Google Matrix Analysis for Elections

Interdisciplinary Project Report

Supervisors

Samer El Zant

Eldar Babayev

Prepared By

CS017 Group 4

March, 2021

Table of Contents

Introduction

The inventors of Google PageRank are Sergey Brin and Larry Page. This algorithm was created in 1996 and the prototype for PageRank was released and became the backbone behind Google search results in 1998.

The Basic Concept of Google's PageRank is to work by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying reason is that more important websites are likely to receive more links from other websites. Website consists of pages and those pages can be connected to one another via inbound or outbound links and the PageRank algorithm generally assumes that if many other pages are linking to a particular page, then that is most probably more important, than a web page that is linked to less.

There are two possible ways of linking the pages:

1. Inbound link - is a link from website to another page on the same website.
2. Outbound link - is a link from one website to another website.

The web page with high PageRank means that in the google search is going to be within the first result.

The goal of this project is to use Google Matrix algorithm to analyze elections and find the elected candidates.

Classical Election Methods

Before emphasising the Google Matrix analysis for the elections it might be necessary to talk about the classical election methods in order to understand different kinds of election methods and make a better comparison between election with a page rank algorithm and election with those classical ones.

What is the electoral system? It is the set of rules that determines how elections are conducted and how their results are decided. Some electoral systems choose a single winner to a unique position, such as prime minister, president, whereas others choose numerous winners, such as members of the boards, and etc. Thus, the election methods is divided into 3 main types:

- Plurality/Majority Systems
- Proportional Representation Systems
- Mixed Systems

Plurality/Majority Systems

Generally saying, one seat per electoral area/department and one candidate can be elected from a given area.

Plurality - applicants can win a seat after they win the most votes without prevailing over a 1/2 of percent of the vote.

Majoritarian - attempts to make sure that the winning candidate gets an absolute majority, through using voters' 2d choices to produce a winner.

Proportional Representation System

There are lots of various ways to determine who will win and some are more proportional and some are less. A more proportional way would mean that a candidate received 1-3rd of the vote could expect 1-3rd of the winning seats in the department. That means, voters order the candidates while voting and it conducts the proportions between each other.

Mixed Systems

In mixed systems representatives are elected through a combination of different elements of the PR and plurality systems. Let's say, voters use two votes: one to directly elect an individual member to serve as their representative, and a 2nd for a political party or parties to fill seats withinside the legislature allotted in line with the proportion of the vote percentage they receive.

Google Matrix, Mathematical explanation

The PageRank rules work at the primary concept that the extra essential and beneficial a web page is, the extra different pages will hyperlink to it. Therefore, a web page that has many different pages linking to, it's far extra essential, and could seem important withinside the seek results. So, technically, importance is the essential eigenvector of the transition matrix of the Web.

If page j has d outbound links, each link gets rj / d votes , their summation with all inbound links becomes the rank. $r_j = \sum_{i->j} \frac{r_i}{d_i}$

Gaussian elimination method works for small examples,however we may have large web sized graphs,so that means, we need a better method → Matrix

We can represent the graph as an adjacency matrix S which will be column stochastic. This means that all the columns must add up to 1. So for node j all the entries, in column j will sum to 1.

If i -> j, then $S_{ji} = \frac{1}{d}$ $else\ S_{ji} = 0$

Also knowing that the importance score of page ri should sum to 1,from that we can rewrite it as r=Sxr where the PageRank vector will be the eigenvector of the stochastic web matrix S where the eigenvalue equals to 1.

Random walk formulation:

Random Walks has a strong relationship with Pagerank.When the iteration is at given time t, it will be at the i-th page, Afterwards when any outbound hyperlink is chosen with randomness at time t+1, p(t) is then a probability at a given time t , and this method can functionalize infinitely. So the formula:

P(t+1) = S x p(t)

By the time, when t goes to infinity, the random walk will be no more random but constant:

P(t+1) = S x p(t) =p(t)

$S_{ij} = P(X_t = p_j | X_{t-1} = p_i)$

Since S= [Sij], then we can also deduce that S is a transition matrix

PageRank: Problems

1. **Dead ends:** “in some pages there are not outbound-links-but only inlinks From that, it is deduced that the adjacency matrix is no longer column stochastic and will not construct the importance correctly, since there is no way for that to be equal to 1”

2. **Spider traps**: All out-links are within the same group. Power iteration will converge within the same node having all the importance and leave a with no importance.

