BACHELOR

Analyzing politicians in the Dutch House of Representatives using PageRank

Nederlof, Jochem R.

Award date:
2022

Awarding institution:
Tilburg University

BEP

Analyzing politicians in the Dutch House of Representatives using PageRank

Jochem Nederlof
1344064
j.r.nederlof@student.tue.nl

,
Technische Universiteit Eindhoven and Tilburg University
Netherlands

1 Abstract

This report will investigate whether PageRank Scores per politician per period in the Dutch House of Representatives can say something useful about the next period. More specifically, it will be investigated whether there is a correlation between the PageRank Scores of one period and the next. There will also be research on how the PageRank Scores summed up from the politicians per party correlate to the number of seats earned in the upcoming elections. Based on the PageRank scores in the prior period, it will be determined whether one can say something useful about the chances of a politician becoming a minister in the upcoming period. The data comes from the Dutch House of Representatives. The network on which the PageRank scores are based is a network containing links between politicians if they filed a motion together, where weights are adjusted based on the number of motions filed together. Four variations of the PageRank algorithm are used: the Standard Random Walk PageRank, Weighted PageRank, Personalized PageRank, and Weighted Personalized PageRank. The number of motions handed in per politician will be used to obtain the Personalized and Weighted Personalized PageRank Scores.

Keywords: PageRank, Dutch House of Representatives, Python, Networkx, Parties, Politicians, Ministers.

2 Introduction

Larry Page and Sergey Brin started working on a searching engine in 1996, which worked based on links from websites on the internet, which they first called Backrub [1]. The name Backrub was later changed to Google, which is a name everyone knows today. The name Google is derived from a mathematical term called 'googol', which equals 10^{100}. This enormous number represents the vast amount of data available worldwide that Google tries to order. The founders of Google introduced PageRank to objectively say something about the relative importance of web pages [2]. It is a clever algorithm that has since then not only been used to rank websites given a search input. An example where PageRank is used for a different purpose is where PageRank was used to distinguish phishing websites from genuine websites, which back in 2012 already had a 98% accuracy [3]. That is still an example based on web pages, but PageRank can be used in a much broader sense. It can even be used in sports. PageRank can be used to measure performances in ball sports, like Basketball. The algorithm can be applied not only for the team performance or the coaches' performances but also for individual players. It is possible to make a ranking that is not only based on the traditional statistics, like points scored, by including extra data such as partaking in a successful play [4].

In this report, research will be done on a social network. It is always interesting to analyze a network and determine what node is the most important or most central. In a social network, this node could mean the most influential person. As for economics networks, tons of network analysis applications are already widely used. One can, for example, process a change over time in economic transactions between nations to study changes in the world economic system [5]. Somewhat closer to the subject of this report is the use of social network analysis in American Politics [6], which shows that the political sides of social networks are also valuable.

This report will investigate whether PageRank can help predict Dutch political networks based on motions handed in together by politicians. Data is collected from 09-09-2008 up until 10-05-2022, not just the motions filed together by politicians in the House of Representatives but also additional data that can help predict and explain the networks and analysis results. The dates of the elections, the installation dates of newly appointed politicians, the seat distributions, and the coalitions throughout the years. All this data will be combined to make sense of the network and answer some questions. Can PageRank be run on a weighted network that represents motions handed in together by politicians over a period before the 2017 elections to predict the outcome of the seat distribution? Or can one perhaps distinguish the importance of politicians based on their PageRank scores in past periods?

Various hypotheses will first be named and elaborated. The goal of having various hypotheses is to eventually help form a conclusion on how helpful PageRank

is in (these specific) political networks. After the hypotheses are explained in section 3, the reader will be informed of the data used in this research. Section 4.1 will elaborate on where the data comes from, how it was gathered, and what precisely it represents. It will also be explained how the data was made ready for the hypothesis testing, and some numbers will be shown to demonstrate how big the network became. After going through the data, some variations of the PageRank algorithm will be explained in section 4.2. This paper uses four types of PageRank: Standard Random Walk PageRank, Personalized PageRank, Weighted PageRank, and Weighted Personalized PageRank. They will be explained one by one by showing an example of a network and its corresponding PageRank scores. After the examples, it is time to get into the maths. Some math knowledge may be required to understand the algorithm correctly. However, hopefully, one can still grasp the basic idea of the calculations even without prior knowledge after reading section 4.2.5. There will be an example calculation for the first iteration of the Weighted Personalized PageRank algorithm to provide a feeling of how the algorithm works. After obtaining the necessary knowledge, it is time to show what coding was needed to get results for the hypotheses. In this section, which is section 4.3, apparent problems will be mentioned when trying to work with the data. It will also be explained how some results were obtained. Afterwards, the results will be presented and elaborated on shortly in section 5. Next, in section 6 there is a discussion section in which it will be reviewed what may have been coded differently or what other research could lead to some interesting insights. At last, the conclusion will be presented in section 7.

3 Objective

The project's primary goal is to investigate whether one can use PageRank to say something useful about the importance of a politician and whether PageRank can be used to predict changes in the network over time. Multiple hypotheses are set up to analyse whether that is the case.

Hypothesis one investigates whether a politician manages to take a more central position in the network after being active in a previous period. Are the different types of people a politician filed motions with together in the last period of importance on how central a politician will get in the network in the new period? Whether the number of motions handed in per politician means something to their PageRank scores will also be tested. The first hypothesis is as follows:

Hypothesis 1: There is a strong correlation between the PageRank score of a politician in a previous period and the politician's current PageRank score.

The following hypothesis is about finding out whether there exists a connection between the PageRank scores of politicians and their corresponding parties regarding seat distribution. Worded differently; are the parties where the sum of the PageRank scores of their politicians is higher favoured by voters? The second hypothesis is:

Hypothesis 2: There is a strong correlation between the number of votes earned by a party and the summation of the PageRank score of the corresponding politicians in the period before the elections.

The third and final hypothesis dives into whether the PageRank score of politicians can be used to see whether a politician will be rewarded with a role as a minister. The idea is that if someone manages to claim a central position in the network, he or she becomes more powerful and thus a candidate for a ministry. That is only if the politician is a party member of a party in the coalition. Otherwise, one is very unlikely to become a minister. It must be mentioned that this hypothesis is less extensively dealt with, as the other two hypotheses were already filling up the time available for this project. The hypothesis is as follows:

Hypothesis 3: There is a strong correlation between the PageRank score of politicians in a prior period and the chances of getting a role as a minister in a new cabinet.

4 Methodology

4.1 The Data

To answer the hypotheses and help form the main question's conclusion, data is needed over a more extended period from which a political network can be obtained. Such a political network could, for example, be a network in which politicians are linked if they worked on a motion together, which is precisely what is done for this research. In other words, let A be the adjacency matrix of the graph, where A takes a $n \times n$ shape with n being the total amount of politicians in the network. Now A_{ij} equals 1 if it happens to be the case that politician i and j filed in a motion together. Then that would also mean that A_{ji} will be 1, as there cannot be one-sided cooperation in handing in a motion. A_{ij} and A_{ji} are set to 0 if the politicians did not work together on any motion within the data set. To build the network, data is obtained from the website 'Open Kamer' [7].

Next to having the network, some other data is helpful as well. For instance, the number of seats per party shows the seat distribution within the House of Representatives. Which parties were in the cabinets during the various periods in the data can also be scraped. Furthermore, the dates on which the newly appointed House of Representatives members were installed are stored to distinguish the various periods in the House of Representatives, such that questions can be asked about changes over time. For all politicians, their corresponding parties are stored to answer the questions where the cabinet plays a role. Also, if it happens to be the case, the minister's role for every politician will be stored.

4.1.1 Collecting the Data

The website needs to be scraped because there is no download button available on 'Open Kamer'. The so-called 'Web Scraper'-tool was used to achieve this [8]. With the tool, it is possible to go through each page of the website, go through all the motions, amendments, and bills, and collect all the information needed on those separate pages. For the research in this paper, no further distinction will be made between the motions, amendments, and bills. Because all three of them can show collaboration between politicians and more data is helpful to have more differences in the number of motions, amendments, and bills filed per politician, this decision was made. Therefore, from here on, they will all be referred to as motions, which was already done in the paragraphs before this one.

When scraping for a specific motion, various data is obtained. An example of what a motion looks like on the website is shown in figure 1. The date is shown in the top-left corner, in the format of Year-Month-Day. Whether the motion was accepted or not is indicated with a symbol in the top-right corner; in the example of figure 1, the motion is rejected by the House of Representatives. The name of the motion is stored right under the text 'Motie' in the

centre of the top. Underneath the name is stored which parties voted in favour ('Voor') and against ('Tegen') the motion or the parties that did not vote at all ('Niet Gestemd'). This example shows that some members of the House of Representatives are acting independently, like Nilüfer Gündogan [9]. The names at the bottom of the figure show the submitters of the motion, where it is also indicated on behalf of which party they operate.

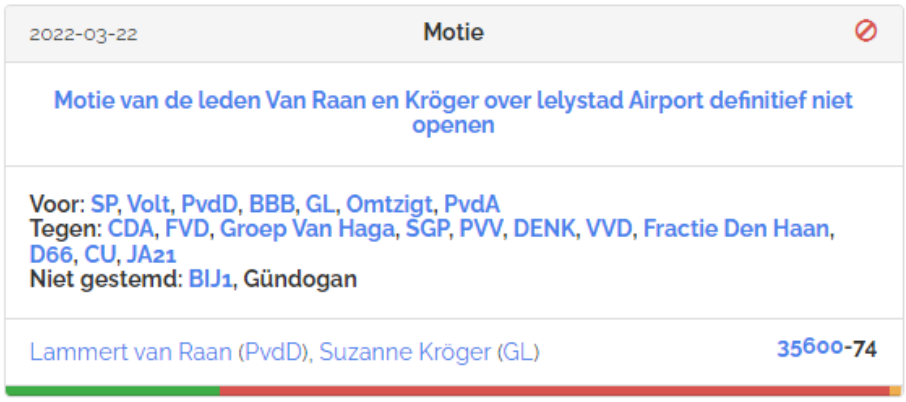


Figure 1: Example of a motion from 22-03-2022

The first step in scraping a motion is to take the date when the motion was filed. This data is needed later on to specify specific periods. On the website 'Open Kamer', data is available from 09-09-2008 up until the present. The percentage of votes in favour per motion is also taken because, with that information, one can check whether a motion passed. A motion passed if more than 50 per cent of the votes were in favour. Next to that, the name of the motion will be taken so that it can be assured that there are no duplicates. Lastly, the submitters of the motion are stored. Sometimes, only one person submits a motion, and only one name is stored. In other cases, groups of different people worked together. Then, all names are stored. Together with the names, their parties will also be stored (if they are not working independently within the House of Representatives), and, if it is the case, their function as minister will be stored. The data is formatted in a CSV file from which the cleaning will start with the help of a Python script.

