The Reordered Google PageRank Algorithm

Jonathon D'Arcy
S1607860

1 Introduction

It's well known that while PageRank was the spark that ignited Google, it was not the final iteration of the search engine we know today. Since then the company has reworked and adapted their back-end many times in order to improve their services, whether it was combating spam, expanding international profile, or implementing new search categories. While these improvements allowed for Google to become so successful it became the foremost example of a proprietary eponym the company always needed to make sure that with each update they could economically store their indexes and quickly provide results to the user. In the very early days of Google, this could be kept under control with the help of Stanford's engineering department, but with the rapid expansion of the web it became obvious that new methods had to be created to improve storage and speed. In this report we seek to investigate an early solution to such needs proposed by Langville and Meyer in a paper titled the Reordered PageRank [1], which aimed to take advantage of the large proportion of pages containing no outward links in order to decrease the size of indexes and by extension reduce the operations required to find the PageRank vector.

2 Notation

1. Hyperlink Matrix	1. $\mathbb{H}$
2. Google Matrix	2. $\mathbb{G}$
3. Teleportation Probability Factor	3. α
4. Teleportation Vector	4. v
5. PageRank Vector	5. π
6. Set of Non-Dangling Pages	6. ND
7. Set of Dangling Pages	7. D

3 An Unexploited Property

What makes PageRank so powerful is the exploitation of many unique properties in the Google Matrix, $\mathbb{G}$, such as irreducibility and stochaticity. These proprieties allow us to use powerful theorems such as Perron-Frobenius to ensure we get the results we desire. Though while we use a lot from $\mathbb{G}$, Langville and Meyer noticed that we seem to ignore and even dispose of a powerful property in the Hyperlink Matrix, $\mathbb{H}$. This is the property that $\mathbb{H}$ is a sparse matrix[1], which means that a majority of the elements of $\mathbb{H}$ are zero. While typically sparse matrices are taken advantage of for their ability to reduce both storage and computation time, here the matrix was instead filled with elements that portrayed leaving to a random website. Langville and Meyer, decided to postpone this inclusion of teleportation and use this property of $\mathbb{H}$ to their advantage when creating a new algorithm that computes the PageRank vector. To do this

[1]We note that $\mathbb{H}$ has no requirements for being a sparse matrix, but for Google's uses in indexing websites this is assumed.

they first reordered $\mathbb{H}$ so that all the dangling pages were moved towards the end of the matrix as below

$$\mathbb{H} = \begin{matrix} \\ \text{D} \\ \text{ND} \end{matrix} \begin{matrix} \begin{matrix} ND & D \end{matrix} \\ \begin{pmatrix} \mathbb{H}_{11} & \mathbb{H}_{12} \\ 0 & 0 \end{pmatrix} \end{matrix}. \tag{1}$$

Where ND denotes the set of non-dangling pages and D denotes the set of dangling pages. This form of $\mathbb{H}$ allows for a simple reduction in storage size as there is no sense in storing anything that is not in the sub-matrices[2], $\mathbb{H}_{11}$ or $\mathbb{H}_{12}$, as we know it will be 0.

4 The New Algorithm

While the iterative power method is the most common method of calculating the PageRank vector, π, it is not the only way. One way is to formulate the problem as a linear system as shown below.

Theorem 1. *Solving the linear system*

$$x^T(I - \alpha\mathbb{H}) = v^T \tag{2}$$

where v^T denotes the teleportation vector and letting $\pi^T = x^T/||x^T||_1$ produces the PageRank vector.

