POLITECNICO DI TORINO

Master's Degree in Mechatronic Engineering (Ingegneria Meccatronica)

Master's Degree Thesis

Analysis of political influence from Italian Senate voting data

Advisor:

Prof. Giuseppe C. Calafiore

Co-advisors:

Dr. Fabrizio Dabbene

Dr. Chiara Ravazzi

Candidate:

Antonio Longo

Academic Year 2017/2018

Acknowledgements

First and foremost I would like to personally thank my Advisor, Prof. Giuseppe Calafiore, and my Co-Advisors, Drs. Fabrizio Dabbene and Chiara Ravazzi. They have been an inestimable source of guide, notions and inspiration and this thesis would have not be possible without their help. I would also like to thank the Institute of Electronics, Computer and Telecommunication Engineering (IEIIT) of the National Research Council. This thesis has been developed in conjunction with the IEIIT and I would like to thank the Turin headquarter for warmly welcoming me.

The authors would also like to thank Vincenzo Smaldore and the OpenPolis Foundation for providing data and insights on topics related to this work.

To my family and my friends.

Contents

List of Figures

List of Tables

Introduction

The twenty-first century has been one of the most revolutionary periods in the context of social interactions. The exponential rise and evolution of the available technology, in particular the explosion of the World Wide Web and smartphones, has allowed the birth and the popularization of several social media platforms. In parallel to this kind of evolution there has been an unquestionable trend in the field of computer science: the ever-increasing interest and development in the subjects of machine learning, artificial intelligence and in general of data science. The two trends are intimately connected by the availability of *big data* providing researches with the needed tools to applies the techniques born in the field of computer science to the source of data coming from the aforementioned social dynamics.

Among the possible research interests emerging from this intersection, social influence analysis is becoming a cardinal one [10]. This is the result of a growing interest for the context of social networks. These networks represent an intricate web of social interactions between a collection of individuals, which in the academic framework are referred to as *agents*. The relations between these agents are the key element describing ultimately how the actions of an individual may influence the actions of the other agents. The key challenges in social network analysis can be summarized into three broad categories [11]: *modeling*, *analysis* and *control*. Modeling has the aim of finding a coherent mathematical description of the interactions underlying the social network. Analysis, starting from this mathematical model, serves the purpose of describing the dynamic evolution governing the interactions among the agents. Finally, control wishes to identify which are the most influencing agents in the network and to insert external agents able to direct its evolution.

The building of a mathematical model capable of explaining the interactions between the agents of a network has received a considerable amount of interest in the recent years from the control community. For example, the model developed by Friedkin and Johnsen [12] is able to explain the dynamic evolution of an opinion discussed inside groups of small and medium size. Other models deal with the emergence of a shared agreement , denominated *consensus* [13] or the disagreement

in networks where agents are present who never change their initial opinion [14]. One category of problems in which the social influence analysis framework reveals to be essential is in the description of *trust* among agents [15]. Networks are able to expose the degree of familiarity expressed between the agents and consequently explain the collective action undertook by the agents to reach an agreement.

To build and validate the model explaining the network it is of vital importance to correctly infer the influence that the agents exert on each other while shaping the dynamics of the social relationships. This is of particular interest when the influence is coming from a specific ideology to which the agent may be attached [16]. This kind of collective behavior is observable in a variety of different contexts, not only the one of social media, and one particular case of interest is the one of agents participating in politics [17]. The role of social networks in the context of political behavior has been subject of study for more than half a century, particularly in the United States of America [18, 19]. Recently the European Parliament has been interested by a similar analysis [34]. However, the Italian Parliament has never been subjected to this kind of studies.

From these considerations we have drawn the prime *motivation* of this work of thesis: we wish to apply this field of research to the Italian Senate. The Italian political scenario represents a very interesting case of study for its intrinsic complexity, compared to the foreign deliberative institutions. The variable political ideology characterizing the political members of the Italian Parliament provides an interesting and stimulating research challenge. To fulfill this challenge, we followed a machine learning approach.

The general task of machine learning is to derive from a set of sampled data the mathematical model used to describe data and eventually make predictions and decisions [20]. It is a broad field encompassing several disciplines [21] with statistics being the fundamental one and is mainly categorized in two settings: *supervised* and *unsupervised* learning [22]. In the supervised learning setting, the algorithm responsible to generate the model infers it from a labeled set of training data consisting of pairs composed by an input object and a corresponding output. The generated model can then be used to classify a new set of unknown data. On the other hand, in the unsupervised learning setting, the algorithm has to work with a set of data that has not been classified. In this sense, it has to learn the similarity embedded in the data, a task mainly linked to the extraction and explanation of *data features*, with a feature being a measurable propriety characterizing the data being observed.

We mainly drew the mathematical tools adopted in this thesis from this subfield of machine learning and in general from the broader pool of parametric statistical models. In particular, a

seminal work in the context of American political analysis [27] has led to the development of a parametric model denominated NOMINATE, which is a family of statistical techniques especially crafted for the analysis of political data. We combined this model with the more general algorithm of Principal Component Analysis (PCA) [31] to extract the features of our data. This process is called *dimensionality reduction* and it allows the model to better explain the data. Furthermore, to add a layer of physical interpretability to this process we also implemented the sparse variation of this algorithm knows as Sparse PCA [35, 37]. This algorithm directly extracts only the most relevant features corresponding to the most influencing voting sessions we collected.

A second category of techniques has been adopted to identify the *outliers* in the data. In the context of our research, the outliers are the legislators exposing the most interesting degree of influence in the network and can be used to empirically evaluate how well the model is capable of explaining the data. Since our data is naturally organized in groups (or *clusters*), corresponding to the political groups formed in the Senate, we decided to adopt several clustering techniques. The first is the k-means algorithm, which is probably the most famous unsupervised learning technique used to separate data into a predetermined number of clusters [48]. From the field of convex optimization, we adapted the minimum volume ellipsoid (MVE) [55] covering a finite set of points as a supervised learning technique to easily spot the outlying Senators. The last clustering technique has risen during the research process and involves the discovery of outliers by applying the Sparse PCA algorithm.

This toolbox of mathematical and statistical instruments allowed us to achieve the main result and correspondingly the main *contribution* to the research field of social influence analysis. This contribution is the introduction of a new influence measure based on the extracted data features that we denominated the Political Data aNalytic Affinity (Political DNA) of a Senator. This measure is based on an information-theoretic ground, by modeling the votes as outcomes of a *mixture* of random Gaussian variables and by reformulating the computation of Political DNA as an estimation of class posterior probabilities. This measure is interpretable as an index of similarity exposing the degree of *rebellion* or conversely of *discipline* to the ideology of the group. By examining the results obtained applying the model on the data of the XVII Legislature and the ones obtained by testing the model also on the present Legislature, they reflect the political structure underlying the Italian Senate and so we can reasonably affirm that we fulfilled our declared goal.

Outline

The work has been organized in the following way:

- In Chapter 1 we briefly explain how the Italian Parliament operates and how the Italian Senate is organized (1.1). We also explain how we mined the data from the public OpenParlamento (1.4) platform, in which way we derived the list of Senators and their nominal affiliation (1.4.2) and how we preprocessed it in preparation for our analysis (1.4.3).

- In Chapter 2 we describe the techniques we adopted to reduce the dimensionality of our dataset. More specifically, we expose the NOMINATE algorithm (2.1.2), which is the present state of the art tool in political science analysis and PCA (2.2.2) and Sparse PCA (2.2.3), two of the most representative techniques of unsupervised learning. In particular the sparse variation is very useful to also add a layer of interpretability to our results. The data has been reduced to 2 or 3 dimensions allowing us to represent it graphically.

- In Chapter 3 we expose the group of techniques that we adopted to identify the outliers, i.e. those Senators who expose a political affinity considerably different from the nominal one. These techniques too belong to the field of machine learning. They are the k-means algorithm (3.1), Minimum Volume Ellipsoids (3.2), adapted as a supervised learning technique, and Sparse PCA (3.3). For the Senators identified at the end of this analysis we show their individual Political DNA, the computation of which is described in the fourth Chapter.

- In Chapter 4 we introduce the stochastic model implemented to infer the influence measure for the Senators, which is a Gaussian Mixture Model (GMM) (4.1). By applying the GMM on the dimensionality-reduced dataset, we can derive a probability vector π for each Senator. This probability vector is the measure we developed to measure the degree of influence exerted on the Senators that we denominated Political Data aNalytic Affinity (Political DNA). We then expressed this measure as a convex combination of the vertices of a regular polytope. In this way we can represent it graphically both on a plane, producing a *political map* of the Italian Senate (4.2.1), and on a line, exposing the Political DNA for the individual Senators identified as outliers (4.3). At the end of the Chapter we also evaluate the model with the data of the present XVIII Legislature (4.4).

- Finally, in Appendix A we expose the complete list of the Political DNA of all the Senators of the XVII Legislature and in Appendix B the Political DNA computed for each group as an average of the belonging Senators.

Adopted notation

In this work the following notation has been adopted. Column vectors are denoted with lower-case letters and matrices with upper-case letters. Given a matrix X, its transpose is denoted with $X^\top$. X_{ij} is the element corresponding to the i-th row and j-th column. The i-th row is indicated by $x^{(i)}$ associated to a column vector while the j-th column is indicated by x_j, i.e.

$$X = \begin{bmatrix} x^{(1)\top} \\ \vdots \\ x^{(m)\top} \end{bmatrix} = [x_1, \ldots, x_n]$$

Given a vector $z = [z_1, \ldots, z_n] \in \mathbb{R}^n$, $\|z\|_2$ represents the standard Euclidian norm

$$\|z\|_2 \doteq \sqrt{\sum_{i=1}^{n} z_i^2},$$

$\|z\|_1$ the ℓ_1-norm

$$\|z\|_1 \doteq \sum_{i=1}^{n} z_i,$$

and $\|z\|_0$ the ℓ_0-pseudonorm which is the number of non-zero elements of z. Given a matrix $X \in \mathbb{R}^{m,n}$, $\|X\|_F$ represents the Frobenius norm

$$\|X\|_F \doteq \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} X_{ij}^2} = \sqrt{\mathrm{tr}\,(X^\top X)},$$

where $\mathrm{tr}(\cdot)$ is the trace of a square matrix $X \in \mathbb{R}^{n,n}$, i.e. the sum of the elements on the main diagonal: $\mathrm{tr}(X) = \sum_{i=1}^{n} X_{ii}$. Given a vector $z \in \mathbb{R}^n$, with the symbol $\mathrm{supp}(z) = \{i \in \{1, \ldots, n\} : z_i \neq 0\}$ we denote the set of non-zero elements of z.

$\mathbb{R}$ denotes the set of real numbers and $\mathbb{N}$ denotes the set of natural numbers. $\mathbb{S}_+^n$ and $\mathbb{S}_{++}^n$ denote respectively the sets of semi-positive and positive definite symmetric $n \times n$ matrices. A set is denoted by an upper-case calligraphic letter, e.g. $\mathcal{X}$. Given a non-empty set $\mathcal{X}$, we denote the cardinality of the set as $|\mathcal{X}|$.

Chapter 1

The Italian Senate

In this Chapter we briefly illustrate the Italian Parliament, what is the role of the Senate and how it is internally composed. We also show how a bill can be proposed to and processed by the Italian political system. We then expose the U.S. Congress study case, pointing out the differences with respect to the Italian Senate that have inspired this work of thesis. After that we explain how the OpenParlamento platform works and the data mining procedure we performed to build the dataset in a raw format. Finally in the last section we explain how we encoded and standardized the data in preparation for our analysis.

1.1 Parliament organization and basic functioning

The Parliament is one of the main constitutional institutions of the Italian political system and it is the one holding legislative power, i.e. the faculty to create new laws. It is structured with a *perfect bicameralism*: it is assembled by two chambers, the Chamber of Deputies and the Senate, both having equal powers and responsibilities. The Senate is composed by 321 members: 315 elected Senators, the former Presidents of the Republic and the Senators for life which can be nominated by the President of the Republic. The Senate is chaired by the President of the Senate, who is elected by the assembly and whose responsibility is to assure the proper operation of the Senate. Internally the Senators form political groups that organize the presence of political parties in the Senate. These political groups can form and disband during the whole Legislature and the Senators are free to migrate from the original political group to another.

In Table 1.1 it is shown the list of the political groups active at the end of the XVII Legislature, which lasted from the 15th of March 2013 to the 22nd of March 2018. In case of a political group

containing multiple parties, only the most indicative has been indicated in the name:

Political groups of the XVII Legislature		
Name	**Acronym**	**Political orientation**
Alleanza Liberalpopolare-Autonomie	ALA-PRI	Center
Nuovo Centrodestra	AP-CPE-NCD-NCI	Center-Right
Liberi e Uguali	Art.1-MDP-LeU	Left
Per le Autonomie	AUT(SVP-UV-PATT-UPT)-PSI	Center
Popolo della Libertà	FI-PdL	Center-Right
Grandi Autonomie e Libertà	GAL-UDCeDC	Center-Right
Lega	Lega	Center-Right
Movimento 5 Stelle	M5S	Independent
Gruppo Misto	Misto	Mixed
Noi con L'Italia	Ncl	Center
Partito Democratico	PD	Center-Left

Table 1.1: List of the political groups active at the end of the XVII Legislature

In our analysis we only consider the following political groups (for simplicity we only expose in the acronym the most indicative political party): PD (Partito Democratico), M5S (Movimento 5 Stelle), Lega, PdL (Popolo della Libertà), NCD (Nuovo Centrodestra) and LeU (Liberi e Uguali) while the remaining ones have been collected in the "Other" group.

Since both Chambers are politically symmetrical, a law can be proposed to either one of them. When proposed, the law undergoes an *iter* composed of four basic steps [6] (see Fig. 1.1):

1. The law is proposed to one of the two Chambers (Deputies or Senators) by the Government, the National Council for Economics and Labour (CNEL) or a group of citizens (at least 50,000). At this stage the law is commonly referred to as a legislative proposal (a bill).
2. The bill is discussed in the Chamber to which it has been proposed. While examining the bill, it is firstly analyzed by a Commission performing a preliminary assessment and then it is discussed in the Assembly.
3. When the bill has been approved by one of the Chamber, it is sent to the other one for the approval under the identical formulation. In the case that the other Chamber modifies the text of the bill, it is sent back to require the approval of the proposed modifications (this is the so called *navette*). This cycle is repeated until both Chambers agree on the formulation of the bill.
4. After the bill has been approved by both of the Chambers is submitted to the President of the Republic. The President can apply a veto on the bill requiring it to undergo again the same process of before. After the final approval, the bill is published on the Official Gazzette and the law becomes active 15 days after the publication.