Information about when the members of the House of Representatives were installed is found on the website of the House of Representatives itself [10] by tracing back how many days the current members have been in the House of Representatives. The seat distribution of the House of Representatives throughout the years ('zetelverdeling' in Dutch) is found on CBS [11] and Parlement.com [12]. The information about the various cabinets throughout the years can be

found on the website of Rijksoverheid [13].

The following installation dates are obtained, which will thus indicate the various periods: 30-11-2006, 17-06-2010, 20-9-2012, 23-03-2017, 31-03-2021. As mentioned before, the data on 'Open Kamer' starts on 09-09-2008, so the first period is incomplete. The following table contains the information about the seat distribution within the periods mentioned:

Seat distribution in the House of Representatives					
Party	2006	2010	2012	2017	2021
PvdA	33	30	38	9	9
CDA	41	21	13	19	15
VVD	22	31	41	33	34
D66	3	10	12	19	24
GroenLinks	7	10	4	14	8
SP	25	15	15	14	9
SGP	2	2	3	3	3
ChristenUnie	6	5	5	5	5
PVV	9	24	15	20	17
PvdD	2	2	2	5	6
50Plus	0	0	2	4	1
DENK	0	0	0	3	3
FVD	0	0	0	2	8
Volt	0	0	0	0	3
JA21	0	0	0	0	3
BBB	0	0	0	0	1
Bij1	0	0	0	0	1

The various cabinets and the dates they started are the following:

Cabinets in the House of Representatives	
Date	Parties involved
22-02-2007	CDA, PvdA, ChristenUnie
14-10-2010	CDA, VVD
05-11-2012	PvdA, VVD
26-10-2017	CDA, VVD, D66, ChristenUnie
10-01-2022	CDA, VVD, D66, ChristenUnie

4.1.2 Cleaning the Data

As the scraper tool is not that comprehensive, much cleaning is left to do. The data is currently not usable for testing. For instance, some data is missing on the website. There are four motions where not all information that is needed is present. These motions are deleted from the data as they do not add anything

to the research. An example of a motion with missing information is shown in figure 2. All the motions are stored in a Pandas DataFrame structure, meaning

Figure 2: Example of an incomplete motion from 22-12-2011

the Python library Pandas is used. More Python libraries are utilised, which are mentioned at this report's end. With a DataFrame structure, one can easily loop through all the motions to further edit the data to make it ready for hypothesis testing. For instance, submitters' names are not yet in a useable format. The data can look like this:

Indiener(s):
Jasper van Dijk
,
Mahir Alkaya
(SP)

The string is split in such a way that a list is created per motion in which the members' full names are stored. The party of which those politicians are a member is also stored with the names. This makes sure that later on, research can also be done on the network of parties instead of individuals in the form of politicians. With the number of votes in favour of the motion, a dummy variable is created stating whether a motion was approved.

A graph can be created when the adjacency matrix of everyone that has filed a motion together is built. In figure 3, one can see the network for the period of 31-03-2021 up until 10-05-2022, where, as mentioned before, there is an edge between politicians if they filed in at least one motion together, where each node in the graph represents a politician. There are also politicians without edges in the network, as they only filed motions individually. All edges have the same width. As can be seen, it is not a complete graph, making it more interesting for general research and PageRank. For the various periods, where in the table the date means the starting date of the period which goes on until the next date, the following amount of politicians and motions are obtained:

Amount of Politicians and Motions per period		
Period	Politicians	Motions
22-02-2007	180	2660
17-06-2010	196	3186
20-9-2012	236	7022
23-03-2017	222	5943
31-03-2021	169	1402

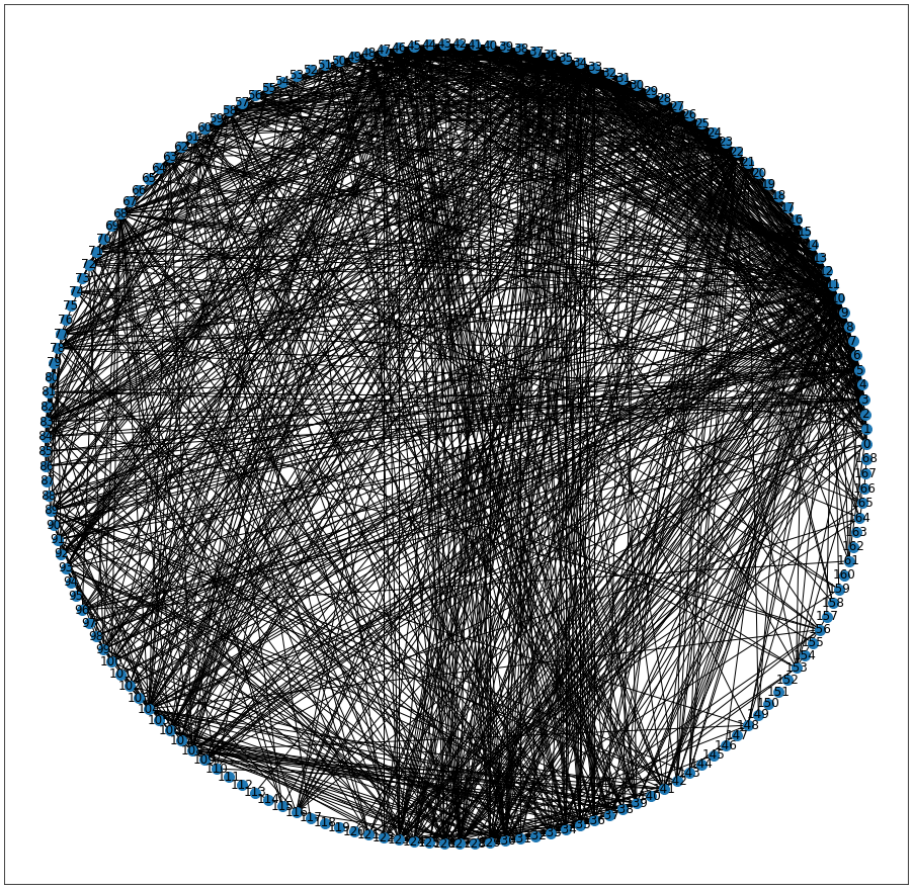


Figure 3: Network from 31-03-2021 up until 10-05-2022

4.2 PageRank

Various variations of the PageRank algorithms exist, all with advantages and disadvantages, depending on the network. In this subsection, the different PageRank algorithms used for this report's analysis will be explained. However, the general idea of PageRank will be explained first.

As mentioned in the introduction, PageRank was in the first place created to measure the importance of various websites. However, it just mathematically objectively measures the centrality of nodes in a network, where nodes can represent anything the network is made of. A simple example will be given on a created data set, which looks like the data that will be used further on in the report, but for simplicity, it will be smaller.

4.2.1 Standard Random Walk PageRank

Imagine a political network consisting of 6 nodes representing politicians. The nodes would share an edge if the politicians filed a motion together. No extra data has been used yet. The network is shown in figure 4, along with the corresponding adjacency matrix. In this example, a standard random walk will be performed, which is one of the ways to calculate the PageRank scores and is overall seen as the simplest form of the algorithm. The mathematics will be explained later on, but what the PageRank scores mean to represent can be described as follows. The algorithm starts at a random node in the graph. For now, this starting node is 'A'. This node 'A' is now visited once, which is something that is stored. From node 'A', the algorithm can go to either node 'B' or 'C' because the edges of 'A' lead to those nodes. A coin is tossed to decide which route is used, where all neighbours of the current node have an equal chance of being visited. The coin favours node 'B', so node 'B' is visited once as well. Node 'B' is connected to nodes 'A', 'D', and 'E', and a coin will be tossed to decide where to go. This goes on and on, and after many runs of the algorithm, the number of visits to each node form a distribution. This distribution can be normalized simply by dividing the number of visits of each node by the sum of visits of all nodes. This is what the PageRank score essentially represents. The PageRank scores for the Random Walk are given in the table below, calculated by the PageRank function included in the Python library NetworkX.

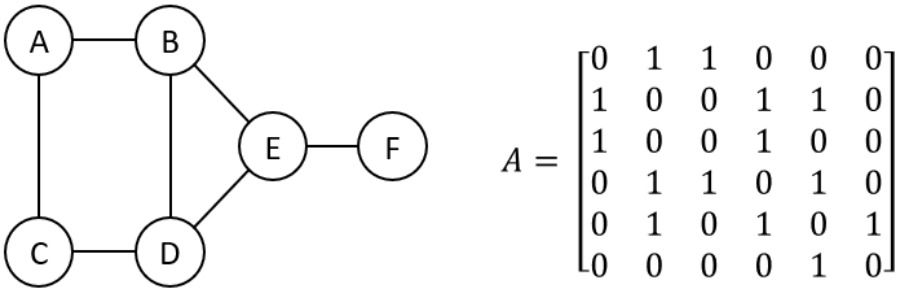


Figure 4: Example Structure Politicians Random Walk PageRank

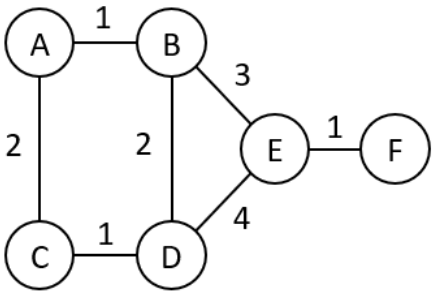
PageRank Scores corresponding to figure 4	
Node	Standard Random Walk PageRank Score
A	0.144675
B	0.205369
C	0.144675
D	0.205369
E	0.214217
F	0.085694

The sum of the scores will add up to 1, potentially slightly deviating based on rounding. This is always the case, such that the position of the nodes is relative to the network; the sum of the scores will never exceed 1, even if the network becomes incredibly large. What can be noticed is that the scores of nodes 'A' and 'C' and nodes 'B' and 'D' are the same. This is because the graph is symmetric right now; the nodes are connected to the same kind of nodes, which again have the same connections, meaning the calculations for those nodes are the same. The scores will always be the same for this network, no matter how many times one tries to run the algorithm because of its mathematics. In this example, there is no distinction in what edges are more important. However, different results are obtained if weights are put on the edges.

