PageRank

Matrix methods for ranking web-pages

Ruben Bromée
Mithushan Krishnamoorthy
2021/12/09

Examiner:
Berkant Savas

1 Introduction

The relevance, applications and node representation of the web-pages are introduced in this section.

1.1 Relevance and applications

When we are searching for a topic on the internet, for example using Google, there is an extensive number of websites which are relevant for our query and search engines need a way to rank all of these websites in order to present the most relevant websites first. PageRank is a method to rank these websites according to relevance. The PageRank method is useful beyond ranking web-pages. Example for applicable areas are keyword extraction, ranking how influential users are in social networks, etc.

1.2 Node representation

Web-pages on the internet link to each other through hyperlinks. Let us represent these web-pages as nodes as seen in Figure 1. Where I_i represents the in-links to a web-page i and O_i represents the out-links from the web page.

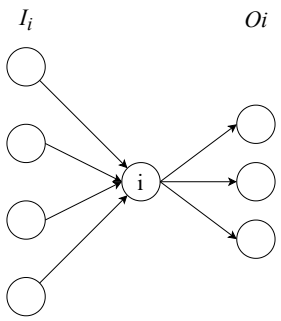

Figure 1: Web-pages represented as nodes.

A node graph can be used to describe several web-sites and their links to each other, as seen in Figure 2.

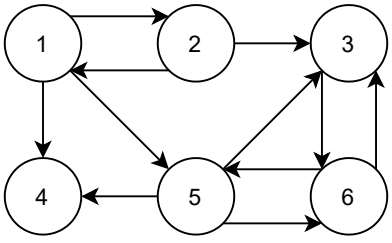

Figure 2: The node graph of 6 websites and their links to each other.

Another representation of a node graph that is more compact and useful when performing computations is the matrix representation. The node graph in Figure 2 can be represented using the matrix in Equation 1. The columns of the matrix represent the out-links from the node and the rows of the matrix represent the in-links to the node. The value in column 2 of row 1 represents the in-link to node 1 from node 2. In the same manner the values in row $2, 4$, and 5 in column 1 represent the out-links from node 1 to nodes $2, 4$ and 5.

$$L = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \tag{1}$$

2 Method

This section contains the pre-processing procedure of the link matrix, a description of the power method and a description of the HITS method.

2.1 Rank of a web-page

A web-page is said to be important if it has many in-links but a ranking system solely based on in-links could be easily manipulated by link-farms [1]. To prevent this the rank of a given page is defined as a weighted sum of the pages which has a out-link to the said page. The weighting is such that the rank of the given page is distributed equally to its out-links. The rank of web-page i in Figure 3 then becomes one third of the rank of j because j's rank is evenly distributed to all three of its out-links.

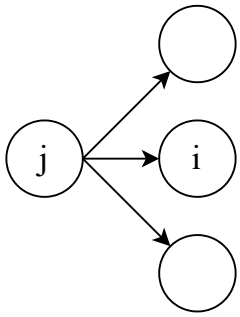

Figure 3: Node j linking to node i.

This definition is recursive so in implementation it becomes a fixed-point iteration as shown in Equation 2, where r_i^{k+1} is the rank of the web-page i in iteration $k+1$, I_i is the in-links to i, r_j^k is the rank of node j in iteration k and N_j is the number of out-links from node j.

$$r_i^{(k+1)} = \sum_{j \in I_i} \frac{r_j^{(k)}}{N_j}, \qquad k = 1, 2, 3, ... \tag{2}$$

2.2 Column stochastic and irreducible matrix

Imagine a random surfer traversing the internet, clicking links at random. The total probability to travel to all out-links of a web-sites should be 100% or 1. To achieve this, each column sum in the link matrix should be 1 making the matrix *column stochastic*. Let us begin by modifying the link matrix in 1 to have evenly distributed probabilities in each column, giving the matrix in 3. This is done by dividing the columns with a non-zero column sum.

$$Q = \begin{pmatrix} 0 & \frac{1}{3} & 0 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 & \frac{1}{3} & 0 \end{pmatrix} \tag{3}$$