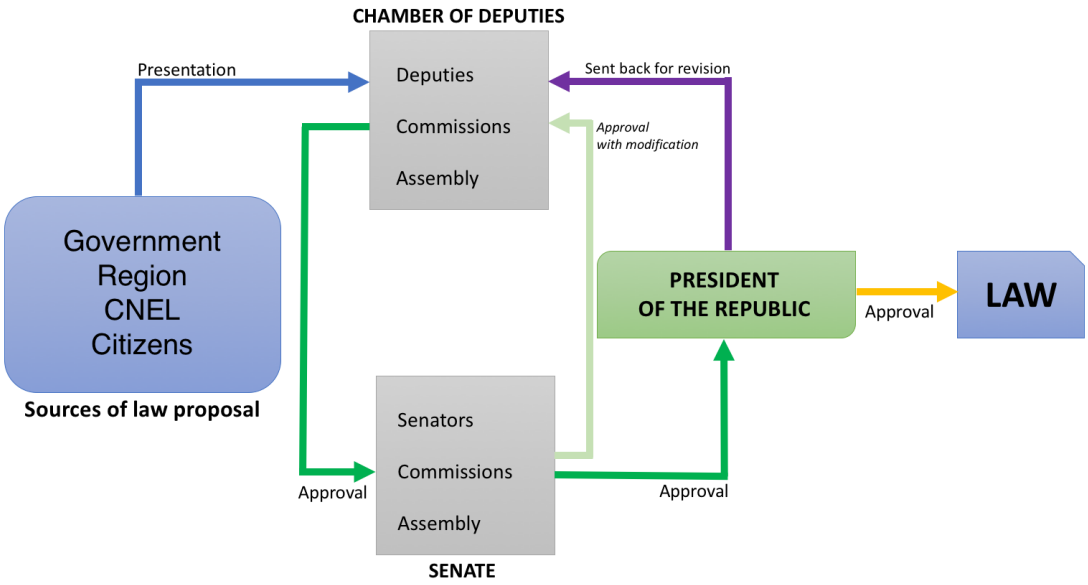


Figure 1.1: Simplified process for the presentation and approval of a law

1.2 The U.S. Congress study

The Congress of the United States of America has already a relevant history of research and study attached to it. A cardinal tool in the analysis of the U.S. Congress is the website GovTrack [1] launched by Joshua Tauberer in 2004. GovTrack enables its users to track the members and the bills in the Congress while providing comprehensive open data. Tauberer proposed to analyze the data by means of a mathematical model, in particular by assuming that a hidden Markov chain underlies the political dynamics in the Congress [23].

Mainly the research activity has focused on searching for spatial and network models to describe this kind of data. Poole developed the NOMINATE model [27] which has become the central reference to generate a spatial model of the U.S. political data and visualize them as a political map. An example is shown in Fig. 1.2: each point on the map represents a Senator in the U.S. Congress. It is immediate to notice that the Democrats (in blue) and the Republicans (in red) are well separated. The axes have not a physical meaning and are subject to the interpretation given by the viewer. In this case the horizontal axis can be a measure of the political orientation of the Senators (Democrats are located on the left while Republicans on the right). The vertical axis in this particular case has been interpreted as the closeness of a Senator to region or social issues.

Other kind of probabilistic models have been proposed, for example Clinton *et al.* [24] elaborated on the model proposed by Poole developing a Bayesian statistical framework to interpret

the U.S. political data. There has been also great deal of interest in the identification of an underlying network of influence among the members of the Congress. Scaglione *et al.* [25] applied the DeGroot model with stubborn nodes (i.e. agents who never change their initial opinion) to study the evolution of opinion dynamics during the voting sessions of the U.S. Congress.

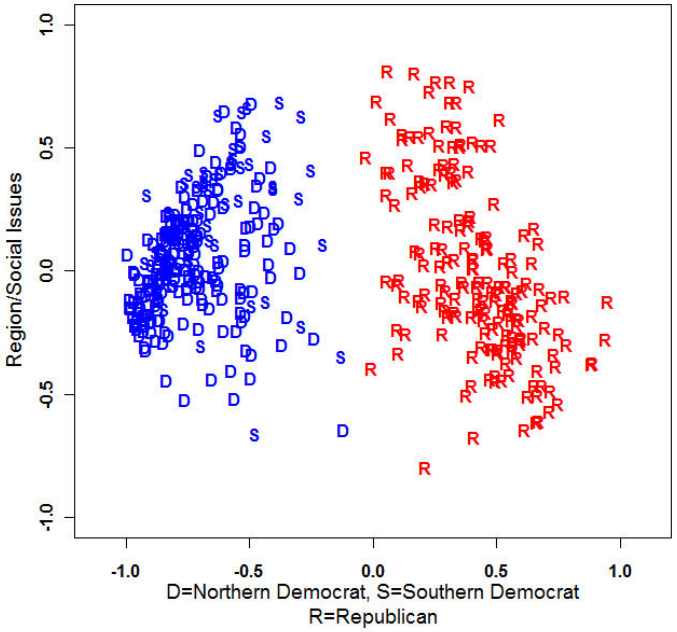


Figure 1.2: W-NOMINATE coordinates of members of the 111th House of Representatives [9]

Since this kind of analysis has never been applied outside the scope of American politics, this thesis has drawn inspiration from this line of research to gain insight on the political composition of the Italian Senate based on the data collected by the OpenParlamento platform. The Italian Senate however presents some intrinsic difficulties and differences with respect to the Congress case, rendering the analysis more challenging and calling for some adaptions. In particular, the higher number of legislators and their high fragmentation among several political groups have required a sensible effort while exposing some interesting results.

1.3 Openpolis and OpenParlamento

Openpolis is a no-profit foundation launched in 2008 with the purpose to promote in an open and simplified way access to public data, especially political and economical data. One of the main projects of the platform is the OpenParlamento platform [2]. On this platform it is possibile to access the vote history for each Senator, their membership to a political group in the Senate in a simplified fashion with respect to the official website of the Italian Senate [3]. The platform

publishes also a series of custom data analytics (e.g. the amount of *rebel votes*, i.e. votes who were different with respect to the one casted by the political group to which the Senator belongs). The votes are organized between the Chamber of Deputies and the Senate, organized first by legislature and then by Government, including the key votes. Each session is exposed in a page containing:

(a) The number of the session

(b) The date of the session

(c) The name of the bill

(d) A description of the bill

(e) The final outcome of the session (if the bill has been approved or rejected)

(f) A summary exposing the total percentage of favourable, contrary and abstained votes

(g) The list of the political groups with their percentage of favourable, contrary and abstained votes

(h) The list of Senators with their political affiliation, their vote and their regional district

The collected data useful for our purposes are (a), (b), (c), (d) and (h) and the collection process is exposed in Section 1.4. (e), (f) and (g) data can be inferred from the others and thus have been ignored. The core of the dataset is represented by the votes casted during the XVII legislature. Since a voting session implies to vote also on the amendments, sub-amendments and single articles composing a particular bill, it has been decided to reduce the redundancy in the dataset by limiting our analysis on only the "final votes" which ultimately determined if the proposal has been accepted or rejected by the Senate.

The votes are divided in four segments:

1. Key votes: This subset of the voting data, extending across the three governments, represents the most important votes during the legislature both for topic relevance and political value. They are 160 in total.
2. Letta government: This is the first government of the legislature with Enrico Letta as prime minister (from the 28th of April 2013 to the 22nd of February 2014). The votes are 52 in total.
3. Renzi government: This is the second government of the legislature with Matteo Renzi as prime minister (from the 22nd of February 2014 to the 12th of December 2016). The votes are 249 in total.

4. Gentiloni government: This is the third and last government of the legislature with Paolo Gentiloni as prime minister (from the 12th of December 2016 to the 1st of June 2018). The votes are 74 in total.

Since the key votes span across all the three governments, for their importance they can be assumed to be a reasonable summary of the legislature and are the part of the voting data on which we decided to focus our analysis.

1.4 Mining the Italian Senate

1.4.1 Data extraction

To extract the data presented on the OpenParlamento platform a web crawler (or *spider*) has been developed in Python adopting the Scrapy framework [4]. A crawler is a software used to extract specific data from website in an automated fashion. The spider works essentially implementing the following cycle (see Fig. 1.3):

- Opens a connection to the OpenParlamento website
- Generates the list of links representing the pages containing the needed data (the key votes)
- Iterates through each one of the links, filtering the data to extract only the Senate data
- Exports the extracted data to a MongoDB collection
- Closes the connection

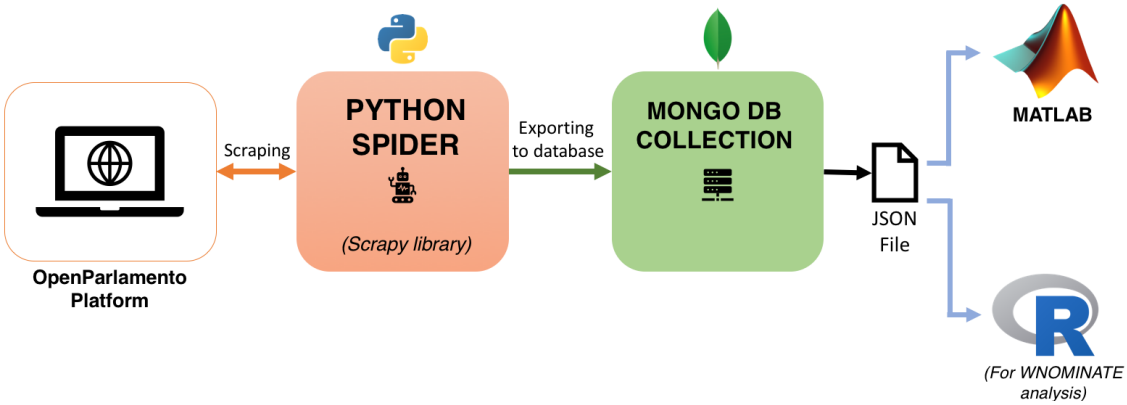


Figure 1.3: Chain representing the mining process of the Italian Senate

MongoDB [5] has been chosen because it is a document-oriented database and this kind of implementation allows for our case an easier data management with respect to a traditional relational

database. A *collection* is a grouping of documents in the MongoDB framework, equivalent to a table in a relational database. From the database the data has been exported as a JSON (JavaScript Object Notation) file to easily import it into MATLAB and R for our analysis purposes. Before proceeding with the analysis, the data is arranged in a matrix structure, with the bills present on the columns and the Senators on the rows. Each element Z_{ij} of this matrix represents the vote that the i-th Senator casted on the j-th bill and its construction is exposed in the following section.

1.4.2 Building the vote matrix

The first step to build the vote matrix is to obtain the list of all the Senators present during the XVII legislature from the dataset acquired at the end of the acquisition chain illustrated before. The operation is summarized in Algorithm 1.

Algorithm 1 Obtaining the list of all Senators present during the legislature

Set M_{tot} as the maximum number of Senators present over all the sessions
for each voting session j in $\{1, \dots, n\}$ **do**
  Obtain the list of all Senators $\mathcal{S}_j$ present at session j
  Obtain the total number of Senators M_j present at session j
  if $M_j < M_{\text{tot}}$ **then**
    Fill $\mathcal{S}_j$ with $(M_{\text{tot}} - M_j)$ zeros
  end if
end for
Obtain the total list of Senators $\mathcal{S}$ as $\cup_{j=1}^{n} \mathcal{S}_j$
Remove the 0 element from $\mathcal{S}$
Return the total list of Senators as $\mathcal{S} = \{s_1, \dots . s_m\}$
Return the total number of Senators as $|\mathcal{S}|$

The next step is to retrieve for each Senator also their most recent political affiliation in the Senate, i.e. to *label* the data. This operation is summarized in Algorithm 2.

Algorithm 2 Associating each Senator to their most recent political group

Set v as the most recent voting session
while there is a political group missing **do**
  for each Senator s_i in $\mathcal{S}$ **do**
    if the i-th group g_i is not missing **then**
      Skip the cycle
    else if s_i has voted in v **then**
      Assign g_i
    else
      Assign g_i as missing
    end if
  end for
  Set v has the immediately previous voting session
end while
Return the total list of political groups as $\mathcal{G} = \{g_1, \dots, g_m\}$

The set of labeled data is then obtained as $\mathcal{S}_L = \{(s_1, g_1), \ldots, (s_m, g_m)\}$. To easily obtain the vote expressed by a Senator during a voting session, each session has been organized as a *map data structure* (see Fig 1.4). A map stores information in the form of key-value pairs, without allowing duplicates. By accessing the map with a key we obtain the value corresponding to that key. In our case the key is the Senator's name and the value is their vote. In case that the Senator was not present during the session, a missing value is returned. It also compensates for the fact that the Senators may be listed in a different order for different sessions. This kind of data structure allows to quickly build the *vote matrix*. Each Senator is placed as a row on the matrix and each session (or bill) is placed as a column. By querying iteratively each map, at the end we obtain a filled matrix where each element Z_{ij} represents the vote expressed by the i-th Senator upon the j-th bill.

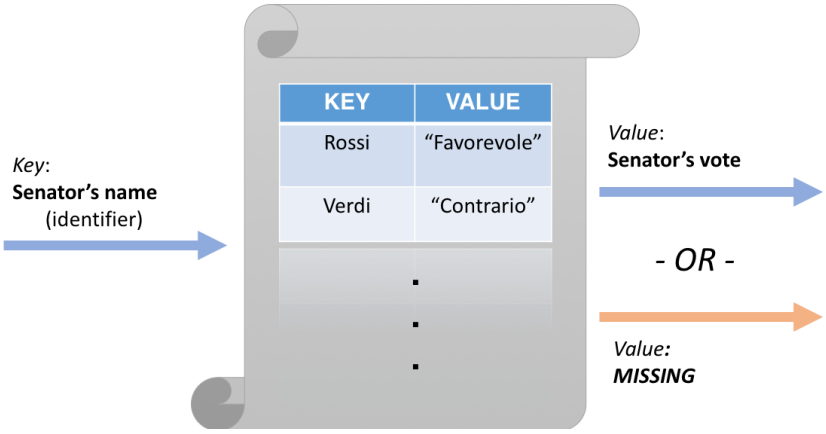


Figure 1.4: Associative map to obtain the preference expressed by a Senator

1.4.3 Vote matrix encoding

At this stage Z_{ij} denotes a verbal preference, which can be one of the following:

Unencoded elements of the vote matrix	
"Favorevole"	Approved the proposal
"Contrario"	Rejected the proposal
"In missione"	Not present for institutional reasons
"Assente"	Not present
"Astenuto"	Abstained from voting
"Voto segreto"	Secret ballot
"Presidente di turno"	President of Senate
"Richiedente la votazione e non votante"	Requested the voting and doesn't vote
N.A.	Not present in legislature

Table 1.2: Meaning of the verbal preferences present in the unencoded vote matrix.

The N.A. case is reserved for the Senators who started their mandate after the beginning of the

legislature or for those who left the Senate before the legislature ended (e.g. deceased Senators). The result is a matrix belonging to a $m \times n$ space, where m is the number of senators and n the number of bills. Before proceeding we need to encode the matrix to convert the verbal preferences into numerical values, obtaining the dataset that will be analyzed in the next sections. The first kind of encoding is the following one:

Standard encoding of the vote matrix	
"Favorevole"	1
"Contrario"	-1
"In missione"	0
"Assente"	0
"Astenuto"	-1
"Voto segreto"	0
"Presidente di turno"	0
"Richiedente la votazione e non votante"	0
N.A.	0

Table 1.3: Codification of the vote matrix adopted for PCA and Sparse PCA.