4.2.2 Weighted PageRank

Weights, in this case, could mean the number of motions the politicians worked on together. Assume that politicians A and C submitted two motions together. Now the weight of the edge going from A to C becomes 2 instead of 1. Further changes for the edge weights are made, visualized in the new graph and adjacency matrix in figure 5. A new situation is created in which some nodes

now have more important connections than others. Again, the mathematics will be explained later on; for now, it is only necessary to realize that weights can affect the scores drastically, and here is the reason. Consider again the explanation of what the PageRank scores represented by the example before weights were added. The algorithm flipped a coin to decide what node to visit next, and it still works like that, only the coin is now favoured towards edges with higher weights. Imagine the algorithm being at node 'A', the chances of going to node 'B' are now $\frac{1}{3}$, as the sum of the weights from node 'A' are 3 and the weight of the edge going from 'A' to 'B' is 1. This way, nodes with important edges are favoured over nodes with less important edges, thus rearranging the PageRank score distribution in the graph. This adjacency matrix will yield the following PageRank scores once Weighted PageRank is performed:



$$A = \begin{bmatrix} 0 & 1 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 & 3 & 0 \\ 2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 & 4 & 0 \\ 0 & 3 & 0 & 4 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Figure 5: Example Structure Politicians Weighted PageRank

PageRank Scores corresponding to figure 5		
Node	Standard Random Walk PageRank Score	Weighted PageRank Score
A	0.144675	0.123132
B	0.205369	0.201455
C	0.144675	0.122809
D	0.205369	0.230874
E	0.214217	0.268231
F	0.085694	0.053500

It is interesting to see that nodes A and C and nodes B and D no longer have identical scores. Both 'A' and 'C' got lower scores, where 'A' is now slightly more central than 'C'. Node 'B' also got a lower PageRank score, while node 'D' went up. 'E' also got a higher score, as both 'D' and 'E' got more important links than all other nodes. Node 'F' already had the lowest score in the Random Walk algorithm, but the node got an even lower score now as it only has one

edge with a small weight compared to other edges.

4.2.3 Personalized PageRank

One more alteration can be made to decide to which node the algorithm goes. As mentioned before, the original purpose of the algorithm was to rank websites by following links to different websites. There was a problem that some websites do not have external links leading to other pages, meaning that the algorithm could get stuck. In a graph, those websites would be nodes without outward links. Those nodes are called dangling nodes [14]. In the example of a graph representing politicians who worked together on a motion, dangling nodes cannot exist, as there cannot be a one-sided collaboration on a motion, but what can happen is that there are politicians in the network who are disconnected from the main network. A clever solution exists to be still able to reach all politicians. Before deciding what edge to take from the current node, another decision has to be made. Again, a coin is flipped, where there are two options. Option one is to choose an edge from the current node, doing the same thing as before. The other option is to jump to a random node in the network. The chances for this coin are decided by a so-called damping parameter a, where usually $a = 0.85$ [15]. This means that there is an 85% chance of taking an edge from the current node and a 15% chance of jumping to another random node, which does not have to be connected to the current node. By default, after having decided to jump to a random node, the chances of reaching node 'A' are $\frac{1}{N}$, where N is the number of nodes in the network. However, one can alter the odds of going to specific nodes so that the distribution is no longer uniform. To stay with this example, the chances of reaching a node could be decided by taking the number of motions the politician worked on divided by the sum of all motions handed in. Changing those odds is what is done in Personalized PageRank. For context, the damping parameter was 0.85 for the Random Walk and Weighted PageRank scores above. This value will always be used throughout this report as the damping parameter. Consider the graph in figure 6 together with the according adjacency matrix and a vector representing the motions handed in per politician, which in this case is the out-degree per node. The PageRank scores are given in the table below. Note that the Weighted PageRank Scores are still the scores according to figure 5 and that the weights are set to 1 for the Personalized PageRank Scores.

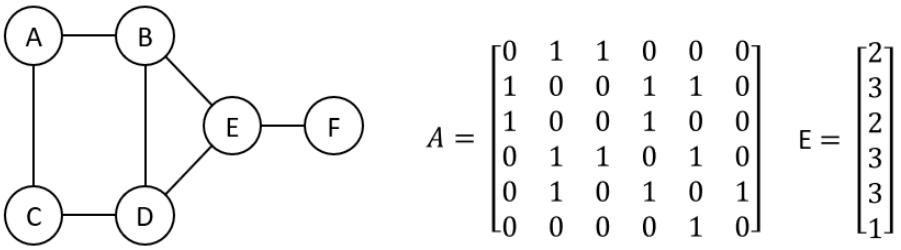


Figure 6: Example Structure Politicians Personalized PageRank

PageRank Scores corresponding to figure 6			
Node	Standard Random Walk PageRank Score	Weighted PageRank Score	Personalized PageRank Score
A	0.144675	0.123132	0.142857
B	0.205369	0.201455	0.214285
C	0.144675	0.122809	0.142857
D	0.205369	0.230874	0.214285
E	0.214217	0.268231	0.214286
F	0.085694	0.053500	0.071428

Notice how nodes 'A' and 'C', and 'B' and 'D' again have the same score, as weights are not taken into account, meaning that the graph is symmetrical again. Compared to the Random Walk PageRank scores, nodes 'A', 'C' and 'F' got a lower score, and 'B', 'D' and 'E' went up, although the differences for nodes 'A', 'C' and 'E' are almost negligible. The algorithm behaves as expected, as the chances of visiting nodes 'B', 'D', and 'E' is increased by the manipulation of the random jumps. There is one final manipulation that will be done in this research to get the fourth and final alteration of the PageRank algorithm.

4.2.4 Weighted Personalized PageRank

The Weighted and Personalized PageRank algorithms will be combined to create a Weighted Personalized PageRank algorithm. Combining these algorithms means that the weights of the edges will not consist of the same values, and the jumps to random nodes will not be completely uniform. Consider figure 7, where the weights are the same as in figure 5, and the personalization vector is changed accordingly to contain the number of motions handed in per politician, matching the out-degree. The newly obtained Weighted Personalized PageRank scores are in the table below, where the scores of the previous figures are as well.

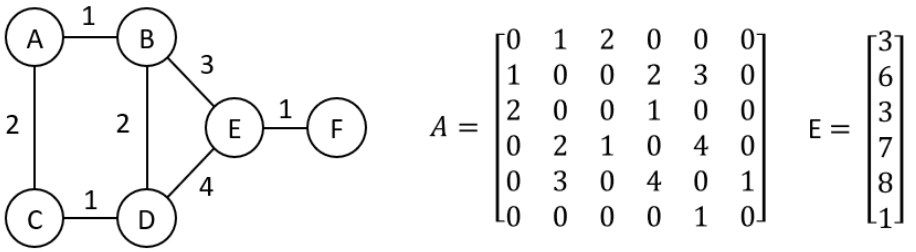


Figure 7: Example Structure Politicians Weighted Personalized PageRank

PageRank Scores corresponding to figure 6				
Node	Standard Random Walk PageRank Score	Weighted PageRank Score	Personalized PageRank Score	Weighted Personalized PageRank Score
A	0.144675	0.123132	0.142857	0.107143
B	0.205369	0.201455	0.214285	0.214286
C	0.144675	0.122809	0.142857	0.107142
D	0.205369	0.230874	0.214285	0.250001
E	0.214217	0.268231	0.214286	0.285714
F	0.085694	0.053500	0.071428	0.035714

It can be noticed from the table that nodes 'D' and 'E' got an even more central position within the network according to Weighted Personalized PageRank compared to all other versions of the algorithm applied before. Later in this report, all four different algorithms will be used to see which will help provide the best answer to the various hypotheses. Now onto the mathematics.

4.2.5 Mathematics behind PageRank

There are various ways of writing down the formula of PageRank. Notice that thus far, an undirected graph was used in the example figures. Officially, PageRank was created for directed networks, but any undirected graph can easily be changed into a directed graph by splitting the edge of the undirected graph in two; if in an undirected graph, there is an edge between nodes 'A' and 'B' with weight 3, then in a directed graph two edges will be created. One goes from node 'A' to 'B' with weight 3, and the other goes from 'B' towards 'A', again with weight 3. After explaining the formula, an example of an iteration will be given to show how the algorithm's calculation works. Here, the following mathematical recursive equation will be used to explain what the PageRank scores have to comply with for the Standard Random Walk algorithm for a directed graph:

$$R_i = \alpha \sum_{j=1}^{N} \frac{A_{ji} R_j}{k_j^+} + (1 - \alpha) E_i. \tag{1}$$

Every term can be explained separately. The PageRank vector itself is R, where R_i depicts the PageRank score of node i. The term α, where $0 < \alpha \leq 1$, is the damping factor. This factor represents the chance of not jumping to a different random node in the network. Again, as mentioned before, in the examples given earlier in this chapter, the damping factor a was set to 0.85, as it is widely used as the default value. The next part is the summation of the equation, which is

$$\sum_{j=1}^{N} \frac{A_{ji} R_j}{k_j^+}. \tag{2}$$

The summation essentially sums over all nodes j pointing to node i, where $A_{ji} = 1$ if node j has an edge going to node i. The PageRank score of node j is depicted by R_j, meaning the algorithm takes into account from which node the link is coming to increase the score if it comes from an already important node. In other words, a connection to a central node j increases the PageRank score of node i more than a connection not that connected to the network. The out-degree of node j is k_j^+, meaning the total amount of edges leaving node j. The next bit of the recursive equation is

$$(1 - \alpha) E_i. \tag{3}$$

This term shows the chances of randomly jumping to another node in the network, where α is again the damping factor, E_i is the chance of reaching a specific different node, which in the basic form of PageRank equals $\frac{1}{N}$ for every node, with N being the total amount of nodes in the network. This means that each node is just as likely to be reached. This vector can be changed, as was also explained before in the examples, which leads to Personalized PageRank.

