Linear Algebra and Matrix Theory
MATH 3100

Marquette University

Google's Page Rank Algorithm

Anna Aiuppa
Peter Dobbs
Morgan Hackelberg
Kelly Nantais

The Google PageRank algorithm is based on the following premise: a website is important if other important websites link to it. To create this thesis Sergey Brin and Larry Page, who later founded Google, made two assumptions. First, a hyperlink is a recommendation of a website. Second, a recommendation is more valuable if the recommender is selective. It follows that the more important a page P_i, the higher its PageRank, r(P_i), which is defined as:

$$r(P_i) = \sum_{P_j \in B_{P_i}} \frac{r(P_j)}{|P_j|} \tag{1}$$

where P_i represents page i = 1, 2, ..., n, B_{P_i} is the set of pages that link to P_i, and $|P_j|$ is the number of sites that P_j points into. These ranks fill the columns of the PageRank vector, defined as $\boldsymbol{\pi}^T$ = (r(P_1),r(P_2),...,r(P_n)), where n is the number of pages being ranked. To find the successive rank of a page, the calculations begin with an estimate of each page, $r_o(P_i)$, and the PageRank is updated iteratively with the following:

$$r_{k+1}(P_i) = \sum_{P_j \in B_{P_i}} \frac{r_k(P_j)}{|P_j|} \tag{2}$$

The hyperlink matrix **H** has elements $h_{ij} = \frac{1}{|P_i|}$ for each page P_j which is linked to by page P_i and $h_{ij} = 0$ otherwise. If each row in **H** sums to either 1 or 0, that matrix is called *substochastic*. This case results in page sinks, if a page has no link to other pages (when a row sums to 0). To avoid PageRank sink due to a row of zeros, replace each element in that row with $1/n$ where n is the number of pages in the web.

With what is defined so far, there still exists a possibility for cycling between states. To avoid this, Google founders introduced the idea of the *random surfer*. According to this concept, there is some probability α ($0 \leq \alpha \leq 1$) that the web surfer moves following a hyperlink. There is also the probability that a web surfer moves by directly linking to one of the other n pages directly, represented by 1-α. With this concept and the hyperlink matrix **H**, the Google PageRank update matrix was formulated.

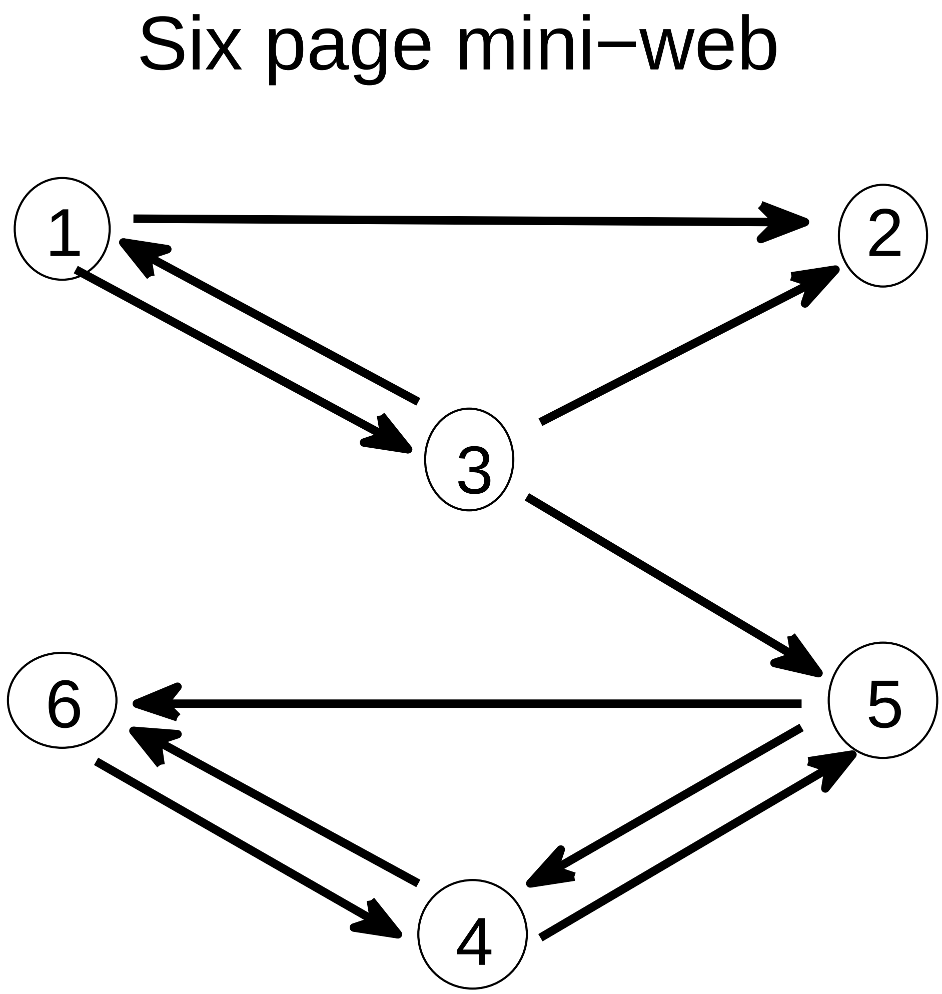
$$\mathbf{G} = \alpha\mathbf{H} + \alpha\mathbf{a}\mathbf{1}^T + \frac{1-\alpha}{n}\mathbf{1}\mathbf{1}^T \tag{3}$$