In this case $+1$ has been assigned for a favourable vote, a -1 for a contrary one and 0 in the other cases. Note that, following the Senate ruling active during the legislature, abstention has been considered equivalent to a rejection. A different kind of encoding has been adopted particularly for the NOMINATE procedure, as suggested by [29]:

NOMINATE-style encoding of the vote matrix	
"Favorevole"	1
"Contrario"	6
"In missione"	9
"Assente"	9
"Astenuto"	6
"Voto segreto"	9
"Presidente di turno"	9
"Richiedente la votazione e non votante"	9
N.A.	0

Table 1.4: Codification of the vote matrix adopted for the NOMINATE procedure.

In this case it is possible to discriminate between an abstention and a missing Senator in legislature. At the end of the encoding procedure we obtain a matrix $Z \in \{-1,0,1\}^{m,n}$ in the first case and $Z \in \{0,6,9\}^{m,n}$ in the second one. The final step is to clean and standardize the dataset. Before the cleaning procedure the total number of Senators is $m = 339$ and the total number of bills $n = 160$. The cleaning is performed by removing all the rows and columns equal to 0 from the vote matrix. Rows equal to 0 correspond to Senators who never voted during the legislature, while columns equal to 0 correspond to bills that have been voted in a secret ballot (i.e. they were

not observable). At the end of the cleaning procedure the total number of Senators is $m = 335$ and the total number of bills is $n = 155$. After that we perform the standardization procedure, i.e. we both remove the mean value from the columns and we normalize their norm. This is a usual preparatory step especially for PCA analysis [30, 32] since centering and scaling the data prevents the formation of spurious results. Formally,

$$X_{ij} = \frac{Z_{ij} - \frac{1}{m}\sum_{i=1}^{m} Z_{ij}}{\sqrt{\sum_{i=1}^{m}(Z_{ij} - \frac{1}{m}\sum_{i=1}^{m} Z_{ij})^2}}.$$

At the end of this process we obtain the standardized vote matrix $X \in \mathbb{R}^{m,n}$, which will be the fundamental data object undergoing our analysis. To expose the information embedded inside this matrix we need to reduce its dimensionality, limiting it to the least amount necessary to obtain enough insight on the data. This is achieved by projecting the dataset on a k-dimensional subspace (with $k < n$) with one of the techniques exposed in the next chapter.

Chapter 2

Dimensionality Reduction

The main goal of this Chapter is to introduce the techniques adopted to better describe our data, i.e. to expose the political informations embedded inside the vote matrix. This is achievable by reducing the dimensions of the data space, limiting them to the least amount necessary to gain enough insight on the data. Brigadir *et al.* illustrated a similar analysis applied on the votes of the European Parliament [34]. There are two approaches that may be followed: one is to use a so-called scaling method, in particular NOMINATE and its variant W-NOMINATE that are a family of multidimensional scaling techniques developed especially for roll-call data. These are illustrated in Section 2.1, denominated Multi-dimensional Scaling. The alternative is to adopt more general approaches derived from the domain of modern algebra, such as PCA and Sparse PCA, to obtain a reduced dataset by projecting the n-dimensional data on a k-dimensional subspace (with $k < n$). They are exposed in Section 2.2 that has been denominated Feature Projection.

2.1 Multi-dimensional Scaling

Multi-dimensional scaling (MDS) is a technique adopted to visualize the amount of similarity between the elements of a dataset by performing a dimensionality reduction. It operates by placing the each element of the dataset in a k-dimensional space while preserving the distance between the elements as much as possible. The number of dimensions k is specified a-priori and can be freely chosen. However, it has been shown [40] that $k = 2$ is the usual upper limit to optimize the data plot, in particular $k = 2$ optimizes the visualization on a scatterplot. Values of $k > 3$ are rarely used since they hinder visualization. In the next Sections we will focus on NOMINATE, which is a technique of MDS especially developed for political analysis.

2.1.1 The spatial model of roll-call data

A widely known and used method to visualize parliamentary roll call data is the use of spatial maps generated by a geometrical representation of the legislators and the votes. The idea is to represent each senator by one point and each law proposal by two points: one for "Favorevole" (Y) and one for "Contrario" (N). At the end, a spatial map is formed that summarizes and allows for the visualization of the information embedded in the vote matrix. This theory was first introduced by Downs [26]. A comprehensive summary on the theory of modeling legislative preferences through a spatial model has been produced by McCarty [41]. From this spatial maps a scaling method can be implemented allowing us to identify these geometrical points and predict the outcome of a voting session. This is possible since in the statistical approach to the spatial model, it is assumed that the legislator preference on one of the outcomes respects the following two basic proprieties:

1. Single-peakedness: For all the possible outcomes, the legislator cannot have two preferences ranking higher then all the other alternatives. In other words, a legislator can always choose a single preferred outcome identified by a point, which is the legislator's ideal point.

2. Symmetry: If two outcomes have equal distance from a legislator's ideal point, the legislator is indifferent between the two. The assumption that the legislator will always choose the outcome that is the closest to their ideal point is called *sincere voting*.

From these assumptions, each roll-call (i.e. each bill column x_i of the vote matrix) can be characterized by a cutting line dividing the ideal points of the legislators supporting the Y outcome from the ones supporting the N outcome. In the mono-dimensional case, the cutting line is simply a point, called a cutting point. An example is offered in Table 2.1, where each row represents a possible voting pattern during a voting session.

Vote	**Senator A**	**Senator B**	**Senator C**
1	Y	N	N
2	Y	Y	N
3	N	Y	Y
4	N	N	Y
5	Y	Y	Y
6	N	N	N

Table 2.1: Example of roll-call voting patterns

Each of these patterns can be explained by a simple spatial model as indicated before: each Senator can be assigned to an ideal point and a cutting point can be assigned to each vote to divide the Senators who voted Y from the ones who voted N. For example, a possible model using

the ordering C > B > A is illustrated in Fig. 2.1, explaining all six of the voting patterns.

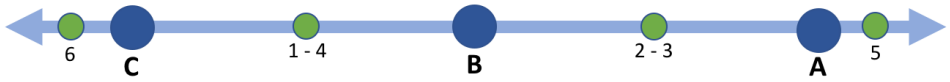


Figure 2.1: A possible 1-D spatial model for the vote patterns in Table 2.1

The Senators' ideal points are represented by their letters (in blue) while the votes cutting points are represented by their number (in green). For simplicity the points representing the Y and N outcome for each vote are not shown.

Other Senators' ordering (e.g. C < B < A) are capable of explaining the patterns as well. When two or more ordering are possible, they are formally equivalent. A choice can be made by the researcher exploiting a-priori information about the political orientation of the Senators (e.g. left-wing or right-wing legislators). On the other hand, any kind of ordering incapable of explaining all the votes is inconsistent with a one-dimensional spatial model for Table 2.1 and should be discarded. It is immediately clear that unanimous votes like 5 and 6 are explainable by any kind of Senators' ordering. As such they are statistically irrelevant and should be discarded as well. It is also worth noticing that this kind of data is unable to produce a preference scale: we are unable to know whether A is closer to B than B is to C.

The problem arises when some patterns can not be explained by a consistent spatial model. For example, a row like

Vote	**Senator A**	**Senator B**	**Senator C**
7	Y	N	Y

cannot be explained by the model in Fig. 2.1. When there a few cases like this, it is reasonable to assume that they may have been generated by a stochastic behavior (i.e. a noise source) casting the model into a probabilistic framework. In this kind of framework, a Senator i with an ideal point x_i is assumed to choose the outcome of an alternative z_j according to an utility function $U(x_i, z_j) + \varepsilon_{ij}$ where ε_{ij} is a stochastic error term. In this framework, the identification of the ideal points and the vote locations is sensitive to both the structure of the utility function U and the distribution of the error terms. The main advantage of the probabilistic framework however is the possibility to generate a cardinal ideal point measurement, allowing to deduce how much a Senator is closer to or farther from another Senator. This is possible because, by assuming a stochastic behavior of the error terms in advance, the closeness of two Senators is related to the improbability (or probability) that these random events have led them to vote in the same way.

In the mono-dimensional case, the utility functions for voting Y or N can be written as

$$\begin{aligned} U(x_i, y_j) + \varepsilon_j^y \\ U(x_i, n_j) + \varepsilon_j^n, \end{aligned} \tag{2.1}$$

where x_i is the ideal point of the i-th Senator, y_j the spatial location of the Y outcome for the j-th bill and n_j the spatial location of the N outcome. The quantities ε_j^y and ε_j^n are the random error terms for the Y and N utilities, respectively. Within this framework, the assumption is that each Senator will vote for the outcome that maximize their utility function. Furthermore, by specifying a functional form for the error terms, it is possible to derive the choice probabilities and so the likelihood function of the observed votes. This kind of analysis has been mainly developed by Poole and Rosenthal through the NOMINATE (Nominal Three-step Estimation) procedure and exposed in their seminal paper [27]. It is illustrated in Section 2.1.2.

2.1.2 NOMINATE

For the analysis of the NOMINATE procedure we will consider for simplicity the mono-dimensional case. Defining the two possible outcomes of a vote as y_j and n_j for Y and N respectively, we can compute the middle point

$$m_j = \frac{y_j + n_j}{2}, \tag{2.2}$$

in one dimension. This is what we previously called a cutting point. When the spatial voting is perfect (i.e. sincere), all the Senators to the left (or right, polarizing the point doesn't affect generality) of the cutting point will vote Y (or N) and all the Senators to the right will vote for the opposite.

The computation of the legislator ideal point is discussed in [28] and exposed through a four-step process:

1. Compute the agreement score matrix: the agreement score between two Senators is the proportion of times they voted the same across all the considered bills. It is a symmetric matrix.

2. Convert the agreement score matrix into a matrix of squared distance: this is done by subtracting the agreement scores computed in the previous step from 1 and then squaring them.

3. Double center the matrix of squared distances: the row and the column means are subtracted

from each element of the matrix, then the matrix mean is added and finally divided by -2.

4. Take the square root of each diagonal element of the resulting matrix and divide the corresponding column by this quantity.

The resulting values are the legislators ideal points. We can then define the two utility functions for the Y outcome and the N outcome on the j-th bill in a similar fashion as we did in (2.1):

$$\begin{aligned} U_{ijy} &= u(x_i, y_j) + \varepsilon_j^y = u_{ijy} + \varepsilon_j^y \\ U_{ijn} &= u(x_i, n_j) + \varepsilon_j^n = u_{ijn} + \varepsilon_j^n, \end{aligned} \tag{2.3}$$

u is the deterministic portion of the utility function while ε is the stochastic one. Both of them are assumed to be normal distributed. If $U_{ijy} > U_{ijn}$, the i-th Senator will vote Y on the j-th bill. That is, if the difference $U_{ijy} - U_{ijn}$ is greater then zero, i.e. if

$$u_{ijy} - u_{ijn} > \epsilon_{ijy} - \epsilon_{ijn}.$$

In other words the i-th Senator is assumed to vote Y on the j-th bill if the difference between the deterministic terms of the utility function is greater than the difference between the random terms. The same kind of reasoning can be done for the N outcome.

From here we can directly compute the probability that a senator will vote Y or N

$$\begin{aligned} \mathbb{P}(\text{Senator } i \text{ votes Y}) &= \mathbb{P}(\epsilon_{ijy} - \epsilon_{ijn} < u_{ijy} - u_{ijn}) \\ \mathbb{P}(\text{Senator } i \text{ votes N}) &= \mathbb{P}(\epsilon_{ijy} - \epsilon_{ijn} > u_{ijy} - u_{ijn}), \end{aligned}$$

where $\mathbb{P}(\text{Senator } i \text{ votes Y}) + \mathbb{P}(\text{Senator } i \text{ votes N}) = 1$.

Since the utility function of a Senator is a bell-shaped curve (having to respect both the assumptions of single-peakiness and symmetry), two main kind of functions my be used: normal and quadratic. The normal utility function is the most used [28] and will be the considered one.

The i-th Senator's utility function defined by Poole and Rosenthal for the Y outcome on the j-th bill is

$$u_{ijy} = \beta \exp\left(-\frac{1}{2} w d_{ijy}^2\right). \tag{2.4}$$

Similarly the function for the N outcome can be defined. In this way the difference between the

deterministic terms of the utility functions becomes

$$u_{ijy} - u_{ijn} = \beta \left\{ \exp\left(-\frac{1}{2} w d_{ijy}^2\right) - \exp\left(-\frac{1}{2} w d_{ijn}^2\right) \right\},$$

where d_{ijy}^2 and d_{ijn}^2 are the squared distances between the i-th Senator ideal point and the Y outcome and N outcome for the j-th bill, respectively. In the utility function expression we can recognize the salience weight w, used in W-NOMINATE (Weighted-NOMINATE) flavor. This coefficient allows for the weight of the bills and the Senators position to vary in each dimension the case of multi-dimensional estimation. β is a tune parameter used to compensate the noise level. Both of these terms are determined during the estimation procedure.

Various kind of distributions (uniform, normal, logit) have been proposed also for the probabilistic part of the utility function but, also in this case, the normal distribution has been chosen for its proven advantages [28]. This can be motivated by the fact that statistically, the random errors affecting the Senators' choices should be a set of independent and identically distributed random variables drawn from a known distribution. If ϵ_{ijy} and ϵ_{ijn} are random samples of the error distribution, they should respect this conditions from a behavioral point of view:

- ϵ_{ijn} and ϵ_{ijy} should be symmetric and unimodal
- ϵ_{ijn} and ϵ_{ijy} should be uncorrelated

The normal distribution satisfies all of these criteria. Also, if ϵ_{ijn} and ϵ_{ijy} are normal, their difference $\epsilon_{ijn} - \epsilon_{ijy}$ is normal too. Starting from (2.3) then, we can derive the overall utility distribution:

$$U_{ijy} - U_{ijn} \sim N(u_{ijy} - u_{ijn}, \sigma^2). \tag{2.5}$$

We can then rewrite the probability for the i-th Senator to vote Y on the j-th bill as

$$\begin{aligned} \mathbb{P}_{ijy} &= \mathbb{P}(U_{ijy} > U_{ijn}) = \mathbb{P}(\epsilon_{ijn} - \epsilon_{ijy} < u_{ijy} - u_{ijn}) \\ &= \Phi\left[u_{ijy} - u_{ijn}\right]. \end{aligned} \tag{2.6}$$

Adopting the normal utility function we can rewrite Φ as

$$\Phi\left[\beta \left\{ \exp\left(-\frac{1}{2} w d_{ijy}^2\right) - \exp\left(-\frac{1}{2} d_{ijn}^2\right) \right\}\right]. \tag{2.7}$$

The final step is the identification of the parameters characterizing the distribution (2.6). We have to identify:

- The Senator ideal point x_i
- The bill outcomes (both Y and N) y_j and n_j
- The utility function parameters β and w

The standard procedure is than to maximize the likelihood function L of actually observing the vote data:

$$L = \prod_{i=1}^{n} \prod_{j=1}^{m} \prod_{\tau=1}^{2} P_{ij\tau}^{C_{ij\tau}}, \tag{2.8}$$

where τ is the index for Y or N, $P_{ij\tau}$ is the probability of voting for the τ outcome and $C_{ij\tau}$ is equal to 1 if the Senator's choice is actually 1 and 0 otherwise. Usually it is standard practice to work with the log-likelihood function:

$$\log(L) = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{\tau=1}^{2} C_{ij\tau} \log\left(P_{ij\tau}\right). \tag{2.9}$$

As a further restriction, the outcome points are estimated in terms of the midpoint m_j between the two, as defined in (2.2), and the distance d_{ij} between each outcome and m_j.