The base formula can be rewritten if the following term is introduced:

$$\bar{A_{ij}} = \begin{cases} \frac{A_{ij}}{k_i^+}, & \text{if } A_{ij} = 1 \\ 0, & \text{else} \end{cases}. \tag{4}$$

This term transforms the adjacency matrix into a row normalized adjacency matrix. That means that for every row in the matrix, the sum of the values adds up to 1. Now when replacing A_{ji} with this new term in the base formula, the following formula is obtained:

$$R = (\alpha \bar{A} R + (1 - \alpha) E). \tag{5}$$

And because the vector is now already row normalized, the formula can be rewritten as:

$$R = (\alpha \bar{A} + (1 - \alpha)1^T E)R. \tag{6}$$

From this final formula, it can be concluded that the PageRank vector is simply an eigenvector with eigenvalue 1. That means it has a unique solution to the problem, meaning the PageRank scores of a network will always be the same if the network remains the same.

An example of the calculation will now be given for Weighted Personalized PageRank, as this is the most comprehensive calculation. Consider again the graph in figure 7. The terms $\bar{A}$ will be row-normalized, and E will be changed accordingly. The result can be seen in figure 8.

The PageRank algorithm is recursive; it considers the results of the previ-

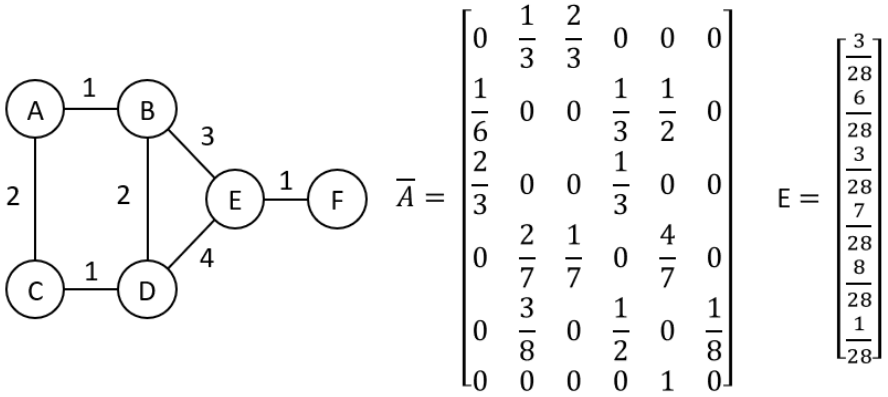


Figure 8: Example Structure Politicians Calculation

ous run. For the first iteration of PageRank, the calculation to get the score of node 'A' by using equation 5 and the newly obtained matrix and vector presented in 8 is as follows:

$$\begin{aligned} R^0(A) &= \frac{1}{N} = 0.167 \\ R^{t+1}(A) &= (a(R^t(B) * \bar{A}_{BA} + R^t(C) * \bar{A}_{CA}) + (1 - a) * E_A)) \\ R^1(A) &= (0.85(\frac{1}{6} * \frac{1}{6} + \frac{1}{6} * \frac{2}{3}) + (1 - 0.85) * \frac{3}{28}) = 0.134. \end{aligned} \tag{7}$$

In the calculations for the successive iterations, $\bar{A}$ and E remain the same, but R is updated with the newly obtained values, such that for other nodes in the calculation, if $R(A)$ is taken, the value will be 0.134 instead of $\frac{1}{6}$. The scores of the other nodes will also be calculated in every iteration; otherwise, the new

score of node 'A' cannot be determined as the calculation uses the PageRank scores of the other nodes. The PageRank vector after zero, one, and two iterations are given in the table below.

PageRank Scores per iteration corresponding to figure 6			
Node	Iteration 0	Iteration 1	Iteration 2
A	0.167	0.134	0.112
B	0.167	0.173	0.227
C	0.167	0.131	0.117
D	0.167	0.203	0.266
E	0.167	0.336	0.235
F	0.167	0.023	0.041

In some cases, there is a big difference in scores between an iteration. At first, these differences will be big, but they will get smaller and smaller. Eventually, if the difference becomes so small that it is almost negligible, it is said that the PageRank vector has converged. In a more mathematical sense, it is clearly explained by Mihalcea, R., & Tarau, P. (2004, July) [16]; There is an error rate for vertex (node) V_i, which represents the difference between the actual PageRank score $S(V_i)$ and $S^k(V_i)$, which is the PageRank score at a certain iteration k. The problem is that the actual PageRank score is yet unknown. The difference between two successive iterations will be taken to solve that issue. The equation for this is $S^{k+1}(V_i) - S^k(V_i)$. If that value drops below the given threshold, the vector has converged. By default, this threshold is set to 0.000001 by the PageRank algorithm provided by Networkx in Python, which is also the value used in this report.

4.3 Combining the Data and PageRank

Before jumping into how the coding for the specific hypotheses has been done, it will be explained how the PageRank scores per period are obtained. These PageRank scores are used for all hypotheses. If the data is fully prepared, it is not hard to calculate the PageRank scores for the various algorithms. As was explained before, an adjacency matrix was created, with edges representing politicians that filed a motion together. The weights of these edges differ based on how many motions the politicians worked on together. For the Standard Random Walk PageRank algorithm, the weights are disregarded. The adjacency matrix for every period is given to the Networkx PageRank algorithm, with weights being 1 if there is a connection. The values are stored for every politician in a column in a Pandas DataFrame for the period. In this DataFrame, the Personalized, Weighted, and Weighted Personalized PageRank scores will be added to a new column. To obtain the Personalized PageRank scores, the adjacency matrix is given to the algorithm again. However, now a dictionary is added that stores for every politician how many motions they have filed. That is all the information that the Personalized PageRank Scores will be based on in this report. With this dictionary, the Networkx algorithm will automatically transform the data into a helpful vector and calculate the scores. For Weighted PageRank, all one has to do is say that the weights of the network must be taken into account, and again the algorithm will ensure it is in the correct format. For the Weighted Personalized PageRank, the dictionary containing information about how many motions a politician has filed and the weights in the network are given to the algorithm. The DataFrames per period now contain all the information they need to contain to start working on hypothesis 1. Testing will be done with the 4 algorithms and the outcomes of all 4 algorithms will also be included in the results. This way, the various algorithms can be compared.

4.3.1 Coding for Hypothesis 1

The hypothesis was as follows: There is a strong correlation between the PageRank score of a politician in a previous period and the politician's current PageRank score. This hypothesis is straightforward regarding programming. All one needs are the various PageRank scores per algorithm per period, which are already available. One thing that still needs to be done is to remove data points (politicians) from the data where no PageRank score was available in either of the two periods that are to be compared because comparing their PageRank scores where one of the scores is 0 will mess up a correlation, if there is one.

To better grasp how the PageRank scores are calculated, it will also be checked whether the number of motions filed per politician and the number of motions filed in the previous period play a significant role in their PageRank scores. This data was also already available, as the dictionary containing the number of motions filed per politician per period was already needed earlier. Taking the data of motions filed in the previous period cannot be done for the first and

last period, as no data exists for the period before the first period about how many motions were handed in per politician. Also, the PageRank scores for the following period cannot yet be calculated for the last period. Plots are created to see whether there exists a correlation.

4.3.2 Coding for Hypothesis 2

Onto the second hypothesis, which is the following: There is a strong correlation between the number of votes that a party gets and the summation of the PageRank score of its politicians in the period before the elections. A problem that instantly arises for this hypothesis is that it is impossible to account for parties that did not have a seat in the House of Representatives but who do enter the elections. No PageRank scores of those parties are known unless the party has members who left their original party halfway through the previous period, as the seats in the House of Representatives are distributed to people who can choose to leave their party and go further as an independent fraction [17]. In the last elections, eight seats were earned by new parties, as is shown in section 4.1.1, and five seats were appointed to new parties in the 2017 elections. If a plot is made for this data where on one axis the PageRank scores are drawn and on the other the number of seats earned, those data points would be stuck on the axis representing the PageRank score. Nothing can be done about this problem without having extra data about the new parties.

For the coding, several data is needed. First, the various PageRank scores per period per politician are needed. Fortunately, those are already available as they were also needed for the previous hypothesis. How those scores were obtained will not be explained again. Secondly, data is needed about which party politicians represented during the specific periods. Sometimes it happens to be the case that people step out of a party. The data did have some flaws regarding what party politicians represent when filing a motion. If a party was mentioned, it was the correct party, but quite often, no party is mentioned. For this research, it is decided that politicians are appointed to the party they filed the most motions for during the specific period. Suppose it happens to be that the most occurring party for a politician is 'nan' in the data, but the second most occurring is a valid party. In that case, the politician is assigned to this second party, as that prevents more loss of data. Now that parties are assigned to the politicians, an algorithm is coded that sums up all the scores of the politicians per party. With this data preparation, the first proper plot can be created, which is a plot that shows the sum of the PageRank scores and the seats. Such a plot helps determine whether there is a specific pattern between the variables.

Because the PageRank scores should sum up to 1 and are thus a division, a prediction of the seat distribution could be calculated by multiplying the PageRank scores per party by 150, as there are 150 seats available. This prediction will only be created for the 2017 elections. There is, of course, a very slight chance

of succeeding in making a solid prediction, but it is still interesting to see how it turns out. However, the sum of all the PageRank scores per party combined turned out to be less than 1, as some politicians were removed from the data because they did not represent a party. For that reason, the formulation for the prediction of seats is as follows:

$$\text{seats for party A} = \frac{\text{Sum of PageRank scores of party A}}{\text{sum of PageRank scores of all parties}} * 150. \tag{8}$$

This formula assumes a linear relationship, which might not be the case. Nevertheless, because this is not the main objective of the hypothesis, there will be no further experimentation with other formulas. The total amount of seats given away in the predictions will be 150 if the outcome of the equation is rounded. This would still not be an entirely fair approximation, as this does not account for the new parties. However, with the data about the last period, it would be impossible to predict how many new parties would gain ground in the elections. Sometimes the data needs some tweaking by hand. For example, in the period 2012-2017, there are still PageRank scores for the party 'GPV', which was already fully merged with the ChristenUnie in 2003 [18]. For obvious reasons, such scores will be added to the new party. The results are presented in a bar plot, where the predictions are given in orange and blue, and the actual outcome of the elections is in red. There are two predictors, as not only the data will be taken over the whole period of 2012 up until 2017, but also just the last year before the elections will be taken. This is to ensure that there are no significant changes in the network during the last year. The mean squared error will be taken for the various PageRank algorithms to measure how good the predictions are.