Random Jumps: Damping Factor, Google Matrix Solution

When there is no where to go, or it is just trapping back to the same group, to prevent adjacency matrix not being able to become stochastic, the damping factor helps there and it jumps into the different links. The key step in computing page rank is the matrix-vector multiplication

- With probability β, it will continue a link at random with order

- With probability 1-β,will jump to other random page
- Based on the conducted researches, β is varying in the range 0.8 to 0.9

Never get stuck in a spider trap by teleporting out of it in a finite number of steps

Make matrix column stochastic by always teleporting when there is nowhere else to go

So with all above mentioned, Initial PageRank Algorithm becomes:

$$r_j = \sum_{i \to j} \frac{r_i}{d_i} + (1 - \beta)\frac{1}{N}$$

Where, Google Matrix assumes S has no dead ends and traps.

$$G = \beta S + (1 - \beta)[\frac{1}{N}]_{NXN}$$

However, We want to be able to iterate as many times as possible in order to improve the rank performance, but if N is a big number. Then there memory problem will occur, so the formula needs some configuration with simple mathematical steps:

$r = Axr,\ where\ A_{ji} = \beta S_{ji} + \frac{1-\beta}{N}$ So, $r_j = \sum_{i=1}^{N} A_{ji}\, x\, r_i$

$r_j = \sum_{i=1}^{N}\left[\beta S_{ji} + \frac{1-\beta}{N}\right] x\, r_i = \sum_{i=1}^{N} \beta S_{ji}\, x\, r_i + \frac{1-\beta}{N}\sum_{i=1}^{N} r_i = \sum_{i=1}^{N} \beta S_{ji}\, x\, r_i + \frac{1-\beta}{N}$ (since $\sum r_i = 1$)

So, Final PageRank equation: $r = \beta S\, x\, r + \frac{1-\beta}{N}$

Data Collection, Preprocessing

In order to implement Google Matrix analysis in elections, data should be collected first, so it is what has been done in the early stages of the project. The elections for "student body presidency" were held via the Google Forms platform, where each student from L3, exactly 114 students, should have voted for 5 other L3 students, by mentioning their exact full names and the major they are pursuing. There was no definite number of candidates chosen beforehand to make the data bigger.

The results of the elections are automatically added in the google sheets document, from where we took the data to preprocess it. From that data, it can be concluded that 76 students out of 116 in L3 participated in the online elections. From the statistics, we see that 30.26% who voted is from Computer Science major and 17.1% from Geophysical Engineering, which makes 23 out of 42 possible and 13 out of 23 possible respectively. From both Petroleum Engineering and Chemical Engineering 20 students participated in the elections, which makes exactly 26.3%.

As the data is collected, it should be preprocessed before being passed to the program as an argument. The reason for preprocessing is that different students have written the name of the exact student differently, for example, in the data, there were “Elvin Aghalarov”, “Alvin Aghalarov”, which indicate the same person, but the name was written differently. So, in order to get the correct result, the data had to be preprocessed.

Voter	Candidate 1	Candidate 2	Candidate 3	Candidate 4	Candidate 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed: 10
Isa Baghirov	Isa Baghirov	Kanan Habibli	Kamal Taghiyev	Narmin Ganbarli	Mahammad Guliyev	NaN	NaN	NaN	NaN	NaN
Kanan Mikayilov	Samir Aghayev	Anar Mammadov	Turqay Gardashli	Aysu Majidli	Narmin Ganbarli	NaN	NaN	NaN	NaN	NaN
Rashad Aliyev	Javad Alizada	İbrahimali Mammadov	Mirzemehdi Karimov	Niyazi Gadiri	Kanan Habibli	NaN	NaN	NaN	NaN	NaN
Shirin Shukurov	Rufat Huseynov	Almaz Omarov	Reshad	Rashad Aliyev	Teymur Kosayev	NaN	NaN	NaN	NaN	NaN
Javid Guliyev	Anday Ismayilzade	Sadiq Gojayev	Mecid	Ramin Afandiyev	Elvin Salmanov	NaN	NaN	NaN	NaN	NaN

Initial table

This process is described below:

1. First, the full names of students who got the votes were trimmed in order to get rid of unnecessary whitespaces.
2. The names of students written wrongly were replaced by their correct names using the python dictionary.
3. The votes for students, whose names are written too incomprehensible, e.g. the candidate didn’t have their surname, were replaced with *None* using the python dictionary.
4. In order to have a square adjacency matrix, it should have the same number of students who participated and who got votes. So, there were added the students who didn’t vote but who got some votes.