Usually the maximization of the log-likelihood functions is done by taking the first derivatives with respect to all the parameters, setting them to zero and solving each one of the resulting equations. W-NOMINATE instead solves the problem in a more efficient way by adopting a three-steps algorithm:

1. Estimate m_j and d_{ij}, keeping x_i, β and w fixed.
2. Estimate x_i, keeping the other parameters fixed.
3. Estimate β and w, keeping the other parameters fixed.

This procedure is iterated until the current parameter set correlates at 99% or better with the previous set.

To evaluate the quality of the model different statistic measures are possible. The first and most immediate one is the percentage of correct classifications. This measure reflects the percentage of cases where the choice of the Senator identified by the maximum estimated likelihood corresponds to the actual vote

$$\text{Classification Success} = \frac{\sum(\text{Correctly predicted votes})}{\sum(\text{Votes})} \cdot 100.$$

This measure has a main drawback. Suppose that a bill has passed with the votes expressed as in the following table, with a majority vote of the 85%:

Senator	**Vote**
1	Y
2	Y
3	Y
4	Y
5	Y
6	N

Then, a naive model where all the Senators vote for the winning alternative (in this case Y) has a classification success equal to 85%. If the winning margin increases then the baseline classification rate will increase too even if the model is not explaining the additional data. For this reason an alternative measure has been proposed called Aggregated Proportional Reduction in Error (APRE). It is defined as

$$\text{APRE} = \frac{\sum (\text{Minority votes - classification errors})}{\sum (\text{Minority votes})}. \tag{2.10}$$

Intuitively, it classifies how much better the model performs with respect to a naive majority model. The APRE value can vary between 0 and 1. When the APRE is equal to 0, the model cannot explain anything. When the APRE is equal to 1, perfect classification has been achieved.

A second measure called Geometric Mean Probability (GMP) has also been proposed. This measure tries to compensate for the fact that both the classification success and APRE cannot discriminate between a classification error close to the cutting point and an error with a larger magnitude. It is defined as the anti-log of the average-likelihood

$$\text{GMP} = e^{L/mn}, \tag{2.11}$$

where L is the likelihood value, m is the number of Senators and n is the number of bills. GMP value lies between 0.5 and 1. The more the GMP is near 0.5 the more the model is behaving like a completely random binary classifier (a coin flip). The closer it is to 1 the better the model is fitting the data. Both APRE and GMP have been reported in our results.

2.2 Feature Projection

2.2.1 SVD

The singular-value decomposition (SVD) is a particular kind of matrix factorization achieved starting from its eigenvalues and eigenvectors. The principal component analysis technique illustrated in Section 2.2.2 is strongly related to this factorization.

Specifically, given any rectangular matrix $X \in \mathbb{R}^{m,n}$ it holds that

$$X = U\Sigma V^\top, \tag{2.12}$$

where

- U is a $m \times m$ orthogonal matrix,
- Σ is a $m \times n$ rectangular diagonal matrix with non-negative scalars on the diagonal, i.e.

$$\Sigma = \begin{bmatrix} \hat{\Sigma} & 0_{r\times(n-r)} \\ 0_{(m-r)\times r} & 0_{(m-r)\times(n-r)} \end{bmatrix},$$

  with $\hat{\Sigma} = \text{diag}(\sigma_1, \ldots, \sigma_r)$, σ_i all positive values and r equal to the rank of X,
- V is a $n \times n$ orthogonal matrix.

We briefly remember that a matrix X is orthogonal if it holds that $X^\top X = XX^\top = I$, where I is the identity matrix. The columns $\{u_i\}_{i=1}^m$ of U are called *left-singular vectors* of X and are the eigenvectors of $XX^\top$ while the columns $\{v_i\}_{i=1}^n$ of V are called *right-singular vectors* of X and are the eigenvectors of $X^\top X$. Finally, the terms $\{\sigma_i\}_{i=1}^r$, where r is the rank of X, are called the *singular values* of X and are the square roots of the eigenvalues of $XX^\top$ and $X^\top X$.

A common way to interpret the SVD is by remembering that a matrix $X \in \mathbb{R}^{m,n}$ can be seen as a linear map going from $\mathbb{R}^n$ to $\mathbb{R}^m$ via the product $w = Xv$, where $v \in \mathbb{R}^n$ and $w \in \mathbb{R}^m$. Under this perspective, this linear transformation is divided into three steps (see Fig. 2.2)

1. The input vector v undergoes an orthogonal transformation via the matrix $V^\top$,
2. A non-negative scaling is applied on the entries of the rotated vector, possibly adapting its dimension,
3. Finally, the output vector w is obtained by performing a final orthogonal transformation via the matrix U.

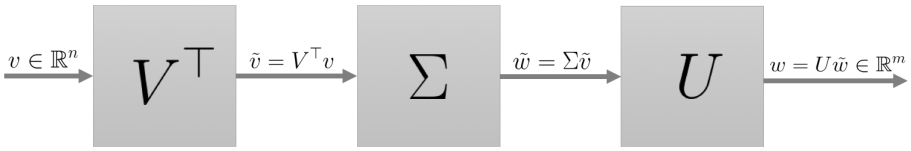


Figure 2.2: Block diagram representing the SVD as a chain of transformations

A full decomposition is usually unnecessary in real applications, instead alternative versions of the SVD are often used since they are faster and occupy smaller storage in comparison to a full SVD. A typical variation is the compact SVD

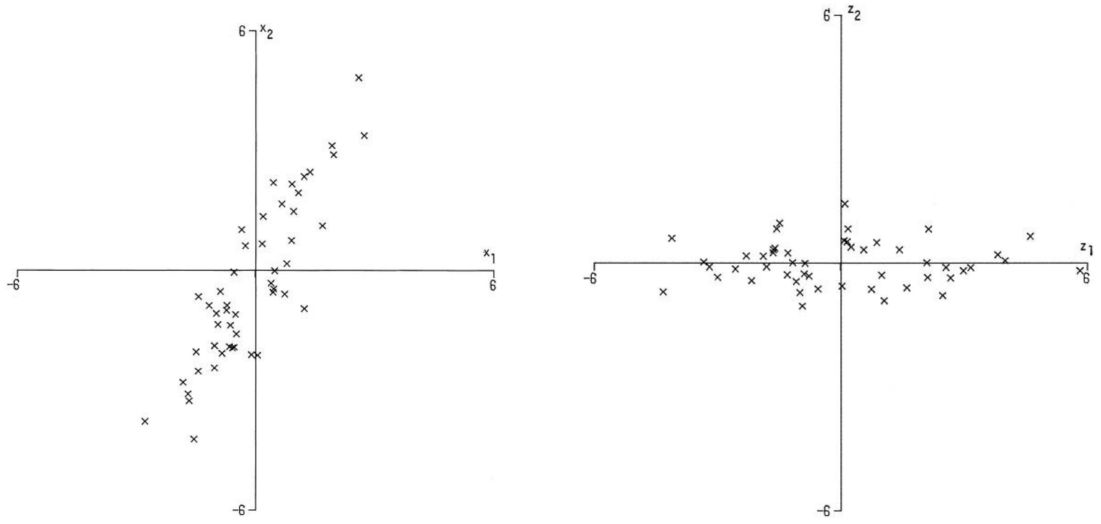
$$X = U_r \Sigma_r V_r^\top,$$

where we consider only the first r columns of U and r rows of $V^\top$, with r still the rank of X. The remaining $m - r$ and $n - r$ vectors are not calculated. This is equal to considering only the first r non-zero singular values of X. This is especially efficient when $r \ll n$.

The SVD, apart from allowing a direct solution to the problem of the PCA, is also a cardinal tool providing important insight on the spectral proprieties of a matrix. From it we can derive information, among other things, about the rank, the spectral norm and the condition number of the matrix undergoing the factorization.

2.2.2 PCA

Principal component analysis (PCA) is a widely known technique used to find the most important directions in a dataset [31]. These are orthonormal directions along which the data has the most variation and are called the principal components (PC). The PCs are a new set of variables ordered in such a way that the first few of them will retain the most amount of variation. Thus, by discarding all but the first few directions the data can be modeled along a low-dimensional subspace exposing the most relevant pieces of information. Recalling that dataset is composed by m Senators, each one of them represented by an element in $\mathbb{R}^n$, our objective is to find $w \in \mathbb{R}^n$, $\|w\|_2 = 1$, a normalized direction along which the variance of our data is maximized.



(a) (b)

Figure 2.3: Example of the change of reference system obtained through PCA [31]

In Fig. 2.3 we can see an example of what is the aim of principal components analysis. In Fig. 2.3.(a) it is clear that there is a preferred direction among which the data is distributed. PCA identifies this direction by defying a new orthogonal reference frame which is able to maximize the expressed variance of the data. In Fig. 2.3.(b) the direction z_2 can be discarded, reducing the data on a mono-dimensional subspace identified by the axis z_1 which is still able to correctly explain the dataset. To obtain the solution we firstly define:

$$\zeta_i = x^{(i)\top} w,\ i = \{1, \dots, n\},$$

which is the projection of $x^{(i)}$ along the the span of w i.e. the components of the data along the w direction. Now we can define:

$$\frac{1}{n}\sum_{i=1}^{n} \zeta_i^2 = \sum_{i=1}^{n} w^\top x_i x_i^\top w = w^\top X^\top X w,$$

that is the mean-square variation of the data along the w direction. To find w, we can cast an optimization problem:

$$\begin{aligned} \max_{w \in \mathbb{R}^m} \quad & w^\top X^\top X w \\ \text{s.t.} \quad & \|w\|_2 = 1. \end{aligned} \tag{2.13}$$

To solve it we can start from the compact form of the SVD of X:

$$X = U_r \Sigma_r V_r^\top = \sum_{i=1}^{r} \sigma_i u_i v_i^\top,$$

then defining $H \doteq X^\top X$, we can reformulate (2.13) as

$$\begin{aligned} \max_{w \in \mathbb{R}^m} \quad & w^\top H w \\ \text{s.t.} \quad & \|w\|_2 = 1, \end{aligned}$$

Where $H \in \mathbb{S}_+^m$ and its eigenvalues can be arranged from the maximum to the minimum one, $\{\lambda_{\max}(H) > \ldots > \lambda_{\min}(H)\}$. In the objective function we can recognize the maximization of a quadratic form of the matrix H, which is directly linked to the Rayleigh quotient induced by H. It is proven via the spectral theorem [33] that the maximum coincides with the largest eigenvalue of H and the maximizer is the eigenvector associated to this eigenvalue. Since we can obtain the spectral factorization of H starting from that of X

$$H = V_r \Sigma^2 V_r^\top,$$

the largest eigenvalue of H is σ_1^2 and the direction along which this variance is expressed is the first column of the matrix V_r i.e. v_1. As such, v_1 is the first principal component of the matrix X. To find the other principal components we can proceed by deflation. We first deflate the data by removing the component along which we found the largest variation:

$$x_I^{(i)} \doteq x^{(i)} - v_1 (x^{(i)\top} v_1) \; i = \{1, \ldots, m\}.$$

Repeating this computation for all the rows of X, we obtain the deflated matrix X_I

$$X_I = \begin{bmatrix} x_I^{(1)\top} \\ \vdots \\ x_I^{(m)\top} \end{bmatrix} = X - X v_1 v_1^\top = X(I - v_1 v_1^\top)$$

(2.13) can be recasted using X_I in the objective function

$$\begin{aligned} \max_{w \in \mathbb{R}^m} \quad & w^\top X_I^\top X_I w \\ \text{s.t.} \quad & \|w\|_2 = 1. \end{aligned}$$

To solve it, we can derive the compact form of the SVD of X_I starting from that of X:

$$\begin{aligned} X_I &= \sum_{i=1}^{r} \sigma_i u_i v_i^\top - \sum_{i=1}^{r} \sigma_i u_i v_i^\top v_1 v_i^\top \\ &= \sum_{i=1}^{r} \sigma_i u_i v_i^\top - \sigma_1 u_1 v_1^\top \\ &= \sum_{i=2}^{r} \sigma_i u_i v_i^\top, \end{aligned}$$

where in the second equation we exploited the notion that $\{v_i\}_{i=1}^m$ form an orthonormal set, so the product $v_i^\top v_1$ is equal to zero for each $i \neq 1$. Defining $H_I \doteq X_I^\top X_I$, we obtain a reformulation of the problem similar to that of the previous case

$$\begin{aligned} \max_{w \in \mathbb{R}^m} \quad & w^\top H_I w \\ \text{s.t.} \quad & \|w\|_2 = 1, \end{aligned}$$

where now the factorization of H_I is

$$H_I = V_{r_2} \Sigma^2 V_{r_2}^\top.$$

With V_{r_2} we are indicating the matrix V_r deprived of the first column v_1. Adopting the same reasoning of before, the maximum is now obtained with the largest eigenvalue of H_I which is σ_2^2 and the maximizer is the second column of the matrix V_r, i.e. v_2 which corresponds to the second principal component. This procedure can be iterated r times until r directions are found which are in descending order of data variation. By choosing $1 \leq k \leq r$ we can project our dataset X on these k principal components obtaining at the end a $m \times k$ reduced matrix $X_k = XV_k \in \mathbb{R}^{m,k}$, where with V_k we mean the first k columns of the matrix V_r. It is worth to notice that the principal components both captures the maximum variability among the columns of X and are all uncorrelated to each other, allowing us to consider only one of them without referring to the others. The number of chosen dimensions is a design parameter dictated primarily by the need to visualize the data (see Section 2.3) and by the amount of variance present in the original dataset needed to be explained by the reduced dataset, creating the need for a trade-off strategy. In this regard it is useful to define a ratio

$$\text{E-Var} = \frac{\sigma_1^2 + \ldots + \sigma_r^2}{\sigma_1^2 + \ldots + \sigma_k^2},$$

which is the amount of variance retained by only keeping the first k principal components, briefly remembering that σ_i^2 is proportional to the mean-square variation along the i-th direction. Relatively high values of E-Var (which by experience are $\geq \sim 30\%$) mean that the data observed on the projected k-dimensional subspace has kept a fair amount of information with respect to the original dataset.