where **a** is a column vector with elements $a_i = \frac{1}{n}$ if P_i links to no pages or 0 otherwise and **1** is a column vector of ones. Utilizing the PageRank update matrix, the PageRank vector can be iteratively updated as follows.

$$\boldsymbol{\pi}_{k+1}^T = \boldsymbol{\pi}_k^T G \tag{4}$$

Based on a given mini-web, different values for α, and different values for k, we intend to determine what α value Google should use. Using MatLab we will be able to easily compute the matrices and vectors neccessary to determine an appropriate α value.

Dr. Spiller has provided a mini-web consisting of six pages with links as shown below.



Given this web, we start by determining the B_{P_i} and $|P_j|$, as follows.

Page	1	2	3	4	5	6
B_{P_i}	3	1, 3	1	5, 6	3, 4	4, 5
$\lvert P_j \rvert$	2	0	3	2	2	1

From our values for B_{P_i} and $|P_j|$, we are able to calculate $\mathbf{H}$, $\mathbf{a}$, $\mathbf{1}$, and $\mathbf{1}^T$, where $n = 6$.

$$\mathbf{H} = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{a} = \begin{bmatrix} 0 \\ 1/6 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \mathbf{1} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \mathbf{1}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Plugging the matrix and vectors determined above into equation 3, the transition matrix **G** can be symbolically determined in terms of α.

$$\mathbf{G} = \begin{bmatrix} 1/6-\alpha/6 & \alpha/3+1/6 & \alpha/3+1/6 & 1/6-\alpha/6 & 1/6-\alpha/6 & 1/6-\alpha/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ \alpha/6+1/6 & \alpha/6+1/6 & 1/6-\alpha/6 & 1/6-\alpha/6 & \alpha/6+1/6 & 1/6-\alpha/6 \\ 1/6-\alpha/6 & 1/6-\alpha/6 & 1/6-\alpha/6 & 1/6-\alpha/6 & \alpha/3+1/6 & \alpha/3+1/6 \\ 1/6-\alpha/6 & 1/6-\alpha/6 & 1/6-\alpha/6 & \alpha/3+1/6 & 1/6-\alpha/6 & \alpha/3+1/6 \\ 1/6-\alpha/6 & 1/6-\alpha/6 & 1/6-\alpha/6 & 5\alpha/6+1/6 & 1/6-\alpha/6 & 1/6-\alpha/6 \end{bmatrix}$$