4.3.3 Coding for Hypothesis 3

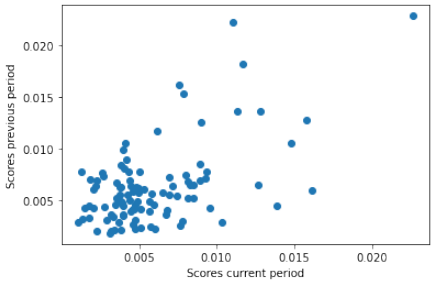
The third and final hypothesis was: There is a strong correlation between the PageRank score of a politician in a prior period and the chances of getting a role as a minister in a new cabinet. Here, the data also needs further preparation to dive into testing and plotting. As explained in section 4.1, in the original data, whether a politician had a function as a minister while filing a motion is also stored. This data is used to find all politicians who, at one point, were ministers. To be more precise, the names of the politicians are stored together with the dates when they were first mentioned as a minister in the data. That way, it can be seen in which specific period the politician became a minister for the first time, and the PageRank score of this politician can quickly be taken from the period before. Now that the minister's data are ready, more data is needed about the other politicians. Otherwise, one cannot determine whether there is a difference in PageRank scores between a minister and a 'regular' politician within the House of Representatives and the coalition. As was explained before, it is improbable to become a minister if one is not part of the coalition. Because

every politician's party was already stored for hypothesis number 2, all that needs to be done is to check whether that party was in the coalition for that specific period. Once all the data is ready for all the periods, plotting can begin.

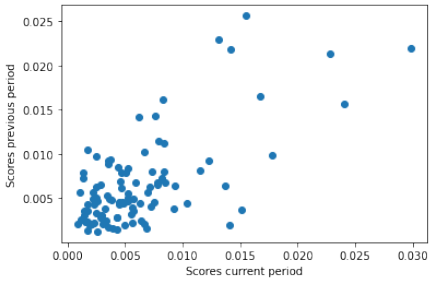
5 Results

5.1 H1

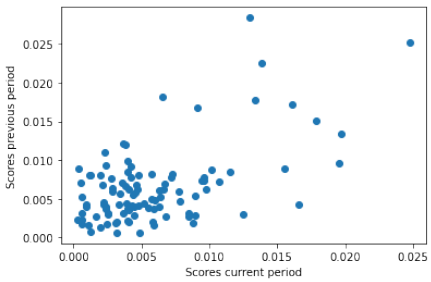
The first plots that will be shown compare the PageRank scores in a period with the previous ones. This can be seen in figure 9. The PageRank scores of the current period are on the x-axis and the PageRank scores of the previous period are on the y-axis. It seems like the data is not normally distributed. A histogram of the PageRank Scores for the same period is created for good measurement, as shown in figure 10. All plots in this section for the results of hypothesis one will be about the period from 2010 to 2012. The results for the other periods will be included in a table.



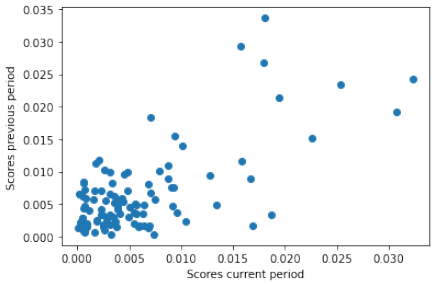
(a) Standard Random Walk



(b) Weighted



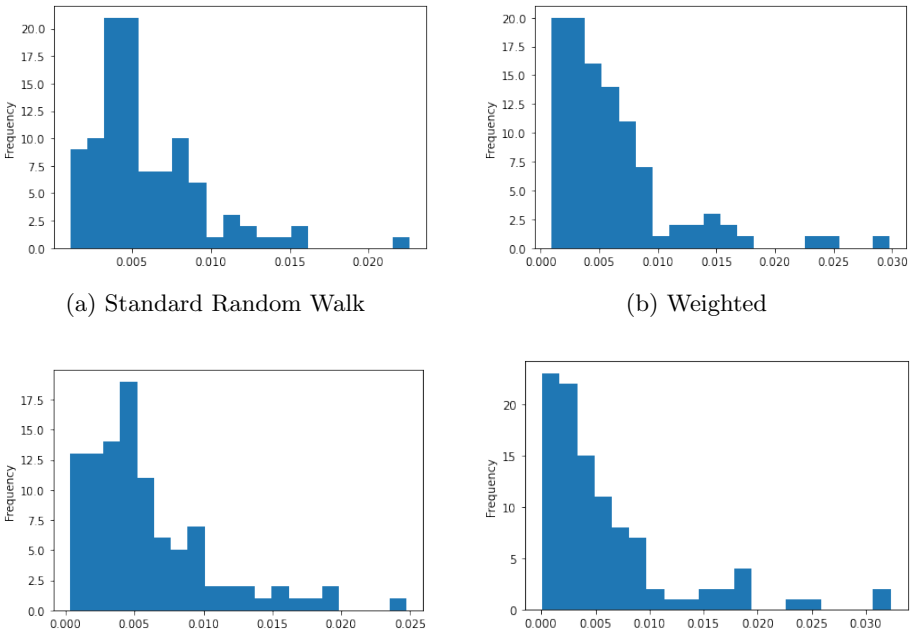
(c) Personalized



(d) Weighted Personalized

Figure 9: PageRank Scores 2010-2012 versus PageRank Scores of the previous period

By looking at the scatter plots in figure 9, it seems there is no real correlation at hand, but if a correlation exists, it might just be linear. An indicator of the strength of the linear relationship between two random variables is the Pearson Correlation Coefficient [19]. This coefficient would be helpful to check how good the correlation between the two PageRank scores per period is. However, the data is thus not normally distributed, which is one of the assumptions for a



(a) Standard Random Walk

(b) Weighted

(c) Personalized

(d) Weighted Personalized

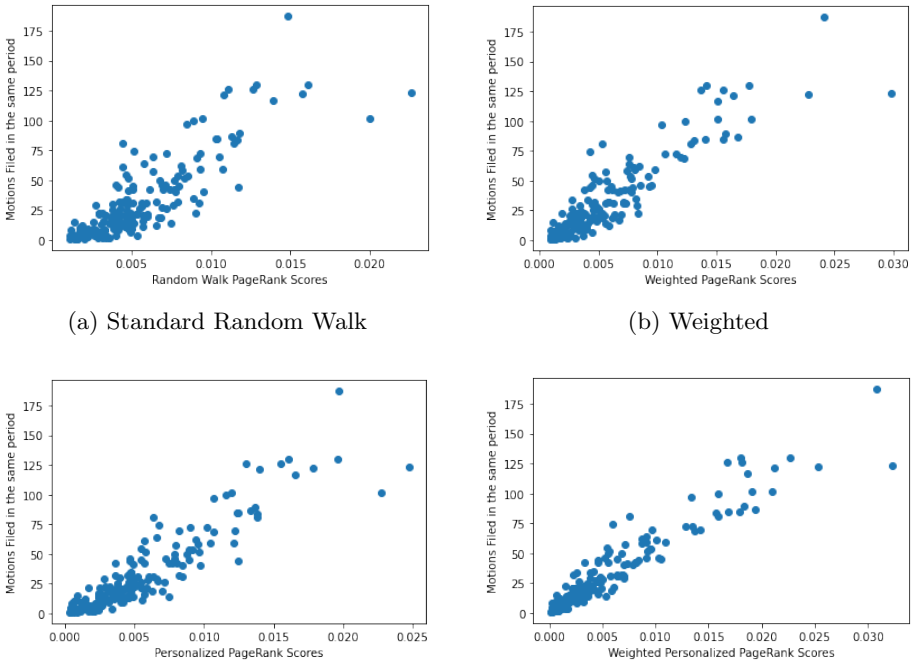
Figure 10: PageRank Scores 2010-2012 distribution

Pearson's Correlation Coefficient test. However, studies suggest that assumption violation has little effect on the outcome [20]; thus, this paper will still use Pearson's Correlation Coefficient. The value of the coefficient is always between -1 and 1. A -1 indicates a perfect negative linear relationship and a 1 a perfect positive linear relationship. The closer the coefficients get to 0, the less of a linear relationship between the two tested variables. When a correlation can be considered strong is up to debate. Different authors use different interpretations of the coefficient, where some say the correlation is already strong after reaching an absolute value above 0.5, and others stick to a higher value of 0.8 [21]. Considering that the normality assumption is violated for this report, it will be said that a correlation is strong only after reaching a higher absolute value than 0.8.

The Pearson Correlation Coefficient for the Weighted Personalized PageRank algorithm in figure 9 is 0.671. The relation between the two PageRank scores is thus moderately linearly positive but not strong enough to say that the previous PageRank score can be seen as a solid predictor for the PageRank score in the next period, as there is no strong linear correlation. The coefficients for the other algorithms per period will be presented in a table.

To explain the results from the previous plots a bit better, the plots displayed in figure 11 are made, where the PageRank scores are plotted against the number of motions handed in in the same period. After looking at the plots, it becomes clear that there is a correlation between the PageRank scores and the number of motions filed. The Pearson Correlation Coefficient for Weighted Personalized PageRank equals 0.950, suggesting a strong positive linear relationship between the two. It just so happened to be the case that the correlation between motions filed in the same period and motions filed in the last period has a way lower Pearson Correlation Coefficient, with only 0.125. The plot for comparing the amount of filed motions can be seen in figure 12.

The PageRank comparisons between the two periods probably fail to get a strong correlation because the PageRank Scores seem strongly influenced by the number of motions filed in that period per politician, and there seems to be no strong correlation between the number of motions filed per politician for these periods. Whether there is a causal relationship was not investigated, as it is not needed to form a conclusion for this hypothesis. In the tables below, the various Pearson Correlation Coefficients are given per algorithm per period for the PageRank scores in the current period and the period before, and the averages are given.