2.2.3 Sparse PCA

The main issue with PCA is that each one of the computed directions is a linear combination of all the elements in $\mathbb{R}^n$, i.e. they are a combination of all the bills belonging in our dataset. This proves challenging when trying to obtain interpretability on the directions: in our case it would be interesting to know what were the bills that played a major role in increasing variance and separating the Senators. For this reason a sparse approach has been implemented based on the work of Zou. *et al.* [35] adopting the SpaSM toolbox for MATLAB [36]. In the Sparse PCA method a constraint is added to (2.13) in order to limit the number of non-zero elements composing the principal direction:

$$\begin{aligned} \max_{w \in \mathbb{R}^m} \quad & w^\top X^\top X w \\ \text{s.t.} \quad & \|w\|_2 = 1 \\ & \|w\|_0 \leq p, \end{aligned} \tag{2.14}$$

where p is the desired level of cardinality of the PC imposed directly as a constraint on the ℓ_0-pseudonorm of w. This approach however is computationally impractical for a high number of dimension since it is a hard combinatorial problem [38]. A relaxed approach is described by [35] and is the one adopted in this work. Other kinds of relaxations have also been proposed, e.g. El Ghaoui *et al.* [37] showed that the Sparse PCA problem can be approximated with a semi-definite programming (SDP) optimization problem involving the maximization of the trace of a symmetric matrix, which is then eventually subjected to truncation. To introduce our chosen approach we have to reframe the classic PCA algorithm under a compression standpoint. Specifically, finding the directions among which the variance of the data is maximized translates into the problem of finding the matrix A which projects the data on a lower dimensional subspace such that the projection made by $AA^\top$ reconstructs the data point $x^{(i)}$ as well as possibile. This can be casted as

an optimization problem with the constraint $A^\top A = I$ to impose the orthogonality of the solution

$$\begin{aligned} \arg\min_{A} \quad & \left\| X - XAA^\top \right\|_F^2 \\ \text{s.t.} \quad & A^\top A = I, \end{aligned} \tag{2.15}$$

Where $X \in \mathbb{R}^{m,n}$ and $A \in \mathbb{R}^{n,k}$, i.e. we are projecting the data on a k-dimensional subspace through the k columns of A with $k < n$. The solution of (2.15) is $A = V_k$ where $V_k = \{v_i\}_{i=1}^k$ are the first k columns of the matrix V obtained via the SVD of X. This is the same problem exposed in Section 2.2.2. Zou *et al.* [35] describes a reformulation of (2.15) where the problem of finding sparse principal components is translated into an elastic net optimization problem with a sparse matrix B as argument. In general, an elastic net problem has a penalty term on both the ℓ_2 and the ℓ_1-norm of the variables in the objective function. In this case, the ℓ_1 penalization is used to relax the cardinality constraint of (2.14) while the ℓ_2 penalization is used to provide unique solutions also when $m > n$ [36]. Since both A and B are orthogonal matrices, we can estimate the columns in a sequential way. For the i-th columns α_i and β_i the problem is casted as

$$\begin{aligned} \arg\min_{\alpha_i, \beta_i} \quad & \left\| X - X\beta_i\alpha_i^\top \right\|_F^2 + \delta \left\| \beta_i \right\|_2^2 + \lambda_i \left\| \beta_i \right\|_1 \\ \text{s.t.} \quad & A_k^\top A_k = I, \end{aligned} \tag{2.16}$$

where $A_k = [\alpha_1, \ldots, \alpha_k]$. 2.16 can be solved by the general SPCA algorithm explained by Zou in [35]. In this work we adopted the variation of this algorithm which utilizes a soft-thresholding rule to impose sparsity on the columns of B denominated the *Gene Expression Arrays SPCA Algorithm*. This algorithm, as the name suggests, was born in the context of gene analysis where high-dimensional data have a number of observations higher than the number of variables, such as in our case. Moreover, it is desirable to simplify the general SPCA algorithm to boost the numeric computation of the solution. This is achieved by setting $\delta = \infty$, causing the estimation algorithm to turn into the soft-thresholding function (see Fig. 2.4) This procedure is exposed in Algorithm 3.

Algorithm 3 Sparse PCA Algorithm in case $\delta = \infty$ for k PCs

Initialize A_k as the first k ordinary PCs
while the sparse vectors in B have not converged **do**
  for each α_i in $A_k = [\alpha_1, \ldots, \alpha_k]$ **do**
    Compute $\beta_i = \left(|\alpha_i^\top X^\top X| - \frac{\lambda_i}{2} \right)_+ \text{Sign}(\alpha_i^\top X^\top X)$ (soft-thresholding function [35])
  end for
  Update the matrix A_k
end while
Normalize each $\beta_i = \frac{\beta_i}{\|\beta_i\|_2}$, $i = \{1, \ldots, k\}$
Return $B = [\beta_1, \ldots, \beta_k]$

$\lambda_i/2$ is the parameter regulating the sparsity value i.e. the cardinality p of each column β_i. It should be chosen to obtain a reasonable trade-off between sparsity and variance (see also the Sparse PCA results in 2.3). Fig. 2.4 gives an illustration of how the soft-thresholding rule operates with $\lambda_i/2 = \Delta$. It should be noticed when analyzing the results that for our purposes we decided to adopt the same coefficient λ_i for each column.

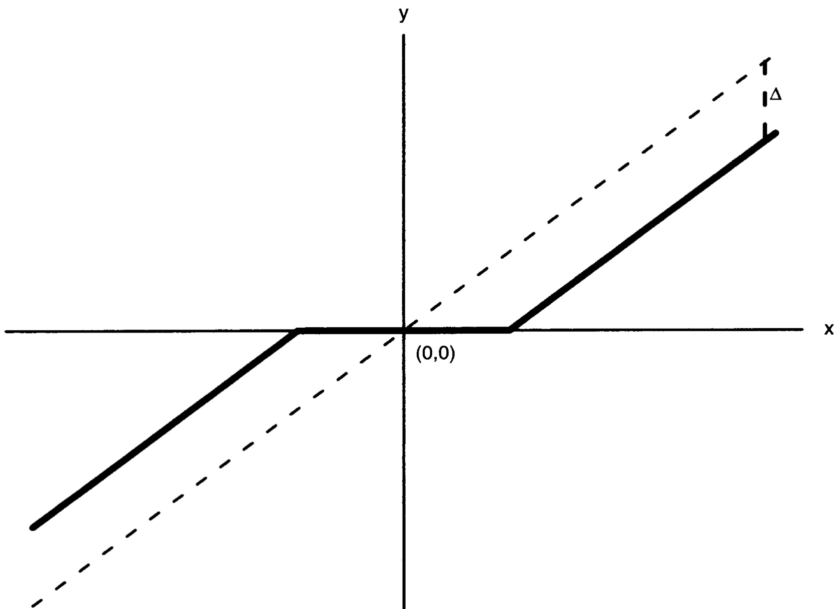


Figure 2.4: Illustration of the soft-thresholding rule $y = (|x| - \Delta)_+ \text{Sign}(x)$ adpoted in Algorithm 3 [35]

The results for all three of the explained techniques are exposed in the following section.

2.3 Results

These results have been obtained by projecting the data on a 2-dimensional and 3-dimensional subspace for every techniques exposed in the previous section. For the Sparse PCA case also different level of sparsity were considered. All the plots share the same following legend:

+	PD
☆	M5S
✳	Lega
×	PdL
△	NCD
♢	LeU
○	Other

Table 2.2: Legend of the political groups adopted both in the DM and outlier analysis plots and in the political maps

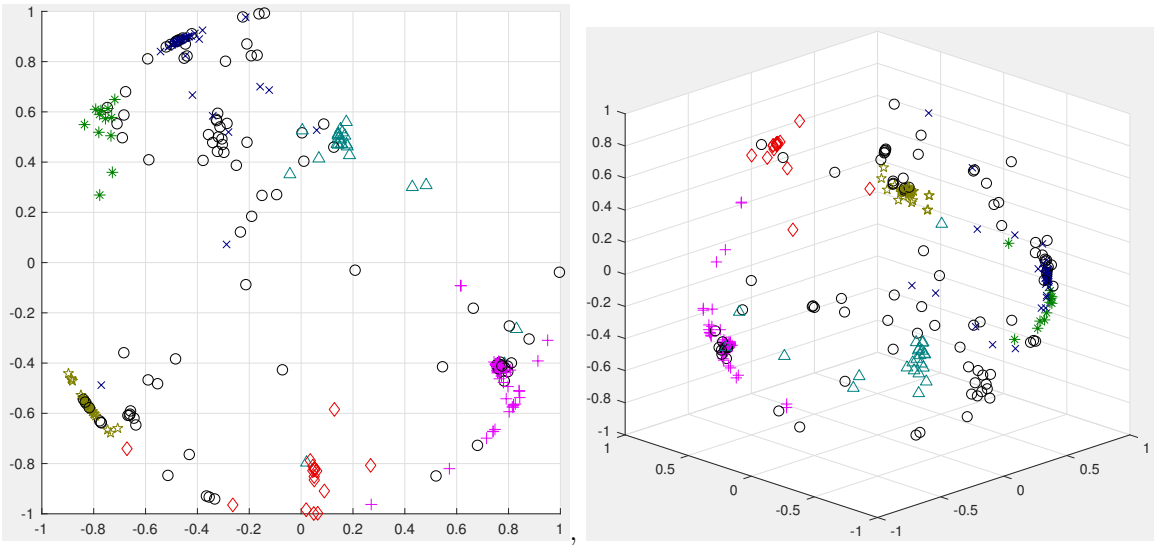
In Fig. 2.5 is shown the dataset reduced using the W-NOMINATE procedure. In particular in Fig. 2.5.(a) is shown the dataset reduced considering $k = 2$ dimensions and in Fig. 2.5.(b) the dataset reduced considering $k = 3$ dimensions. As a mean of comparison the values of APRE and GMP (see Section 2.1.2) have been specified for each plot.

In Fig. 2.6 is shown the dataset reduced using PCA. In particular in Fig. 2.6.(a) is shown the dataset reduced considering $k = 2$ PCs while in Fig. 2.5.(b) the dataset reduced considering $k = 3$ PCs. As a mean of comparison the E-Var value (see Section 2.2.2) has been specified for each plot.

Finally, in Figs. 2.7 and 2.8 is exposed the dataset reduced using Sparse PCA and considering, respectively, $k = 2$ and $k = 3$ sparse PCs, for different level of sparsity. Specifically, in Fig. 2.7 is shown the plot obtained projecting the dataset on $k = 2$ sparse principal components enforcing a degree of sparsity equal to $p = 10$ (Fig. 2.7.(a)), $p = 30$ (Fig. 2.7.(b)) and $p = 50$ (Fig. 2.7.(c)). Instead, in Fig. 2.8 is shown the plot obtained projecting the dataset on $k = 3$ sparse principal components enforcing a degree of sparsity equal to $p = 10$ (Fig. 2.8.(a)), $p = 30$ (Fig. 2.8.(b)) and $p = 50$ (Fig. 2.8.(c)). Also in this case the E-Var value has been specified for each plot as a mean of comparison.

Some individual considerations for the various techniques are made in the following sections.

2.3.1 W-NOMINATE



(a) $k = 2$
APRE = [0.763 0.915]
GMP = [0.841 0.924]

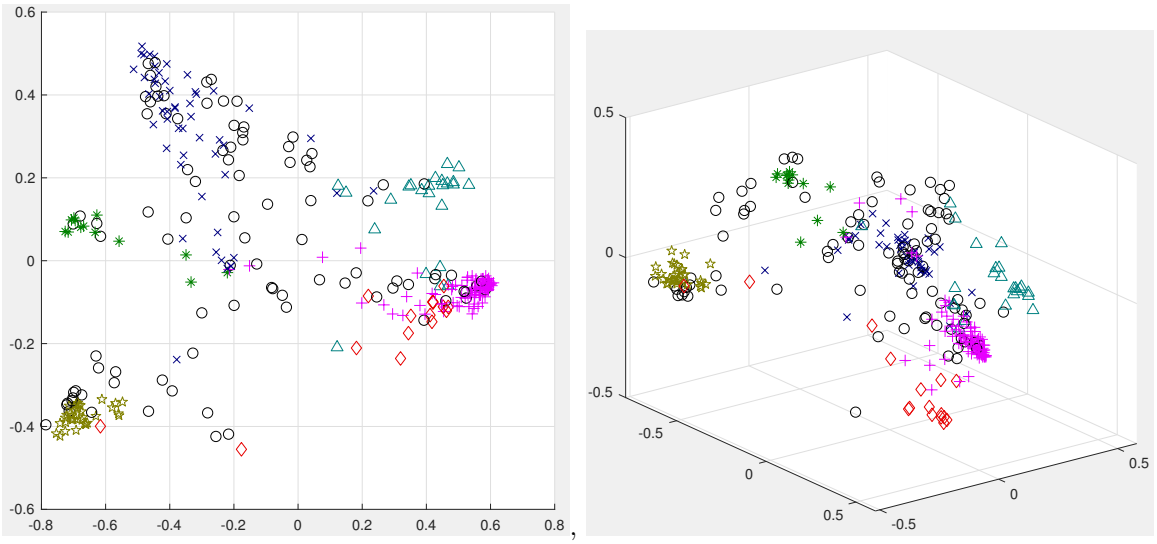
(b) $k = 3$
APRE = [0.764 0.914 0.959]
GMP = [0.841 0.924 0.945]

Figure 2.5: Dimensionality reduction using W-NOMINATE in $k = 2$ and $k = 3$ dimensions.

Here we can see the dataset reduced to $k = 2$ and $k = 3$ dimensions via the scaling method performed by the W-NOMINATE procedure. Some considerations that my hold are:

- The procedure has been able to explain the political structure of the Italian Senate, identifying the aggregation of the various political parties. This is also supported by the relatively high values of the APRE and GMP measures. It is worth to remember here that the algorithm works on unlabeled data.
- The $k = 2$ plot offers a possible interpretation of the axes direction. The vertical direction can be thought as an indicator of the political ideology of the represented parties (see Table 1.1), with the right-wing parties placed on the positive values and the left-wing parties (with the possible exception of the M5S group) placed on the negative values. The horizontal axis may instead be interpreted as a separator between the ruling parties, in particular the PD placed on the far right, and the opposition parties, in particular Lega and M5S placed on the far left.
- The additional dimension in the $k = 3$ doesn't convey a significative amount of new information. This can be seen both by the fact that third axis doesn't offer any kind of new interpretation of the data and that the values for the APRE and GMP measures on the third dimension are very similar to those of the second dimension.

2.3.2 PCA



(a) PCA, $k = 2$
E-Var = 63.38%

(b) PCA, $k = 3$
E-Var = 67.50%

Figure 2.6: Dimensionality reduction using PCA with $k = 2$ and $k = 3$ PCs.

Here we can see the dataset reduced to $k = 2$ and $k = 3$ dimensions by mean of PCA. Some considerations that my hold are:

- PCA has been able to identify a political structure underlying the Italian Senate, in some way similar to the one identified by the W-NOMINATE procedure (Fig. 2.5). This is also supported by the relatively high E-Var value (more than 60%), indicating that by retaining only $k = 2$ PCs the algorithm is capable of explaining a significative amount of data.

- An axes interpretation can be made similar to the one proposed in the W-NOMINATE case.

- The M5S group is the most compact one, with members that don't spread out on the plot. This is reasonable considering that M5S Senators have to follow a code of conduct preventing them to go against the group guidelines [7].

- The LeU group is very close to the PD group: this is reasonable since the foundation of the LeU group is linked to an internal split of the PD group.

- The third dimension added in the $k = 3$ plot, also in this case, doesn't explain a significative amount of additional data also in this case. This can also be seen by the E-Var value incrementing of only about 4%.