As **G** has been determined symbolically in terms of α, we can plug in values for α. With each increasing iteration for $\boldsymbol{\pi}_k$, the values in the vector $\boldsymbol{\pi}_k$ converge to $\boldsymbol{\pi}$. For $\alpha = 0.9$, a minimum of 15 iterations are required to obtain the steady state. The resulting PageRank vector after 15 iterations is

$$\boldsymbol{\pi}_{15} = \boldsymbol{\pi} = \begin{bmatrix} 0.0372 \\ 0.0540 \\ 0.0415 \\ 0.3750 \\ 0.2060 \\ 0.2862 \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} 1.0000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.6101 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.4500 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.4500 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.3705 & 0 \\ 0 & 0 & 0 & 1 & 0 & -0.0896 \end{bmatrix}$$

$$\mathbf{V} = \begin{bmatrix} 0.0715 & 0.2931 & 0.0000 & 0.0000 & -0.0647 & -0.2123 \\ 0.1036 & 0.5094 & 0.0000 & 0.0000 & 0.0139 & 0.8541 \\ 0.0797 & 0.3415 & 0.0000 & 0.0000 & 0.0730 & -0.3636 \\ 0.7204 & -0.5891 & -0.7071 & 0.7071 & -0.6606 & 0.0184 \\ 0.3957 & -0.1414 & 0.7071 & -0.7071 & 0.7376 & -0.3047 \\ 0.5498 & -0.4135 & 0.0000 & 0.0000 & -0.0992 & 0.0082 \end{bmatrix}$$

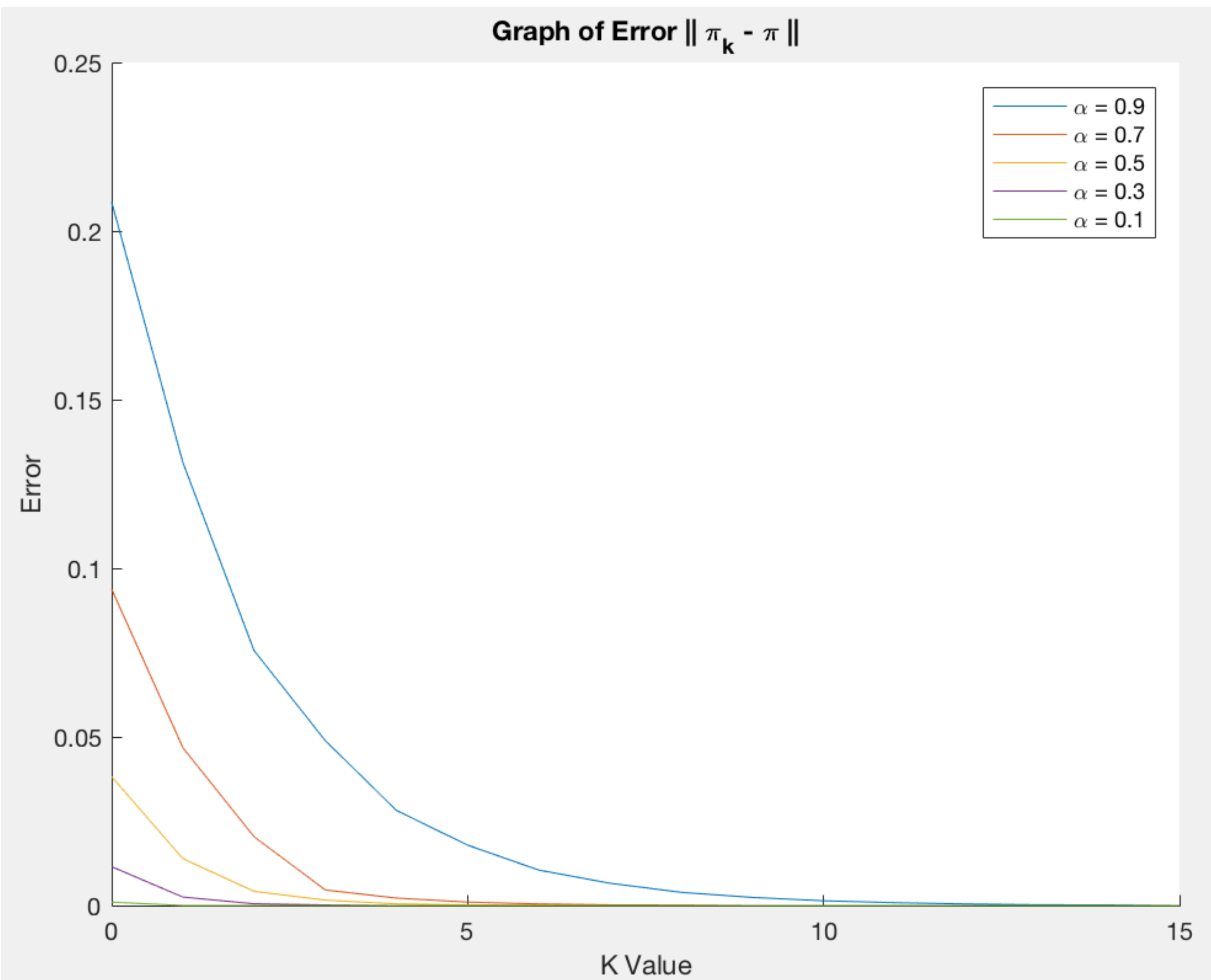
D shows the eigenvalues and **V** shows the corresponding eigenvectors. The PageRank vector is updated iteratively until it converges. The largest eigenvalue and corresponding eigenvector, λ and $\mathbf{v}_{max}$, are important for the update relationship of the Google Matrix, **G**. Because this is designed as a stochastic matrix, as K goes to infinity, the steady state is obtained with the $\boldsymbol{\pi}$ vector. The λ values are raised to the power of K and multiplied by the corresponding coefficient and eigenvector, so that the page vector does not become infinitely large, as would happen if this were a *substochastic* matrix design. The eigenvalue and eigenvector λ and $\mathbf{v}_{max}$ are the last iteration values that contribute to the Page Rank vector, and are the last changes that cause the matrix to converge. The initial vector $\boldsymbol{\pi}_0$ is an arbitrary vector in $\mathbf{R}^6$, and that can be written as a linear combination of eigenvectors of **G**. This is due to the fact that the eigenvectors of **G** form a basis of $\mathbf{R}^6$. While iterating through the $\boldsymbol{\pi}_{k+1}$