Voter	Candidate 1	Candidate 2	Candidate 3	Candidate 4	Candidate 5
Isa Baghirov	Isa Baghirov	Kanan Habibli	Kamal Taghiyev	Narmin Ganbarli	Mahammad Guliyev
Kanan Mikayilov	Samir Aghayev	Anar Mammadov	Turqay Gardashli	Aysu Majidli	Narmin Ganbarli
Rashad Aliyev	Javad Alizada	Ibrahimali Mammadov	Mirzemehdi Karimov	Niyazi Gadiri	Kanan Habibli
Shirin Shukurov	Rufat Huseynov	Almaz Omarov	None	Rashad Aliyev	Teymur Kosayev
Javid Guliyev	Anday Ismayilzade	Sadiq Qojayev	None	Ramin Afandiyev	Elvin Salmanov
...	...	...	...	...	...
Saidanur Sideif-zada	None	None	None	None	None
Salim Kazim-zade	None	None	None	None	None
Sayad Baghirli	None	None	None	None	None
Sona Rustamova	None	None	None	None	None
Turqay Umudzade	None	None	None	None	None

The result of the preprocessing

Note: The last people in the table didn't vote for any candidate, therefore all of their candidates are *None*.

This process may vary depending on the election form. It may have taken some additional time from us to preprocess this data, because our election form was too simple, to make it easy to vote, Otherwise, there could be fewer votes, which would be a disadvantage in our case.

PageRank Algorithm

H Matrix

H_mat was constructed by choosing the importance of each person (0.2 if they voted for all 5 candidates and 0 if they did not vote for anyone). The number 0.2 is obtained basically through dividing each vote by the number of candidates. Since we have 5 candidates, we get ⅕ = 0.2.

Google Matrix

The formula for Google Matrix is

$$G_{i,j} = \alpha * S_{i,j} + (1 - \alpha) * \frac{1}{N}$$

Where α means damping factor (the probability that the user will stop clicking in some reference (c) Sergey Brin). The value of α is a scalar that should be between 1 and 0 and for us it is 0.85.

PageRank

The PageRank is calculated using power iteration until the convergence point of eigenvectors of Google matrix. Actually, we choose the number of iterations manually, but we can easily assume that after 80 it will converge, because of the small number of dimensions in the matrix. The information about the power iteration is written in the chapter **"Google Matrix, Mathematical explanation".**

Recursive formula for eigenvector approximation:

$$\lambda_{k+1} = \frac{A * \lambda_k}{||A * \lambda_k||}$$

In our analysis, we chose to have 100 iterations for the convergence of the matrix.

Final elections

To show the difference between Google Matrix / PageRank and one of the common types of elections Majoriatan, we analyzed the data using both algorithms.

Here are the results obtained:

```
The people, who are elected using PageRank algorithm:
The 1 place: Narmin Ganbarli with rank 0.43006
The 2 place: Shoykyat Sharafyabi with rank 0.31572
The 3 place: Isa Baghirov with rank 0.29199
The 4 place: Kanan Mikayilov with rank 0.2745
The 5 place: Kamal Taghiyev with rank 0.27114
-----------------------------------------------------
The 6 place: Jeyhun Abbasov with rank 0.24742
The 7 place: Almaz Omarov with rank 0.23208
The 8 place: Samir Aghayev with rank 0.23161
The 9 place: Anar Mammadov with rank 0.19118
The 10 place: Sevinc Abbasova with rank 0.14959
```

```
The people, who are elected using Majoritar election algorithm:
The 1 place: Almaz Omarov with number of votes 19
The 2 place: Anar Mammadov with number of votes 15
The 3 place: Kanan Mikayilov with number of votes 14
The 4 place: Narmin Ganbarli with number of votes 13
The 5 place: Samir Aghayev with number of votes 11
---------------------------------------------------------------
The 6 place: Sarvar Mammadov with number of votes 9
The 7 place: Kamal Taghiyev with number of votes 9
The 8 place: Isa Baghirov with number of votes 8
The 9 place: Shoykyat Sharafyabi with number of votes 8
The 10 place: Kanan Habibli with number of votes 8
```