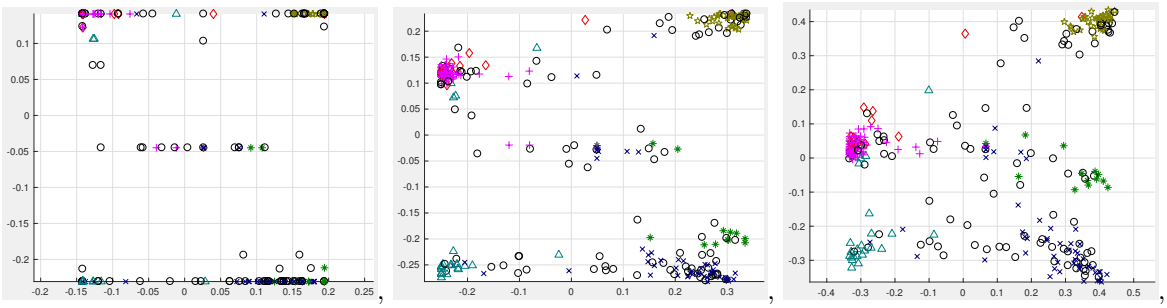
2.3.3 Sparse PCA

The main objective of the Sparse PCA analysis is to give a degree of interpretability to the principal components. This is achieved by imposing sparsity on the elements of the PCs and we should focus on the elements that are not set to zero by the algorithm. Since we are operating inside the bill space, this elements corresponds directly to the subset of bills that are capable of explain the most amount of information in the data. An example is shown in Table 2.3 where we list the bills identified on the first PC with a degree of sparsity $p = 10$.

Date	Description
26-11-2013	Stability Law 2014 - Vote of confidence
27-11-2013	Budget Law 2014 - DDL n. 1121. Final vote
05-12-2013	Decree Extending Military Missions
11-12-2013	Vote of confidence to the Letta Government
23-12-2013	Stability Law 2014 - Vote of confidence
24-02-2014	Vote of confidence to the Renzi Government
05-06-2014	IRPEF Decree
08-10-2014	Enabling act- Jobs Act
19-12-2014	[Legge di Bilancio 2015] Budget Law 2014 - Bill n. 1699. Final vote
25-06-2015	Bill "The Good School"

Table 2.3: Bills identified by Sparse PCA ($k = 10$, $p = 10$).

Regarding the plots, the analysis shares many aspects with the PCA case described in the previous section. Some considerations can be done specifically for the Sparse PCA case.



(a) Sparse PCA, $k = 2$, $p = 10$ E-Var = 10.01%

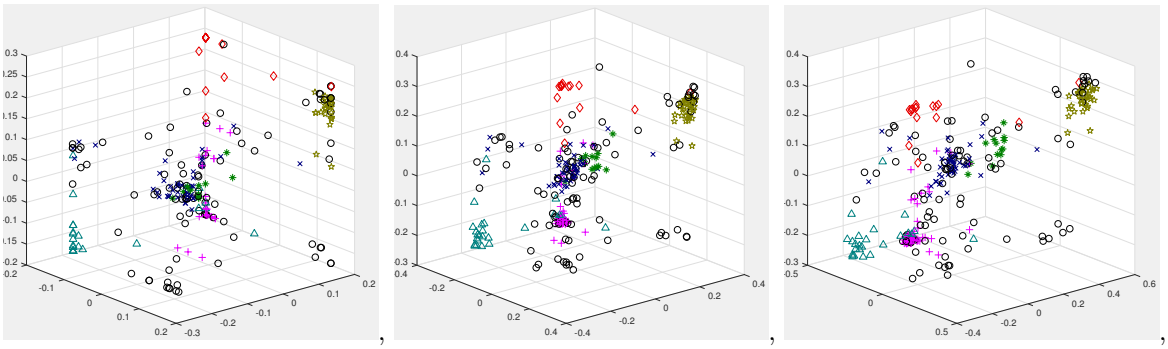
(b) Sparse PCA, $k = 2$, $p = 30$ E-Var = 19.76%

(c) Sparse PCA, $k = 2$, $p = 50$ E-Var = 31.26%

Figure 2.7: Dimensionality reduction with Sparse PCA using $k = 2$ PCs and different level of sparsity p.

- As expected, the E-Var value increase proportionally with the increment of the sparsity level.
- For high degree of sparsity (see Fig. 2.7.(a)) the Senators are not well separated. This kind of behavior shows how the bills identified by the algorithm (see Table 2.3) have polarized the political orientation of the Senate.

- Also for the Sparse PCA case, the third dimension doesn't seem to have a relevant role in explaining the data. This can be primary seen by the E-Var value incrementing in average of about 3% with respect to the $k = 2$ case for each plot. This finally suggests that for our dataset only 2 dimensions are necessary to obtain enough insight on the data, justifying the use of a 2-D polytope for the political maps shown in Chapter 4.



(a) Sparse PCA, $k = 3$, $p = 10$
E-Var = 12.88%

(b) Sparse PCA, $k = 3$, $p = 30$
E-Var = 23.44%

(c) Sparse PCA, $k = 3$, $p = 50$
E-Var = 34.87%

Figure 2.8: Dimensionality reduction with Sparse PCA using $k = 3$ PCs and different level of sparsity p.

Chapter 3

Outliers Detection

Aim of this Chapter is to state a set of measures able to define how much a Senator is distant from their nominal political group. This is basically the idea behind *cluster analysis*. The goal of cluster analysis is to group a set of objects (in our case the Senators) in such a way that the objects belonging to in the same group expose a degree of similarity between each other higher with respect to the objects belonging to the other groups. It sees its birth in the anthropology field [42] and has since expanded as a cardinal tool used in machine learning [43], image analysis [44], gene expression [45] and several other fields.

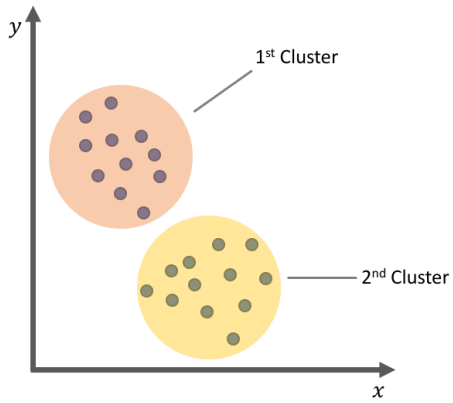


Figure 3.1: Simple example of the results following cluster analysis

There is not a specific algorithm that solves the problem of cluster analysis and the solution can be achieved by mean of various techniques. One of the most common involves the use of a *centroid model*, i.e. each cluster is represented by a single mean vector called a centroid. The most representative algorithm belonging to the class of centroid models is the k-means algorithm, which is exposed in Section 3.1. Another technique considered to perform the cluster analysis involves the

computation of Minimum Volume Ellipsoids (MVE). Having a set of points lying in a geometric space, a MVE is the one covering all the members of the set while minimizing the volume. Since in our case the sets are represented by the political groups composing the Senate, the MVE algorithm needs to work with labeled data. It is illustrated in 3.2. Finally, during the research leading to this thesis an innovative method for the detection of outliers in a dataset reduced with Sparse PCA has been developed and is explained in Section 3.3.

3.1 K-means

K-means is one of the simplest technique of *unsupervised learning*, i.e. the algorithm works on a unlabeled set of data that has not been priorly classified: purpose of the algorithm is in fact to associate a label (i.e. a cluster) to each of the data points. It is possibly the most famous and used clustering algorithm in literature [48]. It has been initially proposed independently by Lloyd [46] and Steinhaus [47]: the fundamental idea is to partition a population of N members into k sets or clusters starting from a sample of the population. This has both the aims of finding outlying members in the clustered population (such as in this work) and also to possibly reduce the size of the starting dataset, e.g. by replacing one or more of the clusters with a representative to increase the efficiency of subsequent numerical analysis.

To classify the observations into clusters, it is necessary to define a measure of distance in order to evaluate the similarity between each pair of observations. If $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_n)$ are two elements of a n-dimensional normed space, the most general metric that can be defined between the the two points is the Minkowski distance

$$d_{mink}(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i| \right)^{1/p} . \tag{3.1}$$

We are interested in particular for the cases where $p = 1$ and $p = 2$. For $p = 1$ (3.1) becomes the Manhattan distance (also called city block distance)

$$d_{man}(x, y) = \sum_{i=1}^{n} |x_i - y_i|, \tag{3.2}$$

while for $p = 2$ it becomes the Euclidian distance

$$d_{euc}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} = \|x_i - y_i\|_2^2 . \tag{3.3}$$

The choice of which distance to use to perform the cluster analysis is of critical importance, influencing the shape of the clusters and so the labels assigned by the algorithm. It has been shown [49] that for high dimensional data (3.2) performs better than (3.3). While it is true that our dataset initially belongs to a high dimensional space, throughout this work we will intensely adopt the dimensionality reduction techniques illustrated in Chapter 2, especially reducing the data to a 2-dimensional space: as such, our distance of choice will be the Euclidian one which also corresponds to that used in the standard k-means implementation.

K-means bases its approach on the problem of minimizing the sum-of-squared errors (SSE) [50] between the data points and the centroids representing the clusters. Formally, given a dataset $X = [x_1, \dots, x_m]$ where $x_i \in \mathbb{R}^n$, the goal is to find K sets $\boldsymbol{\mathcal{C}} = \{\mathcal{C}_1, \dots, \mathcal{C}_K\}$ with $K \in \mathbb{N}$. These sets are the clusters and they are determined such that the sum of the squared distances of the elements in X from the nearest centroid μ_j of the j-th cluster $\mathcal{C}_j$ is minimized

$$\arg\min_{\boldsymbol{\mathcal{C}}} \quad \sum_{j=1}^{K} \sum_{x \in \mathcal{C}_j} \|x - \mu_j\|_2^2 , \tag{3.4}$$

where μ_j is the mean of the points in $\mathcal{C}_j$ and the last term is the squared Euclidian distance. The standard procedure is exposed in Algorithm 4, and is commonly referred to as the Lloyd's algorithm [46].

Algorithm 4 Lloyd's algorithm

Choose K initial centroids (see later for details about initialization)
while the centroids have not converged **do**
  for each data point x_i **do**
    Assign x_i to the cluster with the least squared Euclidian distance i.e,
    Build the i-th cluster as
    $\mathcal{C}_i = \{x_p : \|x_p - \mu_i\|_2^2 \leq \|x_p - \mu_j\|_2^2 , \forall j, 1 \leq j \leq K\}$
  end for
  Compute the new centroids as $\mu_j = \frac{1}{|\mathcal{C}_j|} \sum_{x_i \in \mathcal{C}_j} x_i$
end while
Return the set of clusters $\boldsymbol{\mathcal{C}} = \{\mathcal{C}_1, \dots, \mathcal{C}_K\}$

Problem (3.4) is in general NP-hard in the Euclidian space and Algorithm 4 heuristically computes a sub-optimal solution [51].

Regarding initialization, Celebi *et al.* [52] showed that the placement of the centroids in the first iteration of the algorithm can drastically influence performance. In this work we adopted the k-means++ initialization technique [53]. The idea behind it is intuitively simple: the first K centroids should be spread out on the data space to ensure a good convergence. To obtain

this spreading, the first centroid is chosen uniformly at random from the data points while the successive ones are chosen from the remaining data points with a probability that is proportional to the squared Euclidian distance of the from the closest existing centroids. The procedure is illustrated in Algorithm 5.

Algorithm 5 K-means++ initialization

Choose the first centroid μ_1 uniformly at random from the data points.
while K centroids have not been chosen **do**
  for each data point x_i **do**
    Compute the distance $\mathrm{d}(x_i) = \|x_i - \mu_j\|_2^2$ where μ_j is the nearest existing centroid to x_i
    Choose a new centroid from the data point with a probability P proportional to $\mathrm{d}(x_i)$
  end for
end while
Return the set of centroids $\mathcal{M} = \{\mu_1, \ldots, \mu_K\}$

Finally, the choice of the number K of clusters is an important design parameter that heavily influences the correctness of the algorithm. A tool that is able to analyze the consistency of clustered data is Silhouette Analysis (SA) [54]. The silhouette value ranges from -1 to 1 and is assigned to each element of the dataset. The closer the value is to 1, the more the data point is well matched to the cluster to which it belongs and poorly matched to all the other clusters. This means that if most elements have a high silhouette value, the clustering is well performed. On the other hand if many elements have a low or a negative silhouette value there may been too many or too few clusters describing the dataset. Formally, for each data point x_i we can define two quantities:

- $a(x_i)$, which is the average distance between x_i and all the other points in the same cluster. It represents how well x_i is assigned to its cluster: the smaller is the value of $a(x_i)$ the better is the assignment.
- $b(x_i)$, which is the average distance between x_i and all the other points belonging to its closest neighboring cluster. This is the cluster to which x_i doesn't belong but its average distance is the smallest among all the other clusters, i.e. the cluster with the smallest average dissimilarity with respect to x_i.

The silhouette is then defined as

$$s(x_i) = \frac{b(i) - a(i)}{\max\{a(x_i), b(x_i)\}} \tag{3.5}$$

From the definition it follows that $-1 \leq s(x_i) \leq 1$. In fact, if $b(x_i) \gg a(x_i)$, the dissimilarity with respect to the closest neighboring cluster of x_i is high with respect to $a(x_i)$, bringing $s(x_i)$ close

to 1. This implies that the x_i is well matched to its assigned cluster. The opposite consideration holds for values of $s(x_i)$ close to -1. In this thesis we analyzed the average silhouette value to decide the number of clusters to be used with the k-means algorithm and the results are exposed in Section 3.4.