The node 4 in Figure 2 has no out-links. This is shown by the fact that all elements in fourth column of the matrix in Equation 3 is zero. This means that a random surfer will get stuck if it travels to node 4. In other words, the accumulated rank via the in-links of node 4 is never distributed further. In order to prevent this, the random surfer should have an equal probability to travel to every out-link from node 4. This is done in equation 4. The number of nodes is denoted by n.

$$d_j = \begin{cases} 1 & if \quad N_j = 0 \\ 0 & \text{otherwise} \end{cases}, e \in \mathbb{R}^n, e = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}, P = Q + \frac{1}{n} e d^T, \tag{4}$$

If a node in a node-graph has no in-links, this means that there is no way to traverse from one sub graph to another. This will lead to the random surfer getting stuck in a sub-graph of the node graph. An example of a graph like this is in Figure 4 where the random surfer will get stuck in the sub-graphs on each side of node 4. The link matrix representing this node graph seen in Equation 5 is *reducible*, meaning there isn't a path from every node to every other node in the graph [1].

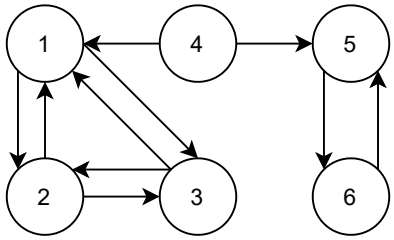

Figure 4: A node graph representing a reducible link matrix.

$$P = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{5}$$

To prevent this, a link from every node to all other nodes must be added. Equation 6 does this by taking the convex combination of matrix P. Since P is column stochastic matrix A is also column stochastic. $\alpha = 0.85$ is often used.

$$A = \alpha P + (1 - \alpha) \frac{1}{n} e e^T \tag{6}$$

2.3 Power method

The Equation 2 is equivalent to the scalar product between the rows of our link matrix and the rank vector. That equation expressed in matrix terms becomes $\lambda r = Ar$ where $\lambda = 1$ and it is now clear that the rank vector r is an eigenvector of the link matrix with eigenvalue one. The fixed point iteration given in 2 could be rewritten as Equation 7 which is called the power method.

$$r^{(k+1)} = Ar^{(k)}, \qquad k = 1, 2, 3... \tag{7}$$

2.4 Redefinition of Matrix A in terms of Q

The link matrix Q is sparse but after the changes made by Equations 4 and 6 the matrix A becomes dense. In practical problems the network graphs contains a very large number of nodes. This makes storing and processing the matrix in dense form impossible. To circumvent this, a sparse matrix can be used which only stores the non-zero values of a matrix. In the Stanford dataset[5] that was used for this implementation there were roughly 0.003% non-zero values.

Equation 8 shows how matrix A is rewritten in terms of Q by using the Equations 4 and 6.

$$A = \alpha\big(Q + \frac{1}{n}ed^T\big) + \frac{(1-\alpha)}{n}ee^T \tag{8}$$

Using Equation 8 in Equation 7 gives Equation 9.

$$r^{(k+1)} = \alpha\big(Q + \frac{1}{n}ed^T\big)r^{(k)} + \frac{(1-\alpha)}{n}e(e^T r^{(k)}) \tag{9}$$

The initial rank vector r is given in Equation 10

$$r \in \mathbb{R}^n, r = \frac{1}{n}\begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix}, \tag{10}$$

The matrix A is column stochastic, r is normalized and has no negative values. This gives the properties in Equation 11.

$$\|r\|_1 = 1, \|Ar\|_1 = 1, \|r\|_1 = e^T r \tag{11}$$

Equation 8 can be rewritten into 12 where part of the second term is denoted as β. Calculating beta is costly and therefore we want to define β another way.

$$\begin{aligned} Ar = \alpha Qr + \frac{1}{n}e(\alpha d^T r + (1-\alpha)e^T r) \\ \beta = \alpha d^T r + (1-\alpha)e^T r \end{aligned} \tag{12}$$