Comparison: PageRank Algorithm vs Standard Election

As seen from results of the experiment there are differences between the PageRank algorithm and Majoritar algorithm. Now let's analyse the key differences and reasons behind them. I will refer to PageRank Algorithm as first and Majoritar algorithm as second till the end of this section.

1. The winner of the first algorithm is Narmin Ganbarli whereas it is Almaz Omarov in the second algorithm. The reason is Almaz Omarov and other candidates which he

voted for such as Isa Bagirov and Shoykyat Sharafyabi have voted for Narmin Ganbarli. Because of this, even if Almaz Omarov wouldn’t vote for Narmin Ganbarli, she would still win.

2. Beside the ranking, the content of the results is also different. Two students, Kanan Habibli and Sarvar Mammadov have high scores in the second algorithm , however they are not included in the first one, instead Sevinj Abbasova and Jeyhun Abbasov are selected. The reason is they are voted by Narmin Gambarli, but voters of Kanan Habibli and Sarvar Mammadov are not highly scored.

Further improvements

In order to take this project to next level several things can be done like creating a website to open this project publicly. The features can include creating the candidates list and showing the results of both algorithms beside showing the differences between them. While choosing among candidates the recommendation system can be added also. This system can consider the previous choices of the voter and recommend the next ones based on them, pretty much like Google. If the voter does not have any idea who to choose he/she can filter the candidates based on some criterias that is defined by the owner of the election.

Conclusion

In this project we have used the PageRank algorithm for election. The experiment was conducted between L3 students. The students were asked to vote for 5 students. After gathering the data we have used two algorithms to select the top 10 students, PageRank Algorithm and Majoritar Algorithm. After comparing and analysing the results we can conclude that the PageRank algorithm can be used as an election method as its results were more relateble.

Bibliography

[1] A. N. Langville, C. D. Meyer, “Google’s Pagerank and Beyond: The Science of Search Engine Rankings”, 2006, Available: sng1lib.org/book/3616260/642749

[2] R. Tanase, R. Radu, “The Mathematics of Web Search”, “PageRank Algorithm - The Mathematics of Google Search”, pi.math.cornell.edu/~mec/Winter2009/RalucaRemus/Lecture3/lecture3.html

[3] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank Citation Ranking: “Bringing Order to the Web”, 18.01.1998, ilpubs.stanford.edu:8090/422/1/1999-66.pdf

[4] K. Mohan, J. Kurmi, “A Technique to Improved Page Rank Algorithm in perspective to Optimized Normalization Technique”, 04.1997, researchgate.net/publication/316698632_A_Technique_to_Improved_Page_Rank_Algorithm_in_perspective_to_Optimized_Normalization_Technique

[5] F. Shi & The Odum Institute, “Learn About PageRank in Python With Data From the Florentine Family Dataset (1994) “, 1994, methods.sagepub.com/base/download/DatasetStudentGuide/pagerank-in-florentine-1994-python

[6] S. Brin, L. Page, “The Anatomy of a Large-Scale Hypertextual Web Search Engine”, 04.1998, infolab.stanford.edu/~backrub/google.html

[7] L. Lyu, B. Fain, K. Munagala, K.Wang, “Centrality with Diversity”, 01.2021, arxiv.org/abs/2101.07371

[8] P. Contucci, E. Panizzi, F. Ricci-Tersenghi, A. Sırbu, “Egalitarianism in the rank aggregation problem: a new dimension for democracy”, 01.2014, arxiv.org/pdf/1406.7642.pdf

[9] Kamvar, S. “The Second Eigenvalue of the Google Matrix. Stanford University Technical Report” . http://ilpubs.stanford.edu:8090/582

[10] WILLS, REBECCA S. “When Rank Trumps Precision: Using the Power Method to Compute Google’s PageRank” , https://repository.lib.ncsu.edu/handle/1840.16/4953