3.2 Minimum Volume Ellipsoid (MVE)

The problem of covering a set of points with an ellipsoid having minimum volume is a classical geometric problem, firstly addressed by Löwner in [55]. It then evolved primarily in the field of convex optimization [56] where several applicative algorithms have been developed in the recent years, e.g. [58, 57]. In particular, in the convex optimization framework the problem belongs to the field of Semidefinite Programming (SDP) which involves the optimization of a convex objective function (in this case representing the volume of the ellipsoid to be minimized) over a set of constraints represented by an affine combination of positive semidefinite matrices. This sub-field of optimization problems embodies the most general representation of Linear Programming (LP) problems. To cast the problem as an optimization problem, we firstly have to adequately characterize the ellipsoid. An ellipsoid $\mathcal{E}$ centered in the origin can be defined in the following way, as described by [33]

$$\mathcal{E} = \{x \in \mathbb{R}^n : x^\top P^{-1} x \leq 1\},$$

with $P \in \mathbb{S}^n_+$. The shape of $\mathcal{E}$ is defined by the eigenvalues of P, which by definition are all positive and act as scaling factors, and eigenvectors of P acting as the directions of the semi-axes of $\mathcal{E}$. Also, the volume of $\mathcal{E}$ is proportional to the determinant of P^{-1}. Since $P \in \mathbb{S}^n_+$ also $P^{-1} \in \mathbb{S}^n_+$ and by the Cholesky decomposition there exists a matrix A such that $P^{-1} = A^\top A$. In this way, $x^\top P^{-1} x = x^\top A^\top A x = \|Ax\|_2^2$. Considering a generic point of $\mathbb{R}^n$ as the center of $\mathcal{E}$, we can then write:

$$\mathcal{E} = \{x \in \mathbb{R}^n : \|Ax + b\|_2 \leq 1\}, \tag{3.6}$$

interpretable as an affine transformation of the unit ball in the Euclidian space. The optimization problem can then be casted in the following form

$$\begin{aligned} \underset{A,\, b}{\arg\min} \quad & \log\det A^{-1} \\ \text{s.t.} \quad & \|Ax + b\|_2 \leq 1, \\ & A \in \mathbb{S}_+, \end{aligned} \tag{3.7}$$

remembering that any monotone increasing transformation of the objective function (such as taking the logarithm) results in an equivalent optimization problem. Since both the objective and the constrains are convex in A and b, (3.7) is convex and solvable for example under MATLAB adopting the CVX toolbox, as in our case. The resulting ellipsoid parametrized by A and b is the MVE. The adoption of the MVE as a clustering tool is already present in literature, e.g. [59]. In this work we followed a customized interpretation of the problem, resulting in a simpler but less robust algorithm. Advantages and disadvantages of our technique are shown in Section 3.4. Basically, the chosen method involves the computation of the MVE covering only an arbitrary percentage $\alpha\%$ of the points in a set. In this way, the points left outside of the ellipsoid are directly the outliers. With respect to the k-means algorithm exposed in Section 3.1, this can be seen as a technique of *supervised learning*. In the supervised learning context, the algorithm has to work with a dataset that has been previously classified, i.e. all the data points should be labelled. In our context the data points, i.e. the Senators, have been labeled with their nominal affiliation to one of the political groups of the Senate (see Algorithm 2). As such, we have to compute n_g ellipsoids where n_g is the number of considered political groups. The procedure is illustrated in Algorithm 6.

Algorithm 6 Computing the ellipsoid covering the $\alpha\%$ of the set

```
Choose the α% of the set to keep inside E
Compute the number k of points to be left out of E as card(x) · α/100
Set the initial number ω of outsiders to 0
Solve (3.7) for A and b
Check for which points x̄_i the constraint is active, i.e. Ax̄_i − b = 1
while ω < k do
    for each x̄_i do
        Remove x̄_i from the set
        Solve (3.7) for A_i and b_i
        Compute the i-th volume V_i as 1/(det A)
    end for
    Choose as solution A_i and b_i corresponding to the minimum V_i
    Update the value of ω
end while
Return A_i
Return b_i
```

The chosen method is to firstly solve (3.7) and obtain the MVE. Then, we check for which data points the constraint of the problem is active, i.e. which are the data points lying on the boundary of the ellipsoid. We procede to remove each one of these data points from the set and solve the problem again. From the resulting ellipsoids, we choose the one with the minimum volume. This procedure is iterated until $\alpha\%$ of the original points are kept inside the ellipsoid. The results exposed in Section 3.4.

3.3 Automatic outliers selection via Sparse PCA

In this section we show how the Sparse PCA technique (see Section 2.2.3) may be adopted to automatically select the outlying Senators, i.e. those Senators whose voting behavior is not representative of their nominal political group. The process has born during the research underlying this thesis. The fundamental insight leading to the discovery of this technique was to consider the transpose of the data matrix $X^\top$. $X^\top$ is a $n \times m$ matrix, with n being the number of bills and m the number of Senators, where each datum $x^{(i)}$ now represents a bill in the m-dimensional space of the Senators. Under this interpretation, the application of one of the dimensionality reduction techniques exposed in Chapter 2 will result in the projection of the bills on a lower dimensional subspace which directions are linear combinations of the Senators. If this kind of reduction is made with the classical PCA, all the directions will be a linear combination of all the Senators present in the dataset. By applying instead the Sparse PCA algorithm with a suitable level of cardinality p, it is possible to limit this combination to a prefixed number of Senators. With great interest we discovered that by adopting this methodology the principal components tend to naturally cluster with respect to the political groups of the Italian Senate. The analysis is summarized through the block diagram shown in Fig. 3.2.

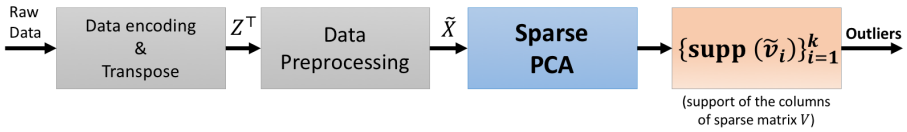


Figure 3.2: Block diagram representing the automatic extraction of outliers using Sparse PCA on the transposed matrix

The procedure is composed of the following steps:

- Firstly, the raw data acquired by mining the Italian Senate is encoded and cleaned.
- Then, the resulting matrix is transposed and encoded following the same method shown in Section 1.4.3.
- To this matrix the Sparse PCA algorithm is applied, by retaining only $k = 10$ principal components each of them with a value of cardinality $p = 50$.
- Finally, from the columns of V_k which are now a linear combination of $p = 50$ Senators and we extract only the non-zero elements.

The choice of k and p is also in this case a design parameter: $k = 10$ has been chosen to mirror the number of political groups present during the Legislature while $p = 50$ represents a reasonable

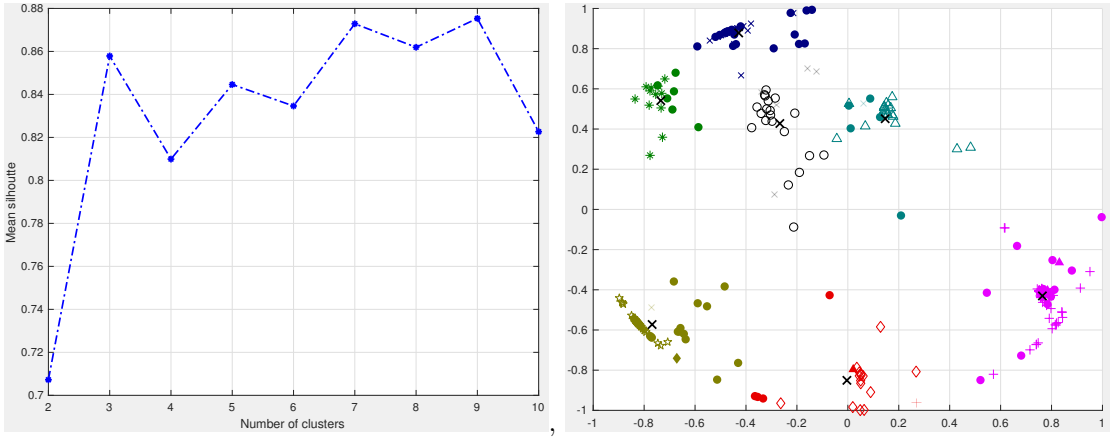
trade-off between the amount of E-Var (about 40%) and the degree of sparsity of the solution. The results obtained at the end of these three techniques of cluster analysis are exposed in the following Section.

3.4 Results

The results have been separated into the sections corresponding to each of the explained clustering techniques. In Section 3.4.1 are shown the results obtained by applying the k-means algorithm. In particular, in Fig 3.3 are shown the results of the cluster analysis made with k-means to the dataset reduced via the W-NOMINATE procedure in $k = 2$ dimensions, and in Fig. 3.4 the results obtained via k-means with the dataset projected on $k = 2$ principal components computed via PCA. In Fig. 3.5 are exposed the results obtained computing the MVE for each political group in the dataset reduced in a $k = 2$ dimensional subspace both via PCA and the W-NOMINATE procedure. The adopted legend is the one exposed in Table 2.2.

3.4.1 K-means

The k-means results are exposed in this section for the dataset reduced both via the W-NOMINATE procedure and with PCA. For the two cases, on left it is shown the evolution of the average value of the silhouette by choosing a number K of clusters going from 1 to 10. This plot is used as a guide, together with a visual inspection of the reduced dataset, to choose the K before proceeding with the analysis. The clusters are shown on the right, separated by the color of the corresponding political group (we apply the same legend of Table 2.2). The outliers have been indicated on the plot by a full marker, keeping the shape of original political group but with the color corresponding to the cluster to which the outlier has been assigned. The final values of the centroids are represented by a black cross marker.

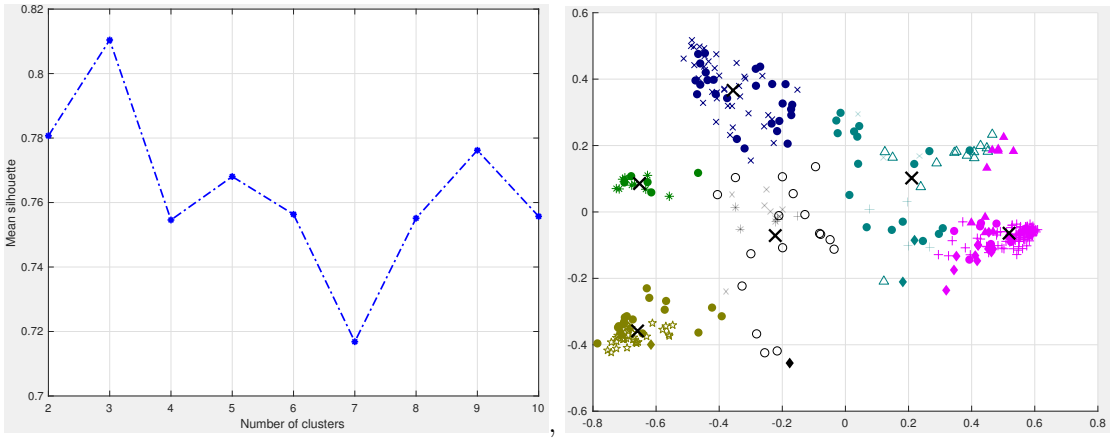


(a) Average value of the silhouette as a function of the number K of clusters

(b) Outliers in the clusters ($K = 7$)

Figure 3.3: Outliers identified by the K-means algorithm for the dataset reduced via W-NOMINATE retaining $k = 2$ dimensions.

In Fig. 3.3 we can see the k-means results for the dataset reduced via the W-NOMINATE procedure using $k = 2$ dimensions. In Fig. 3.3.(a) we can see that for $K = 7$ we have a silhouette value of about 0.88, which is a relatively large value indicating that with $K = 7$ clusters the algorithm can perform well. This can be seen also in Fig. 3.3.(b): the considered political groups have kept a shape on the plot that is very similar to that exposed before performing the clustering procedure. As expected, the members of the Other group are the one exposing the largest number of outliers, spreading out towards all the rest of the political groups. It can also be seen that M5S is the one exposing the least amount of outliers, in accordance to how this political group is the most cohesive of the XVII Legislature.



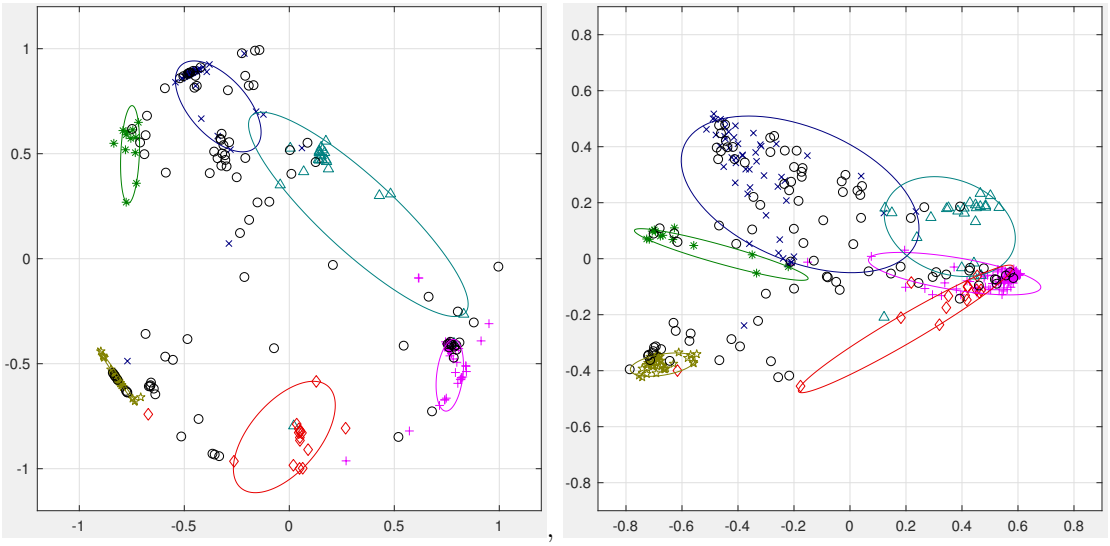
(a) Average value of the silhouette as a function of the number K of clusters

(b) Outliers in the clusters ($K = 6$)

Figure 3.4: Outliers identified by the K-means algorithm for the dataset reduced via PCA retaining $k = 2$ PCs.

In Fig. 3.4 we can see the k-means results for the dataset reduced with PCA procedure using the first two principal components. The main difference with respect to the W-NOMINATE case is that now for $K = 7$ the average silhouette value shown in Fig. 3.4.(a) is relatively small: almost 0.72 which is the minimum achieved by the function among all the possible values of K. This can be understood by looking at Fig. 3.4.(b): by projecting the data on the first $k = 2$ PCs, LeU and PD almost coincide on the plot. This is not a contradicting result: both parties shares a similar political ideology and LeU has been formed as a result of a separation from the PD party. Considering this, we decided to adopt for this case a value of $K = 6$ corresponding to a mean silhouette value of about 0.76. In this way all Senators of the LeU party are shown as outliers inside the PD group. A notable exception is the Senator Campanella Francesco which instead appears as an outlier of the M5S group. This result is coherent also with Sparse PCA analysis exposed in Section 3.3.

3.4.2 MVE



(a) Dataset reduced via W-NOMINATE with $k = 2$ (b) Dataset reduced via PCA with $k = 2$

Figure 3.5: Outliers identified by the MVE procedure covering 90% of the points.

In Fig. 3.5 are shown the results of the MVE procedure with $\alpha = 0.9$ applied to both the dataset reduced with W-NOMINATE and with PCA. In both cases we retained $k = 2$ dimensions. The main advantage of this kind of analysis is that it provides an intuitive way to graphically identify the outliers of a political group, since they simply are the ones outside the ellipsoid covering their nominal political group or the ones inside the ellipsoid covering one of the other political parties.

We can see how the ellipsoid with overall minimum volume is the one covering the M5S, confirming that it is the most compact party. Another advantage of this technique is that, since it belongs to the domain of supervised learning, we don't need to fix a-priori the number of clusters. They directly are equal to the number of political groups considered during the analysis. The main drawback of this algorithm is that it is not robust as much as k-means. The main reason behind it is that the value of α has to be decided a-priori and is equal for all the groups. This is an issue for those groups who are spread among the plot and have a relatively small amount of Senators. An example is given by the NCD group in Fig. 3.5(a). For $\alpha = 0.9$ the algorithm has to exclude only one Senator, and the resulting MVE is intuitively not the best cluster.

3.4.3 Sparse PCA

In this section we expose the results obtained by applying the Sparse PCA technique for the detection of outliers summarized in Fig. 3.2. In Table 3.1 we show for the first three PCs the dominant composition, with the corresponding frequency percentage.