Using the properties in Equation 11 and Equation 12 can be used to define β as seen in Equation 13

$$\begin{aligned} \|Ar\|_1 = e^T\left(\alpha Qr + \frac{1}{n}e\beta\right) = e^T\alpha Qr + e^T\frac{1}{n}e\beta \Leftrightarrow \\ \beta = 1 - e^T\alpha Qr = 1 - \|\alpha Qr\|_1 \end{aligned} \tag{13}$$

The final expression for a new iteration of the rank vector is then given by Equation 14

$$r^{(k+1)} = \alpha Qr^{(k)} + \beta\frac{1}{n}e \tag{14}$$

The power method uses Equation 14 iteratively until the one-norm of the difference of $r^{(k+1)}$ and $r^{(k)}$ becomes smaller than a value $\epsilon = 10^{-4}$.

2.5 HITS method

In the HITS method nodes are given a hub score and an authority score. Many out-links increases the hub score for a node and many in-links increases the authority score for the node. Since the rows in the link matrix represents the number of in-links to a node and the columns represent the number of out-links from a node the hub score h and the authority score a can be calculated using Equation 15.

$$\begin{aligned} h &= La \\ a &= L^T h \end{aligned} \tag{15}$$

Equation 15 can be rewritten to define hub and authority scores in terms of themselves as seen in Equation 16.

$$\begin{aligned} h &= LL^T h \\ a &= L^T La \end{aligned} \tag{16}$$

2.6 Tools

We used Python to implement both page rank methods. The numpy [2] and scipy [3] libraries were used to implement linear algebra operations and implementing the sparse matrices. Matplotlib [4] was used to create the plots.

3 Result

Figure 5 shows the PageRank score for every web-page in the Stanford dataset. Figure 6 show how the residual decreases for every iterations. The results we achieved seem reasonable when compared to the results in [1] on the same dataset. In Figure 7, the authority scores for a number of web-pages related to the search query 'California' is given. In Figure 8 the hub score for the same dataset is shown.

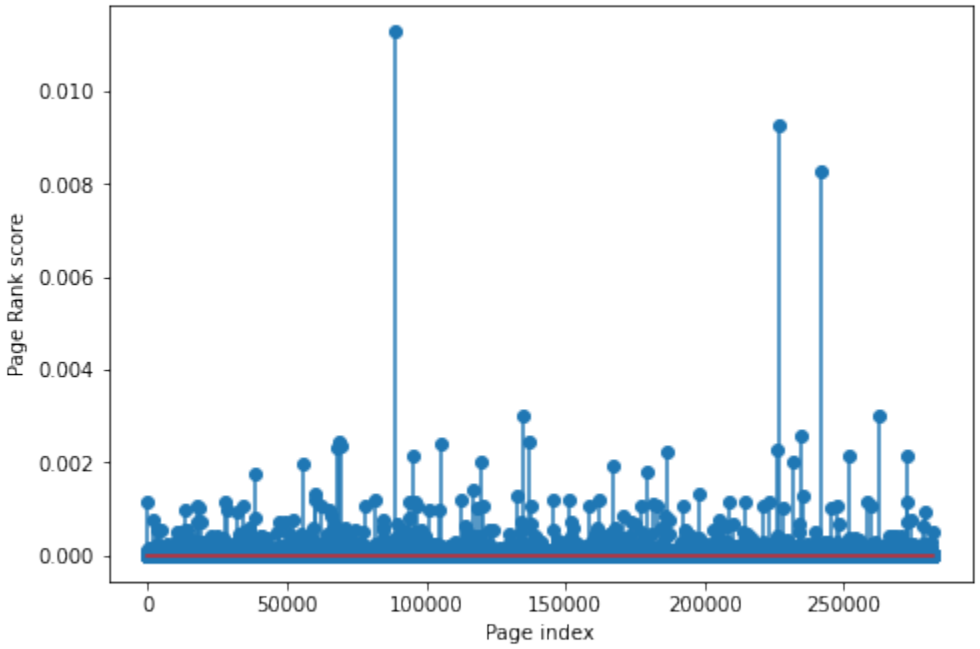

Figure 5: PageRank score for each page in the Stanford dataset.