This method of directly solving the linear system is typically not used as solving large matrices with it is a difficult and expensive process, but implementing our new break down of $\mathbb{H}$ gives us the ability to reform the system into one that is much easier to solve.

$$x^T \begin{pmatrix} \mathrm{I} - \alpha\mathbb{H}_{11} & \alpha\mathbb{H}_{12} \\ 0 & \mathrm{I} \end{pmatrix} = v^T \tag{3}$$

$$x^T = v^T \begin{pmatrix} \mathrm{I} - \alpha\mathbb{H}_{11} & \alpha\mathbb{H}_{12} \\ 0 & \mathrm{I} \end{pmatrix}^{-1} \tag{4}$$

$$x^T = v^T \begin{pmatrix} (\mathrm{I} - \alpha\mathbb{H}_{11})^{-1} & \alpha(\mathrm{I} - \alpha\mathbb{H}_{11})^{-1}\mathbb{H}_{12} \\ 0 & \mathrm{I} \end{pmatrix} \tag{5}$$

We can then partition x^T and v^T into sections which correspond to the size of block $\mathbb{H}_{11}$. This allows us to write,

$$x^T = \left(x_1^T \,|\, x_2^T\right) = \left(v_1^T(\mathrm{I} - \alpha\mathbb{H}_{11})^{-1} \quad | \quad \alpha v_1^T(\mathrm{I} - \alpha\mathbb{H}_{11})^{-1}\mathbb{H}_{12} + v_2^T\right). \tag{6}$$

The benefit of writing x^T like this is that we only have to solve the section x_1^T as a linear system and can use that solution to directly compute x_2^T. This is because we can rewrite x_2^T as

$$x_2^T = \alpha x_1^T\mathbb{H}_{12} + v_2^T. \tag{7}$$

This reduces our original linear system problem down vastly in size, as now instead of solving a linear system the size of $\mathbb{H}$ we now solve for one that is only the size of the set ND. What is even better is that we will later show we can use iterative methods on this smaller system to greatly improve the speed of this step. With these methods in place for finding x_1^T and x_2^T we can combine our results to produce π^T as such,

$$\pi^T = [x_1^T\, x_2^T]/||[x_1^T\, x_2^T]||_1. \tag{8}$$

Giving us a complete algorithm for finding the PageRank vector.

[2]Though we will have to store the permutation that results in this but as long as we have at least one dangling page we beat storage requirements.

Algorithm 1: The Reordered PageRank

Result: The PageRank Vector, π^T.
Initials: The Hyperlink Matrix, $\mathbb{H}$, and the Teleportation Coefficient, α.

1. Reorder $\mathbb{H}$ so that it has the structure of (1).
2. Solve for x_1^T in the system $x_1^T(\mathrm{I} - \alpha\mathbb{H}_{11}) = v_1^T$.
3. Compute $x_2^T = \alpha x_1^T \mathbb{H}_{12} + v_2^T$.
4. Normalize to get $\pi^T = [x_1^T\ x_2^T]/|||[x_1^T\ x_2^T]|||_1$.

5 Iterative Methods for Step 2

While this is an improvement on solving for the PageRank vector as a linear system, we cannot compete with the standard PageRank method, unless we can produce an iterative solve for step 2. Luckily, it is very easy to find a solver for this, in Langville and Meyer's paper they discuss using the Jacobi Method for step 2 of their Algorithm, however they do not discuss why this is chosen over the Gauss-Seidel Method and I would be remiss to not include it as it is typically faster than the Jacobi Method. In either case we need to show that the iterative method will converge for our linear system, and as it happens we satisfy a convergence property which covers both methods.

Theorem 2 (Jacobi Method and Gauss-Seidel Method Convergence)**.** *A sufficient condition for convergence under the Jacobi Method and Gauss-Seidel Method is that the matrix acted upon is strictly diagonally dominant. More formally we write: Suppose $A \in \mathbb{R}^{n\times n}$ is strictly diagonally dominant, i.e.*

$$|a_{ii}| > \sum_{j\neq i} |a_{ij}|, \quad \forall i = 1, \ldots, n$$

Then the both the Jacobi Method and Gauss-Seidel Method applied to $Ax = b$ converges: $\|\mathbf{e}_k\|_\infty \to 0$ as $k \to \infty$, where e_k denotes the error at iteration k. [3]

Claim: $(\mathrm{I} - \alpha\mathbb{H}_{11})$ is always strictly diagonally dominant.