(a) Standard Random Walk

(b) Weighted

(c) Personalized

(d) Weighted Personalized

Figure 11: PageRank Scores 2010-2012 vs number of motions filed in the same period

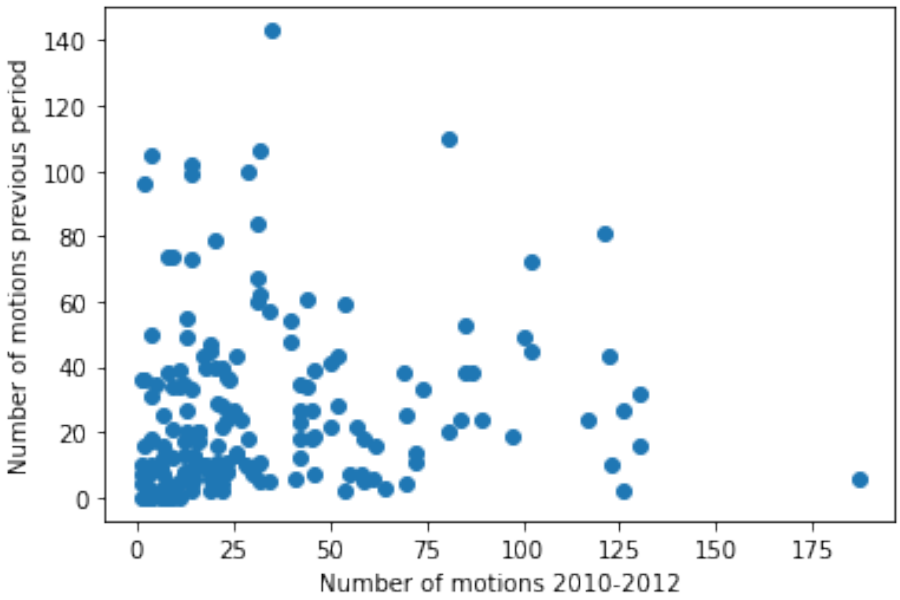


Figure 12: number of motions filed for the period 2010-2012 vs number of motions Filed for the period before 2010

Pearson's Correlation Coefficient for PageRank Scores comparisons with previous periods				
Period	Standard Random Walk	Weighted	Personalized	Weighted Personalized
2010-2012	0.564	0.671	0.571	0.671
2012-2017	0.729	0.751	0.754	0.751
2017-2021	0.364	0.297	0.388	0.317
2021-	0.536	0.548	0.645	0.629
Averages	0.548	0.567	0.590	0.592

From this table, it appears that, on average, the Weighted Personalized PageRank algorithm yields the best correlation, but it is not a very strong correlation, with only a value of 0.592. Notice also that for the 2010-2012 and 2012-2017 periods Weighted PageRank performs as well as Weighted Personalized PageRank, and for the periods 2012-2017, 2017-2021, and 2021- Personalized PageRank performs best. Hypothesis 1 was "There is a strong correlation between the PageRank score of a politician in a previous period and the politician's current PageRank score", but this hypothesis turned out to be false. With the highest Pearson's Correlation Coefficient on average for an algorithm of 0.592, one cannot conclude that there is a strong correlation between the PageRank scores.

5.2 H2

With the seats per party, a plot can be made where, on the x-axis, the summed Weighted Personalized PageRank scores per party are given, and on the y-axis, the number of seats earned by parties in the next elections. Per election, a different colour is given to the data points. Not all colours have the same number of data points because of the different parties that joined the elections and won a seat over the years. The plot is shown in figure 13.

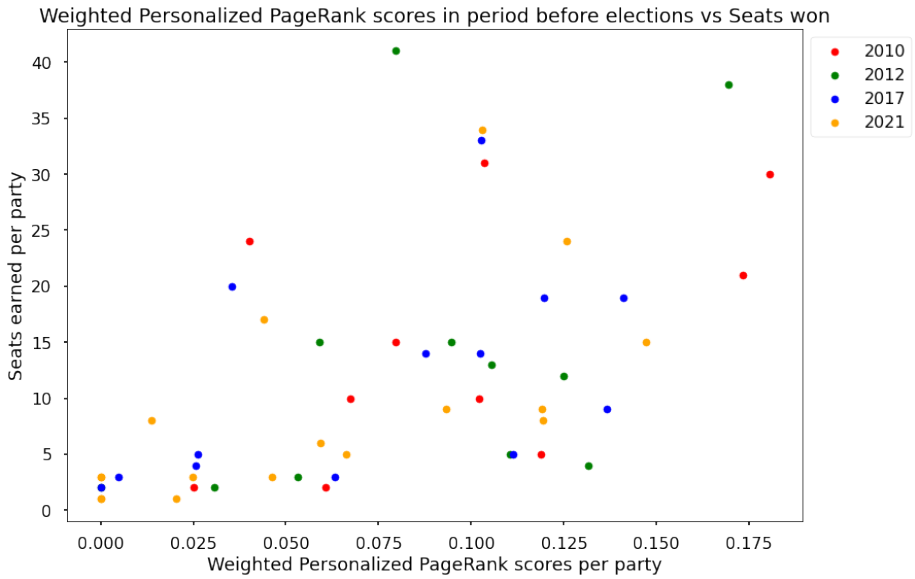


Figure 13: Weighted Personalized PageRank scores vs Seats earned per party

From figure 13 it can be seen that there is at least no strong correlation between the Weighted Personalized PageRank scores and the seats earned per party. If the different elections are disregarded, and the data points are all seen as the same type, a different plot can be created with a linear trend-line, as is displayed in figure 14. It is evident that the linear trend-line does not achieve a good fit here, but it also becomes apparent that there seems to be no correlation. Now, this is not necessarily a problem, as if there exists a correlation between the separate elections and the PageRank scores, the information is still valuable. The elections do not have to be generalized altogether.

Much better results are obtained if the elections are taken separately, for example, for the 2021 elections, where with Standard Random Walk PageRank, there seems to be a strong linear correlation. The Pearson Correlation Coeffi-

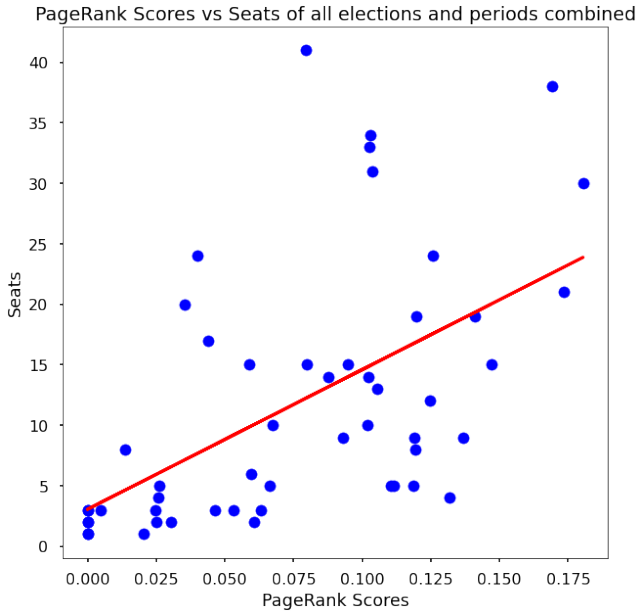
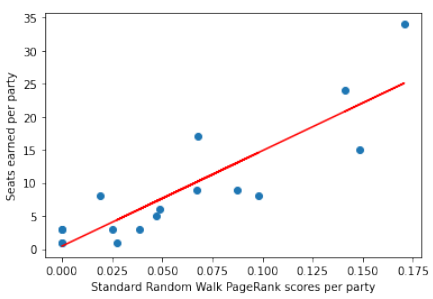


Figure 14: Weighted Personalized PageRank scores

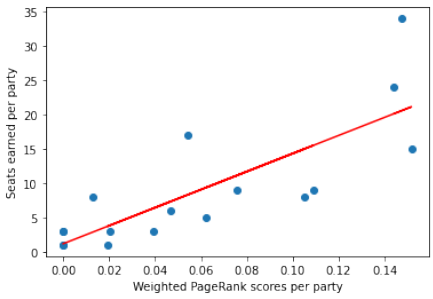
cient gets as high as 0.874. The Standard Random Walk trend-line is shown in figure 15, along with the plots for the different algorithms for the 2021 elections. The Standard Random Walk PageRank has a strong correlation, but this is immediately the highest coefficient achieved over all elections for all the PageRank algorithms. The results per algorithm per election are given in the table below.

Pearson's Correlation Coefficient for PageRank Scores comparisons with Seats earned per party				
Election year	Standard Random Walk	Weighted	Personalized	Weighted Personalized
2010	0.683	0.551	0.709	0.510
2012	0.749	0.642	0.677	0.459
2017	0.650	0.614	0.625	0.517
2021	0.874	0.798	0.772	0.636
Averages	0.739	0.651	0.696	0.531

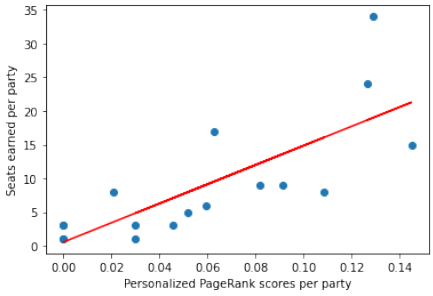
The Standard Random Walk PageRank algorithm performs best for all elections except the 2010 elections. It also performs best on average over all elections, but if an absolute value of 0.8 or higher is taken as the requirement for a strong correlation, only for the 2021 elections does there seem to be a strong correlation. This means that hypothesis 2 is also false, as it would have to be a strong correlation for every election.