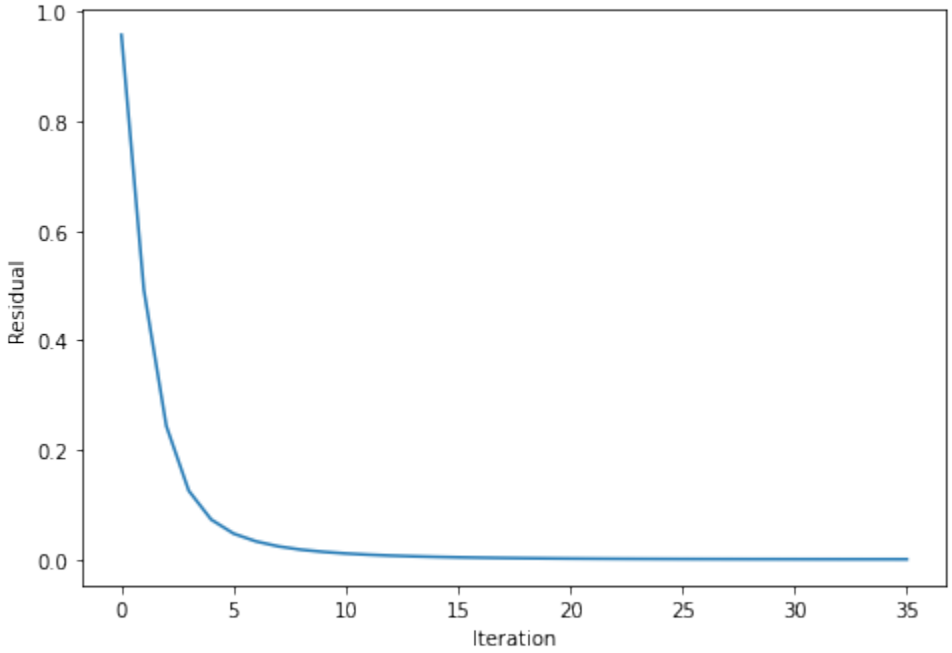

Figure 6: The residual between iterations of the PageRank algorithm using the Stanford dataset.