Proof. We know that in $\mathbb{H}$ any row, r_i, has the following properties:

1. Any element $a \in r_i$ is in $[0, 1]$;
2. $\sum r_i = \begin{cases} 0 & r_i \in \mathrm{D} \\ 1 & Otherwise \end{cases}$

After we rearrange the matrix to form (1) and look at $\mathbb{H}_{11}$, we trivially see the first property must still hold. While the second changes slightly as the rows of $\mathbb{H}_{11}$ can now have their sum in the range $[0, 1]$, rather than strictly being 0 or 1. This is because while we are only looking at rows that are in the set ND, any element in ND which was sent to D will now be outside $\mathbb{H}_{11}$. We then multiply each of these rows by $-\alpha$ which is defined in the range $(-1, 0)$. Making each element in any row and the sum of any row be in the range $(-1, 0]$. Since by convention no element of $\mathbb{H}$ can link to itself, the sum of any row of $-\alpha\mathbb{H}_{11}$ must equal the sum of that row without the diagonal element. This also tells us that when we add the identity matrix the diagonal elements will all equal 1. Since each element not in the diagonal of a row is in the range $(-1, 0]$ and the sum of that row without the diagonal element is also in $(-1, 0]$ we can say that for any row in $\mathrm{I}-\alpha\mathbb{H}_{11}$ the sum of the absolute value of all non-diagonal elements must be in the range $[0, 1) < 1$ and as each diagonal element is equal to 1 we have that $(\mathrm{I} - \alpha\mathbb{H}_{11})$ is always strictly diagonally dominant. □

As we now know that both the Jacobi Method and the Gauss-Seidel Method converge on step 2 of the Reordered PageRank Algorithm, we can apply them making the prospects of Reordered PageRank very competitive with PageRank.

6 Conclusion

To demonstrate the ability of their algorithm Langville and Meyers ran some experimental trails comparing the Reordered PageRank and the original PageRank. They gathered four data sets to directly compare the performance of the two methods. What they found was this new method was both faster and simpler to store than the original PageRank on all four data sets. In fact when testing on their largest data set consisting of over 450,000 pages the new method produced the PageRank vector over 5 times faster. In the analysis of the algorithm they went on to state that "The reordered PageRank algorithm is guaranteed to outperform the original PageRank method, as long as some dangling nodes are present" (Langville and Meyer, 2118). It is unknown if Google ever chose to adopt the algorithm or one of the many others that appeared around the same time, but with a web that grows so large some claim it can no longer be accurately measured[5] we can be quite sure that the original PageRank was shelved for better methods such as this.

References

[1] Langville & Meyer, *A Reordering for the PageRank Problem*. SIAM Journal on Scientific Computing, 27(6), pp.2112–2120.

[2] J. Pearson, Entrepreneurship in the Mathematical Sciences Lecture Notes, University of Edinburgh, 2019.

[3] A. Teckentrup, Numerical Linear Algebra Lecture Notes, University of Edinburgh, 2017.

[4] Antal van den Bosch, Toine Bogers & Maurice de Kunder, *Estimating search engine index size variability: a 9-year longitudinal study*, `https://www.dekunder.nl/Media/10.1007_s11192-016-1863-z.pdf`

[5] Adam Kilgarriff, *Googleology is Bad Science*, Lexical Computing Ltd. and University of Sussex

[6] Erik Andersson & Per-Anders Ekström, *Investigating Google's PageRank algorithm*, UPPSALA UNIVERSITET, 2004.

[7] A. I. Bakari & I. A. Dahiru, *Comparison of Jacobi and Gauss-Seidel Iterative Methods for the Solution of Systems of Linear Equations*, Asian Research Journal of Mathematics, 2018.