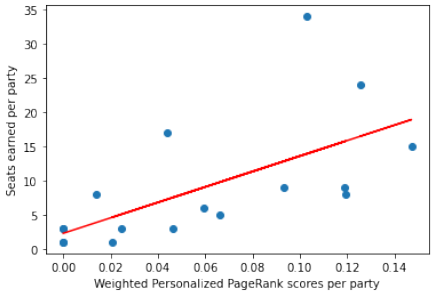
(a) Standard Random Walk



(b) Weighted



(c) Personalized



(d) Weighted Personalized

Figure 15: PageRank Scores per party for the period 2017-2021 vs number of seats earned at the 2021 elections

To illustrate that the correlation is not strong in a different manner, the seat predictions are calculated as explained in section 4.3.2 with Weighted Personalized PageRank for the 2017 elections. The results are shown in figure 16. The predictions for some parties seem to be correct or close, but for others, they are far off. Because the Weighted Personalized PageRank algorithm seems to favour parties that have filed many motions, the PVV and VVD are, for example, predicted to get fewer seats than they earned. Furthermore, PvdA, SGP, and ChristenUnie are predicted to get way more seats. If the whole period of 2012-2017 is taken, the PVV filed 696 motions, which in comparison to D66 (1532), PvdA (1503), VVD (1229), and ChristenUnie (1072) is not that much. From the plot and the number of motions filed, one can see that there is a correlation between the motions filed and the predicted seat distribution, which makes sense, as it was shown in hypothesis 1 that there is a correlation between the number of motions filed per politician and their corresponding (Weighted Personal) PageRank score. As for the difference between taking the whole period before the elections (2012-2017), or just the year before the elections, there is not that big a difference, which is interesting to see. It means that the PageRank distribution for only the last year is almost the same as for the whole period. The Mean Squared Error (MSE) for the prediction of the whole period equals 64.92, and for one year, the Mean Squared Error is 66.31, thus slightly favouring the whole period as a predictor. As for the other PageRank algorithms, the following MSE values are obtained for the 2017 elections:

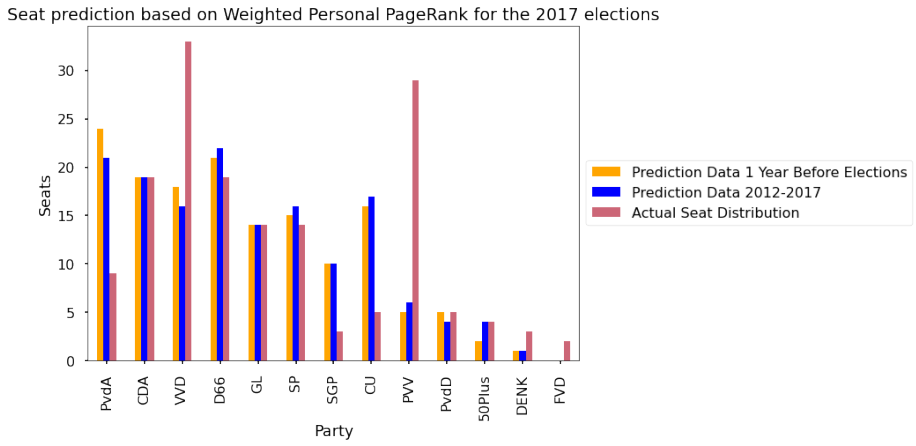


Figure 16: Predictions for the 2017 election based on Weighted Personal PageRank

MSE for predictions of various PageRank algorithms for the 2017 elections				
Data Used	Standard Random Walk	Personalized	Weighted	Weighted Personalized
1 year	74.92	69.69	62.46	66.31
whole period	59.38	54.31	58.92	64.92

Although Personalized PageRank for the whole period still has a high Mean Squared Error of 54.31, it is the best option for this election. The fact that Weighted Personalized PageRank is not the best algorithm, in this case, whilst it is the most complex, most likely has to do with the fact that it favours parties with many motions filed. It is interesting to see that data from the whole period instead of only the last year is a better prediction, as one may guess that the election outcome heavily depends on what happened in the last year.

5.3 H3

The first plot to demonstrate the difference between the ministers and the regular coalition members is a box plot over all periods. This plot is figure 17. It can be noticed that ministers seem to have a higher PageRank score on average than the non-ministers. That is a good sign, as it could potentially mean that it is possible to make a split on a specific PageRank score to classify politicians that might become ministers. One thing that needs to be kept in mind is that there are fewer ministers in the coalition than non-ministers. If the data is presented in a different format, it becomes clear that it is hard to draw a line on the PageRank scores to determine the difference between ministers and non-ministers. Have a look at figure 18. This is data about the period up until 2010. The data points for ministers are in blue and purposefully made bigger so that the data points of non-ministers are still visible even if they overlap. Notice how many orange data points are to the left of the first minister, but there are still way too many non-ministers grouped with the ministers to do a good split. Unfortunately, this was already one of the nicer splits from the data. A worse example is the data for the period 2012 until 2017. In this plot, figure 19, one might as well flip a coin to determine whether someone became a minister for the first time based on their PageRank scores in the previous period.

If all periods are combined into one plot for the Weighted Personalized PageRank Scores, figure 20 is obtained. There seems to be no difference between the distribution of the ministers and the non-ministers. In this case, one must remember that the PageRank scores depend on how many people are in the network, and if more politicians are in the network, the scores need to be distributed among more people. Because of that, PageRank scores plotted over all periods like in figure 20 might not be representative, as it becomes harder to get a higher score if more politicians are involved. Just like the results for hypothesis 2, it would be no problem if the generalized data could not be split. However, based on the plots about particular periods, no solid splits can be made to decide whether one has an increased chance of becoming a minister in the next period. Of course, one can find the optimal split, but as mentioned in the objectives, the time and effort put into this hypothesis is significantly less than for the other two hypotheses. No further tests were conducted to get precise numbers on the achieved plots and distributions. Nevertheless, from these plots, it can already be concluded that hypothesis 3 is false. There is no strong correlation between the PageRank score of a politician in a prior period and the chances of getting a role as a minister in a new cabinet.

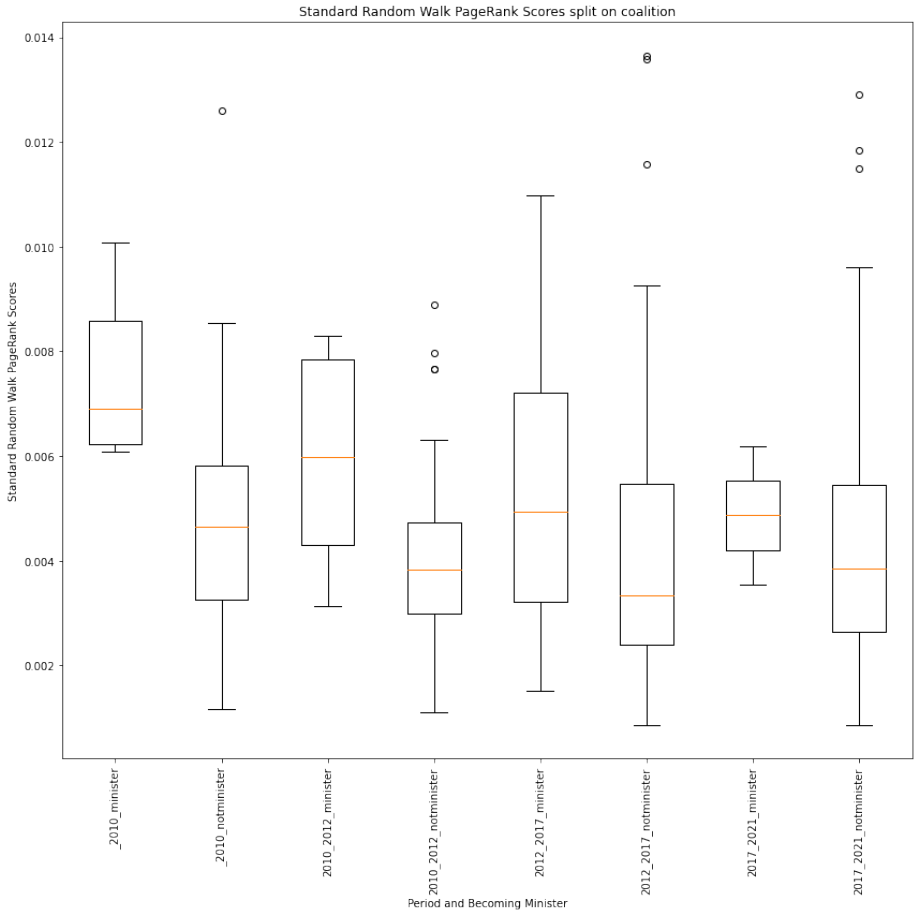


Figure 17: Standard Walk PageRank Scores of ministers and non-ministers in the previous period

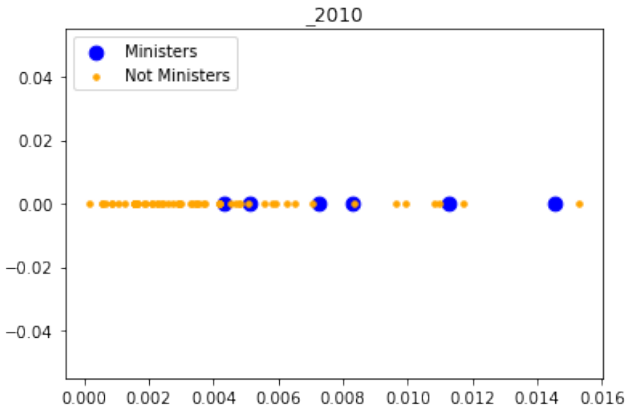


Figure 18: Weighted Personalized PageRank Scores of ministers and non-ministers in the previous period for 2010

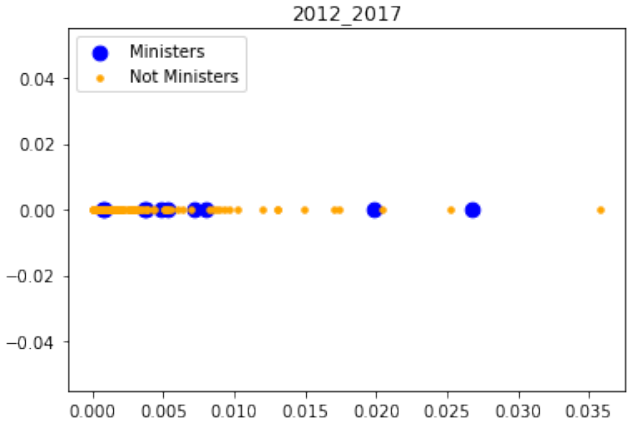


Figure 19: Weighted Personalized PageRank Scores of ministers and non-ministers in the previous period for 2012-2017