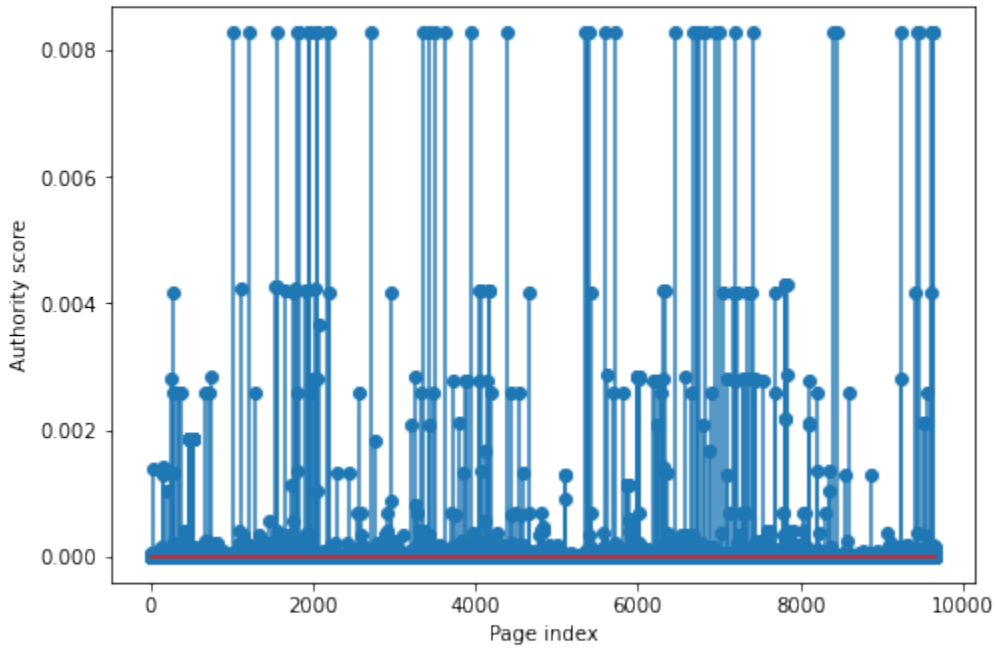

Figure 7: Authority score of the dataset containing results for the query 'California'.

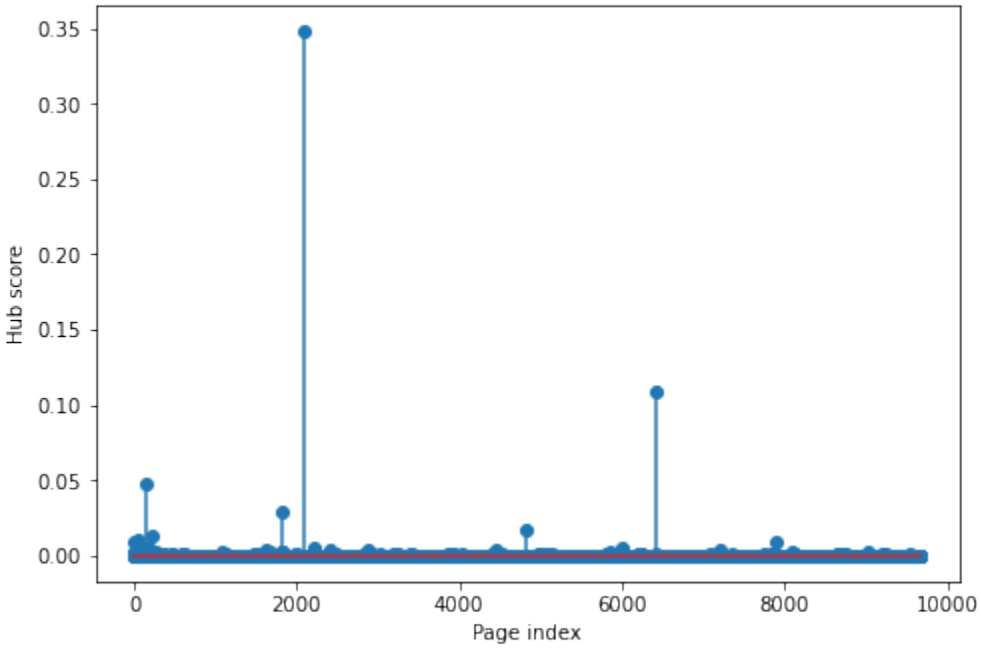

Figure 8: Hub score of the dataset containing results for the query 'California'.

4 Discussion

Sparse matrices are a great way to store large sparse matrices. Especially in a case like this where the data contained many zero-values. It was surprising to see how few iterations were required to get an acceptable residual and how fast the residual reduced. It was also surprising to see how efficient linear algebra operations such as matrix multiplication was with sparse matrices. The size of the matrices used in linear algebra operations was also impressive.

The results from the HITS method show that authority scores are higher than hub scores in general but the highest hub scores are a lot higher than the highest authority scores.

Improvements that could be made to the method is combining the power method with the HITS method by ranking web-pages by PageRank, authority score and hub score. To optimize the power method for real use cases a sub-graph of the internet related to a search query can be used.

References

[1] Lars Eldén, *Matrix methods in data mining and pattern recognition, Second edition*, 2019, siam

[2] Numpy, https://numpy.org/, retrieved 2021-12-09

[3] Scipy, https://scipy.org/, retrieved 2021-12-09

[4] Matplotlib, https://matplotlib.org/, retrieved 2021-12-09

[5] Stanford dataset, https://snap.stanford.edu/data/web-Stanford.html, retrieved 2021-12-09.