. Suppose we have a network $N = (G; c)$ of only 2 nodes A and B, with positive edge weights $c : E \to \mathbb{R}_{>0}$ indicating the proportion of the outflow such that A has a link to B but B has no external links.
In such cases, if you try solving the matrix, you will get a zero matrix. This looks unreasonable as B looks to be more important than A. But, our algorithm still gives same importance for both. [6]

To account for the end states in the Markov-chain, we introduce the concept of teleportation. To describe the process in terms of a webgraph, we introduce a constant probability, $\boldsymbol{\epsilon}$ to each page to compensate for instances where a user teleports from one page to another without any link. [6]

LEMMA 1. *As a consequence, we modify our above equation with a constant $\boldsymbol{\epsilon}$:*

$$\boldsymbol{\pi} = \boldsymbol{\epsilon} * \vec{1} + (1 - \boldsymbol{\epsilon})A^T\boldsymbol{\pi} \tag{2.4}$$

where $\vec{1}$ is a column vector of all 1s, and $\boldsymbol{\pi}[i]$ is the rank of the ith page.

One approach to solving the above equation is to start with a value of $\boldsymbol{\pi}$, where each component is 1/n (where n is the number of nodes) and then perform the eigenvalue/eigenvector solver on the matrix:

$$A' = \boldsymbol{\epsilon} * \vec{1} + (1 - \boldsymbol{\epsilon})A^T \tag{2.5}$$

The pseudocode for the algorithm is described below. Also, I implemented 3 different iterative power methods, and an additional built in eigenvector solver to solving the principal eigenvector, along with their error and convergence rates, included in Appendix A [7].

3. History. In 1976 by Pinski and Narin suggested the eigenvalue problem while working on scientometrics ranking scientific journals. Page and Brin developed PageRank at Stanford in 1996 as part of search engine research. Brin had the idea of "link popularity". They and two others published a paper in 1998, cited below. [5]

In practice, Google does not rely solely on PageRank. On October 15, 2009, a Google employee confirmed that the company had removed PageRank from its Webmaster Tools section, saying that "We've been telling people for

Algorithm 1 NumPy LAPACK Eigenvector Solver Method for PageRank [2]

Require: adjacency matrix of directed graph network $G = (V; E)$, constant teleportation probability $0 < \epsilon < 1$
  return PageRank approximation $pr_c : V \to \mathbb{R}_{\geq 0}$, as a dictionary of nodes with PageRank as value
  n = number of rows in G
  initialize dangling weights vector of size n where each element is 1/n
  for $i = 1, ..., n$ **do**
    if row sums to 0, append row index to array of dangling nodes
  end for
  for $x \in$ *dangling nodes* **do**
    x^{th} column of M = dangling weights vector
  end for
  for *each column* $\in G$ **do**
    for each element in column **do**
      element /= sum(column)
    end for
  end for
  Use eigenvector solver on $\epsilon * M + (1 - \epsilon) * danglingweights$ to obtain principal eigenvector
  sort eigenvalues to obtain index of principal eigenvector
  return normalized principal eigenvector

a long time that they shouldn't focus on PageRank so much. On April 15, 2016 Google has officially shut down their Google Toolbar PageRank Data to public. Google had told about this earlier that they would be removing the PageRank score from the google toolbar months before in advance. As of writing, the top 3 metrics are Links, Content,, and RankBrain. PageRank is deprecated.[8]

4. Current Uses. Besides its originally designed purpose to rank websites, PageRank is now regularly used in bibliometrics, social and information network analysis, and for link prediction and recommendation. It's even used for systems analysis of road networks, as well as biology, chemistry, neuroscience, and physics.

Twitter uses Personal PageRank to present users with other accounts they may wish to follow.

Another is to rank academic doctoral programs based on their records of placing their graduates in faculty positions. In PageRank terms, academic departments link to each other by hiring their faculty from each other (and from themselves).

In a road network, it's used to rank spaces or streets to predict how many people (pedestrians or vehicles) come to the individual spaces or streets. Pagerank has recently been used to quantify the scientific impact of researchers, through the citation and collaboration networks [8]

5. Summary and conclusion. The applications of PageRank are not limited to webpages, rather, the centrailty measure can be applied to any network that can be represented as a directed graph of nodes and edges. In particular, the algorithm can be applied to social networks to quantify how important a person is compared to everyone else. Similarly, we can run the process through data of a road network or Amazon's co-purchasing network.

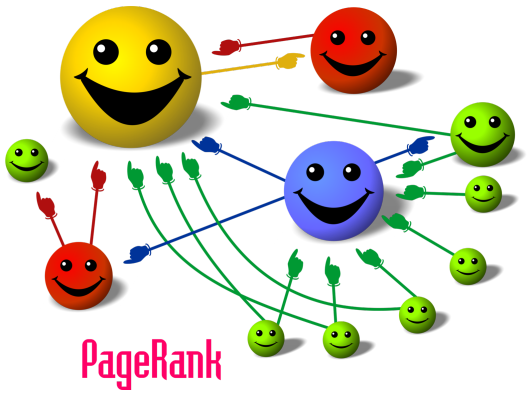

FIG. 5.1. *Size Proportional Smilies[8]*

REFERENCES

[1] Adnan Aziz, *Elements of programming interviews : 300 questions and solutions*, CreateSpace Independent Publishing Platform, Place of publication not identified, 2012.

[2] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart, *Exploring network structure, dynamics, and function using NetworkX*, in Proceedings of the 7th Python in Science Conference (SciPy2008), Pasadena, CA USA, Aug. 2008, pp. 11–15.

[3] Oliver Knill, *Markov matrices*. www.math.harvard.edu/~knill/teaching/math19b_2011/handouts/lecture33.pdf/, Apr. 2011.

[4] Jure Leskovec and Andrej Krevl, *SNAP Datasets: Stanford large network dataset collection*. http://snap.stanford.edu/data, June 2014.

[5] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd, *The pagerank citation ranking: Bringing order to the web.*, Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[6] Tavish Srivastava, *Pagerank explained in simple terms!* https://www.analyticsvidhya.com/blog/2015/04/pagerank-explained-simple/, Apr. 2015.

[7] Justin J. Wang, *Eigfinder power methods: Development, implementation, and analysis*. https://github.com/justwjr/Numerical-Analysis, May 2014.

[8] Wikipedia, *PageRank — Wikipedia, the free encyclopedia*. http://en.wikipedia.org/w/index.php?title=PageRank&oldid=741332093, 2016. [Online; accessed 11-October-2016].