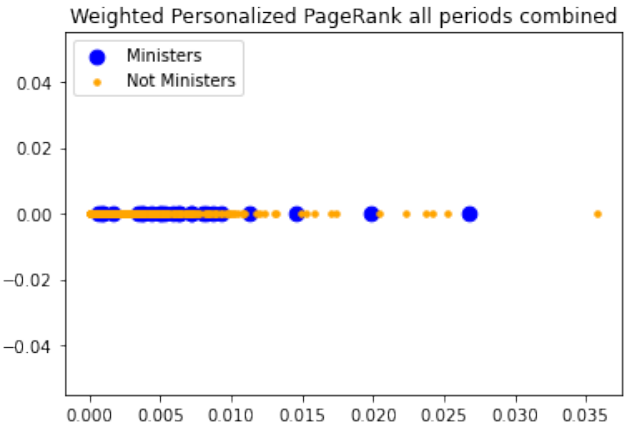


Figure 20: Weighted Personalized PageRank Scores of ministers and non-ministers from all periods combined

6 Discussion and Future Work

6.1 General

As mentioned before, it is debatable when a Pearson's Correlation Coefficient represents a strong relationship. The chosen value of 0.8 might be too high or even too low regarding the used data. To come up with the value of 0.8, the plots were taken into account. In figure 15, there seems to be a correlation for the Standard Random Walk algorithm. In figure 9, a (linear) correlation seems far-fetched for the Weighted Personalized PageRank algorithm. The coefficients for those figures were 0.874 and 0.671, respectively. The decision was made that because the correlation for Weighted Personalized PageRank in figure 9 seemed weak, the score of 0.671 was too low to take as a threshold. For the Standard Random Walk, though, it seemed there is a linear correlation, so the threshold had to be under 0.874. Because other authors picked 0.8 as a threshold, this was done in this report.

Also, in general, all hypotheses could have been tested for the data where only the accepted motions are considered. It is mentioned in the description of the data that information is present about whether a motion was accepted, but testing has only been done on all motions. The reasoning was that now a more extensive network is used, but that does not have to mean that it yields better results than a smaller network. Time was spent on other parts of the research instead of running the same analysis for only accepted motions.

Another bit that could be implemented for every hypothesis is the use of more types of PageRank algorithms. The four types used in this paper are not the only four PageRank algorithms. Algorithms such as Strongly Preferential PageRank

or Dirichlet PageRank [22] can be programmed to fit on this network. Maybe those algorithms will create stronger linear relationships between the data.

6.2 H1

Something that does not help the research is that the PageRank score is dependent on how many politicians were in that period. It does not immediately become a problem if only the period before the period currently being evaluated is taken as then the proportions of the PageRank scores apply to all politicians. However, creating a general (linear) formula that fits all periods is impossible. What would help to get a more general formula would be to split the data into years instead of periods. This way, the data should also achieve a fairer split in size. It does not mean that the same number of motions are filed yearly; nevertheless, it will probably get much closer than the differences between the current periods. If data is prepared per year, it might be possible to analyse whether the number of motions filed the year before strongly influence the PageRank scores in the next year. A time series could be created by taking the average number of motions filed per politician and the average PageRank score over that year, which can be calculated for every PageRank algorithm. This will certainly help explain the network on a deeper level.

6.3 H2

There are various areas in which the current research lacks, thus multiple chances for improvement. Regarding the correlation between PageRank and seats earned, it might be better to compare based on individual votes in the future. This data does exist [23], but it will take a while to mine and clean the data, which is why it was not yet done for this report. This might also give a weird distribution, though, as party leaders are likely to get more votes simply because they are on top of the voting bill, but it is not necessarily the case that the party leaders have the highest PageRank scores. It might be interesting to use the number of social media followers per politician, where those statistics can even be used for the Personalised PageRank algorithm instead of the number of motions filed per politician. The reasoning is that many people will get an impression of politicians through social media instead of the motions they file.

As far as the prediction goes for actual seats earned per party, the model performs way worse than the opinion polling made just before the 2017 elections [24]. That cannot be called a surprise, as the opinion polling predictions are based on more data types and thus form a more accurate prediction. In the future, PageRank might have a place in the algorithm used at hand to come up with the specific amount of seats predicted, but for now, the algorithm for the opinion polling as it is performs quite well already.

6.4 H3

It might also be possible to use the individual votes for this hypothesis. Either from the previous elections or the current elections, after which the coalition will form, and the minister roles will be distributed. Maybe a better split can be done with this data instead of the split solely based on the PageRank scores in the previous period. Perhaps a combination of the two could even work. Social media activity could also be monitored to see whether that influences the chance of becoming a minister in an upcoming coalition.

7 Conclusion

In conclusion, it can be said that although there is some positive linear correlation between the PageRank scores of politicians in the current period and the scores of the previous period, there exists no solid linear positive correlation. Weighted Personalized PageRank performed best but was still not good enough to be a solid predictor. No PageRank algorithm used in this report gets a constant strong linear positive correlation between the sum of the PageRank scores per politician per party and the number of seats won per party in the upcoming elections. Here, Standard Random Walk PageRank performed best, but it was still not good enough. Furthermore, the PageRank scores are also not helpful in determining which politicians have a shot of becoming a minister in the next coalition. Research with PageRank on this data set still yielded some interesting results. Maybe with some extra data like votes earned per individual instead of per party, better results could be achieved in the future.

8 References

[1] Google. (n.d.). How we started and where we are Today. Google. Retrieved June 13, 2022, from https://about.google/our-story/?utm_source=about.google&utm_medium=social&utm_campaign=copy-link .

[2] Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web. Stanford InfoLab.

[3] Sunil, A. N. V., & Sardana, A. (2012, March). A pagerank based detection technique for phishing web sites. In 2012 IEEE Symposium on Computers & Informatics (ISCI) (pp. 58-63). IEEE.

[4] Brown, S. (2017). A PageRank model for player performance assessment in basketball, soccer and hockey. arXiv preprint arXiv:1704.00583.

[5] Wasserman, S., & Faust, K. (1994). Social network analysis: Methods and applications.

[6] Heaney, M. T., & McClurg, S. D. (2009). Social networks and American politics: Introduction to the special issue. American Politics Research, 37(5), 727-741.

[7] Open Kamer. (n.d.). Stemmingen - Stemmingsuitslagen van wetsvoorstellen, moties en amendementen - Tweede Kamer. Retrieved May 10, 2022, from https://www.openkamer.org/stemmingen/ .

[8] Making web data extraction easy and accessible for everyone. Web Scraper. (n.d.). Retrieved May 10, 2022, from https://webscraper.io/ .

[9] N. (nilüfer) gündoğan. Parlement.com. (n.d.). Retrieved May 19, 2022, from https://www.parlement.com/id/vlgntc8p1yao/n_nilufer_gundogan .

[10] Alle Kamerleden. Tweede Kamer der Staten-Generaal. (n.d.). Retrieved May 19, 2022, from https://www.tweedekamer.nl/kamerleden_en_commissies/alle_kamerleden .

[11] Centraal Bureau voor de Statistiek. (2007, January 3). Verkiezingen; Historische Uitslagen Tweede Kamer. Centraal Bureau voor de Statistiek. Retrieved May 19, 2022, from https://www.cbs.nl/nl-nl/cijfers/detail/37278 .

[12] Zetelverdeling Tweede Kamer. Parlement.com. (n.d.). Retrieved May 19, 2022, from https://www.parlement.com/id/vh8lnhronvx6/zetelverdeling_tweede_kamer .

[13] Ministerie van Algemene Zaken. (2022, January 7). Kabinetten Sinds 1945. Regering — Rijksoverheid.nl. Retrieved May 19, 2022, from https://www.rijksoverheid.nl/regering/over-de-regering/kabinetten-sinds-1945 .

[14] Yan, E., & Ding, Y. (2011, July). The effects of dangling nodes on citation networks. In Proceedings of the 13th international conference on scientometrics and informetrics (pp. 4-8).

[15] Boldi, P., Santini, M., & Vigna, S. (2005, May). PageRank as a function of the damping factor. In Proceedings of the 14th international conference on World Wide Web (pp. 557-566).

[16] Mihalcea, R., & Tarau, P. (2004, July). Textrank: Bringing order into text. In Proceedings of the 2004 conference on empirical methods in natural language processing (pp. 404-411).

[17] Kiesraad. (2020, August 12). Kan een kandidaat die is gekozen voor een Bepaalde Partij Zich Na toelating tot het vertegenwoordigend orgaan afsplitsen? Kiesraad.nl. Retrieved June 12, 2022, from https://www.kiesraad.nl/verkiezingen/vraag-en-antwoord/kan-een-kandidaat-die-is-gekozen-voor-een-bepaalde-partij-zich-na-toelating-tot-het-vertegenwoordigend-orgaan-afsplitsen .

[18] ChristenUnie. (n.d.). De geschiedenis van een beginselpartij. Geschiedenis - ChristenUnie.nl. Retrieved June 13, 2022, from https://www.christenunie.nl/page/85 .

[19] Benesty, J., Chen, J., Huang, Y., & Cohen, I. (2009). Pearson correlation coefficient. In Noise reduction in speech processing (pp. 1-4). Springer, Berlin, Heidelberg.

[20] Havlicek, L. L., & Peterson, N. L. (1976). Robustness of the Pearson correlation against violations of assumptions. Perceptual and Motor Skills, 43(3_suppl), 1319-1334.

[21] Akoglu, H. (2018). User's guide to correlation coefficients. Turkish journal of emergency medicine, 18(3), 91-93.

[22] Gleich, D. F. (2015). PageRank beyond the Web. siam REVIEW, 57(3), 321-363.

[23] Kiesraad. (2021, May 10). Uitslag Tweede kamerverkiezing 17 maart 2021 Proces-Verbaal. Proces-verbaal — Kiesraad.nl. Retrieved June 22, 2022, from https://www.kiesraad.nl/adviezen-en-publicaties/proces-verbalen/2021/03/26/uitslag-tweede-kamerverkiezing-17-maart-2021 .

[24] Louwerse, T. (n.d.). Peilingwijzer, slotupdate maart 2017. Peilingwijzer.

Retrieved June 22, 2022, from https://peilingwijzer.tomlouwerse.nl/2017/03/ .

9 Python Libraries Used

- pandas
- math
- re
- numpy
- networkx
- matplotlib
- datetime
- ast
- scipy