Based on the vector, $\boldsymbol{\pi}_{15}$, the ranks of the websites would be ordered P_4, P_6, P_5, P_2, P_3, P_1.

In addition, we can calculate $\boldsymbol{\pi}_k$ for $\alpha = 0.7$, 0.5, 0.3, and 0.1. For each value of α, there is a different number of iterations needed in order for the values of $\boldsymbol{\pi}_k$ to converge to $\boldsymbol{\pi}$. The table below shows the number of iterations necessary and the value of $\boldsymbol{\pi}$ for each value of α.

α	0.9	0.7	0.5	0.3	0.1
k	15	10	7	4	1
$\boldsymbol{\pi}$	$\begin{bmatrix}0.0372\\0.0540\\0.0415\\0.3750\\0.2060\\0.2862\end{bmatrix}$	$\begin{bmatrix}0.0852\\0.1150\\0.0932\\0.2899\\0.1866\\0.2302\end{bmatrix}$	$\begin{bmatrix}0.1162\\0.1452\\0.1245\\0.2390\\0.1759\\0.1992\end{bmatrix}$	$\begin{bmatrix}0.1392\\0.1601\\0.1456\\0.2044\\0.1699\\0.1808\end{bmatrix}$	$\begin{bmatrix}0.1581\\0.1661\\0.1607\\0.1781\\0.1670\\0.1700\end{bmatrix}$

Using this information, we obtain the error that occurs as the iteration increases for each value of α shown in the following graph.



As shown, while the value of α decreases, the number of iterations needed for $\boldsymbol{\pi}_k$ to converge also decreases. Based on the gathered information the most appropriate value that Google should use is 0.1 because it requires only 1 iteration for it to be the most accurate.