PC	Most frequent party	Percentage
1st	PD	98%
2nd	M5S	70%
3rd	PdL	68%

Table 3.1: Political composition of the PCs obtained via Sparse PCA on the transposed matrix with $k = 10$ and $p = 50$.

As we already stated, the PCs are highly correlated with the parties, inducing a natural clustering. It is worth remarking that the clustering induced by decomposition obtained with Sparse PCA is robust against changes in the parameter p: the composition of PCs remain similar to the ones listed in the table for larger values of p. Once identified the principal component with party, we isolate the outliers, i.e. the Senators who are assigned to a principal component/party but who belongs to a different nominal group. It should be noticed that the parties identified in the 3 first PCs correspond to the 3 major parties present in the Senate during the XVII Legislature. More precisely, the first PC corresponds to the largest political group, i.e. the PD. In this component just one outlier is present, Fravezzi Vittorio. The list of outliers identified by the algorithm for the second and third PC is exposed in Table 3.2.

Outliers of the 2nd PC	Outliers of the 3rd PC
BAROZZINO Giovanni - (Misto)	BONFRISCO Anna Cinzia - (FL(Id-PL, PLI))
BIGNAMI Laura - (Misto)	BRUNI Francesco - (NcI)
BOCCHINO Fabrizio - (Misto)	CONTI Riccardo - (GaL-UDCeDC)
CAMPANELLA Francesco - (Art.1-MDP-LeU)	D'AMBROSIO LETTIERI Luigi - (NcI)
CASALETTO Monica - (GaL-UDCeDC)	FERRARA Mario Francesco - (GaL-UDCeDC)
CERVELLINI Massimo - (Misto)	LIUZZI Pietro - (NcI)
DE CRISTOFARO Peppe - (Misto)	MAURO Giovanni - (GaL-UDCeDC)
DE PETRIS Loredana - (Misto)	MAZZONI Riccardo - (ALA-PRI)
DE PIETRO Cristina - (FI-PdL)	PAGNONCELLI Lionello Marco - (ALA-PRI)
DE PIN Paola - (GaL-UDCeDC)	PERRONE Luigi - (NcI)
MUSSINI Maria - (Misto)	TARQUINIO Lucio Rosario - (NcI)
PETRAGLIA Alessia - (Misto)	VILLARI Riccardo - (GaL-UDCeDC)
SIMEONI Ivana - (Misto)	ZIZZA Vittorio - (NcI)
VACCIANO Giuseppe - (Misto)	

Table 3.2: Outliers identified by the Sparse PCA algorithm on the 2nd and 3rd PCs.

Looking at the second PC, the one corresponding to M5S, we notice that the most frequent membership of outliers is the Mixed group. Campanella Francesco is also present in this Table, coherently with the results obtained before both via k-means and MVE analysis. The fact that most of the M5S outliers are from the Mixed group is coherent with the behavior analyzed by a recent report regarding the migration of Italian senators during the XVII legislature [8]. Finally, in the third PC (corresponding to PdL) all outliers belong to parties whose political orientation is in the center. We emphasize that most of the detected outliers are senators who belonged to the party corresponding to the detected PC at beginning of the XVII Legislature and then migrated to other groups. For some of these outliers we expose the individual Political DNA in Section 4.3.

Chapter 4

Political Data aNalytic Affinity (DNA)

In this Chapter we introduce the main contribution of this thesis to the field of research regarding the inference of social and political influence. It is a measure, based on the data features extracted via the techniques explained in Chapter 2, aimed to summarize the degree of affinity (or equivalently that of *rebellion*) that a Senator exposes towards each political group. This index of similarity summarizes the influence that each political group exert on the Senator and it has been denominated the Political Data aNalytic Affinity (Political DNA) of a Senator. To infer this measure, we build a probabilistic model which is a Gaussian Mixture Model (GMM), explained in Section 4.1. We adopt this model to interpret the reduced dataset by modeling the votes as a mixture of random normal variables. Then, by computing the a-posteriori probabilities from the model density through the application of the Bayes' theorem, we can generate a vector of probabilities representing the Political DNA of a Senator. By creating a convex combination of the elements of this vector and the vertices of a regular polytope we generate a visual representation representation of the Political DNA called a political map, exposed in Section 4.2.1. Finally, by adopting the Sparse PCA technique for outliers detection (see Section 3.3) we expose the individual Political DNA of the outlying Senators in Section 4.3.

4.1 The Gaussian Mixture Model (GMM)

To introduce the concept of a mixture model, we can briefly recall the standard mono-dimensional (univariate) Gaussian distribution. It is known that, because of the central limit theorem, the

samples drawn independently from a random variable X of which the true nature is unknown (e.g. a source of noise in a measurement process) can be modeled with a normal behavior when the number of observations is large enough [60]. The probability density function (p.d.f.) of the Gaussian distribution is

$$f_X\left(x|\mu,\sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

where μ is the mean value of the distribution and σ^2 is the variance. The distribution is typically denoted with $\mathcal{N}(\mu,\sigma^2)$. Knowing the p.d.f. we can derive the probability that a random variable X assumes a range of values between a and b as $\int_a^b f_X(x)$ (where for simplicity we have not explicitly stated the dependence from the distribution parameters).

It is often the case however that the data we wish to model under a probabilistic structure is more complex. In particular the data may be composed by a *mixture* of several components, each exposing a simple parametric model [61]. To represent this model mathematically we need to introduce a set of *latent* or *hidden* variables z. These variables are identified by the algorithm and can not be directly observed (hence, hidden). In general, given a dataset $X = [x_1, \ldots, x_m]$ which we wish to model in a mixture fashion, we assume that the points $\{x_i\}_{i=1}^m$ are generated in an independent and identically distributed fashion (i.i.d.) from an underlying density function $f(x_i)$. If we assume that the model is composed by K components, we may write

$$f(x_i|\Theta) = \sum_{k=1}^{K} \alpha_k f_k(x|z_k,\theta_k), \tag{4.1}$$

where:

- $\{\alpha_k\}_{k=1}^K$ are the mixing weights. They represent the a-priori probability that the point x_i was drawn by the distribution f_k. Since they are a vector of probabilities it holds that $\sum_{k=1}^K \alpha_k = 1$ with each $\alpha_k \geq 0$. For this reason the mixed distribution is a convex combination of the mixture components.

- $\{z_k\}_{k=1}^K$ are the hidden variables of the mixture model. They are mutually exclusive and exhaustive, i.e. only one of the z_k can be 1 while the others are 0. As such, they represent the identity of the mixture component that generated x_i.

- Finally, $\{f_k\}_{k=1}^K$ are the mixture components. Each one of them is characterized by a set of parameters θ_k. The mixture model overall is characterized by a complete set of parameters $\Theta = \{\alpha_1, \ldots, \alpha_K, \theta_1, \ldots, \theta_K\}$.

In the case that $\{f_k\}_{k=1}^K$ are all normal distributions, characterized by $\theta_k = \{\mu_k, \sigma_k\}$, (4.1) is called a Gaussian Mixture Model (GMM). An example is exposed in Fig. 4.1.

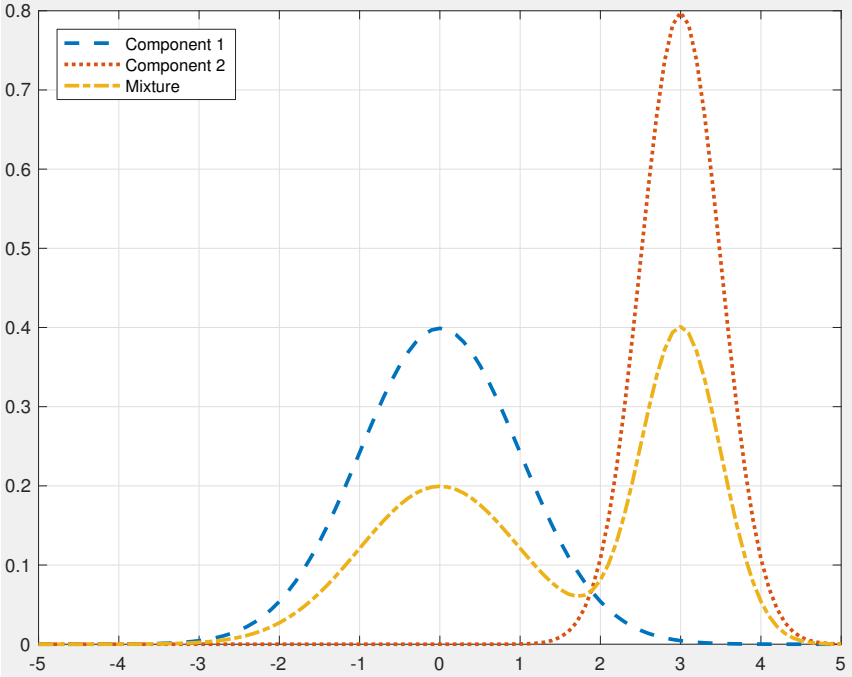


Figure 4.1: Example of a unidimensional Gaussian mixture with 2 components

The first component (in blue, dashed) is a Gaussian distribution $\mathcal{N}(0, 1)$ while the second component (in red, dotted) is a Gaussian distribution $\mathcal{N}(3, 0.5)$. The mixture (in yellow, dash-dotted) is obtained by combining both components with equal weights, i.e. $\alpha_1 = \alpha_2 = 0.5$.

In the context of the Italian Senate, we are assuming that the underlying model is a GMM where the generic mixture component f_i is a multivariate Gaussian distribution. The multivariate Gaussian distribution is a generalization of the univariate case. A m-dimensional random vector $X = [x_1, \ldots, x_m]$ is said to be normally distributed if every linear combination of its components has a univariate Gaussian distribution. It is expressed with the notation $X \sim \mathcal{N}(\mu, \Sigma)$, where μ is the m-dimensional mean vector $\mu \doteq [\mathrm{E}[x_1], \ldots, \mathrm{E}[x_m]]$ and $\Sigma \in \mathbb{S}_+^m$ is the covariance matrix $\Sigma \doteq \mathrm{E}\left[(X - \mu)(X - \mu)^\top\right]$. $E[\cdot]$ is the expectation operator.

The density of the multivariate normal distribution is the following

$$f(X) = \frac{\exp\left(-\frac{1}{2}(X - \mu)^\top \Sigma^{-1}(X - \mu)\right)}{\sqrt{(2\pi)^n \det(\Sigma)}}. \tag{4.2}$$

As such we assume that the vote vector of the i-th Senator represented by the i-th column of the vote matrix $x^{(i)}$ is distributed as a multivariate Gaussian. In particular, the density of this distribution is conditional to the class to which this vector belongs, which is the nominal political

group of the Senator, classified as by Algorithm 2. The ℓ-th class is characterized by its mean vector μ_ℓ and its covariance matrix Σ_ℓ. If we define G_ℓ as the set of indices such as that the i-th group g_i is equal to ℓ i.e. $G_\ell \doteq \{i : g_i = \ell\}$, with $\ell = \{1, \dots, n_g\}$ and n_g the number of considered political groups (see Section 1.1), we can compute μ_l and Σ_l by their maximum-likelihood (ML) estimators [62]

$$\mu_\ell = \frac{1}{|G_\ell|} \sum_{i \in G_\ell} x^{(i)},$$
$$\Sigma_\ell = \frac{1}{|G_\ell| - 1} \sum_{i \in G_\ell} \left(x^{(i)} - \mu_\ell\right)^\top \left(x^{(i)} - \mu_\ell\right).$$

The a-priori probabilities $\{\alpha_i\}_{i=1}^{n_g}$ can also be computed by means of the ML estimator $\alpha_\ell = \frac{|G_\ell|}{m}$, where we are supposing that the largest political group is prioritized when deriving the probability weights. By assuming that the data are realizations of a random variable generated in an i.i.d. fashion from a finite mixture of ℓ normal distributions with n_g components, we write the density

$$f\left(x^{(i)}\right) = \sum_{\ell=1}^{n_g} \alpha_\ell f_\ell \left(x^{(i)} | z_{i\ell}, \mu_\ell, \Sigma_\ell\right), \tag{4.3}$$

Equivalently, the data sample $x^{(i)}$ can be modeled as:

$$x^{(i)} \sim \begin{cases} \mathcal{N}\left(\mu_1, \Sigma_1\right), & \text{with probability } \alpha_1 \\ \vdots & \\ \mathcal{N}\left(\mu_{n_g}, \Sigma_{n_g}\right), & \text{with probability } \alpha_{n_g}, \end{cases} \tag{4.4}$$

and the hidden variable $z_{i\ell}$ is then determined as

$$z_{i\ell} = \begin{cases} 1, & \text{if } x^{(i)} \in \text{group } \ell \\ 0, & \text{otherwise.} \end{cases} \tag{4.5}$$

The GMM is fully characterized by the set of parameters $\{\alpha_\ell, \mu_\ell, \Sigma_\ell\}_{\ell \in \{1,\dots,n_g\}}$. A possible improvement on this procedure is the implementation of an Expectation-Maximization (EM) algorithm to further refine the model parameters, as shown by [63].

Once built the GM model, we can cast our problem as follows: for each Senator $s \in \{1,2,\dots,m\}$ we want to infer their Political DNA i.e. a vector $\pi^{(s)} \in [0,1]^{n_g}$ whose entries $\{\pi_g^{(s)}\}_{g=1}^{n_g}$ represent the influence of group g on Senator s with the propriety $\sum_{g=1}^{n_g} \pi_g^{(s)} = 1$. This vector can be derived exploiting the hidden variable defined in (4.5) with the procedure explained in the following section.

4.2 Computation and visualization of the Political DNA

The Political DNA is given by the a-posteriori probabilities computed on the hidden variable $z_{i\ell}$:

$$\pi_{il} = \mathbb{P}\left(z_{il} = 1 \mid x^{(i)}; \{\alpha_\ell, \mu_\ell, \Sigma_\ell\}_{\ell \in \{1,\dots,n_g\}}\right). \tag{4.6}$$

This probability represents the posterior belief that the i-th Senator belongs to the ℓ-th group based on the GM model, given the evidence provided by the vote vector $x^{(i)}$. This probability is computable by applying Bayes' theorem

$$\frac{\alpha_\ell \frac{\exp\left(-\frac{1}{2}(x-\mu_\ell)^\top \Sigma_\ell^{-1}(x-\mu_\ell)\right)}{\sqrt{\det(\Sigma_\ell)}}}{\sum_{j=1}^{n_g} \alpha_j \frac{\exp\left(-\frac{1}{2}(x-\mu_j)^\top \Sigma_j^{-1}(x-\mu_j)\right)}{\sqrt{\det(\Sigma_j)}}}. \tag{4.7}$$

(4.7) is heavily dependent on the estimation of the group covariance estimation and its inversion. This mainly generates two issues:

- The group covariance may suffer in accuracy if the group is too small, i.e. for parties with too many few members.
- The covariance matrix may be singular preventing its inversion. When the covariance matrix is not full rank, it is an indicator that the components are concentrated on a low dimensional subspace [64].

Mapping the dataset on a low dimensional subspace has both the advantages of solving these problems and allowing for a better diversification of the political composition of the Italian Senate.

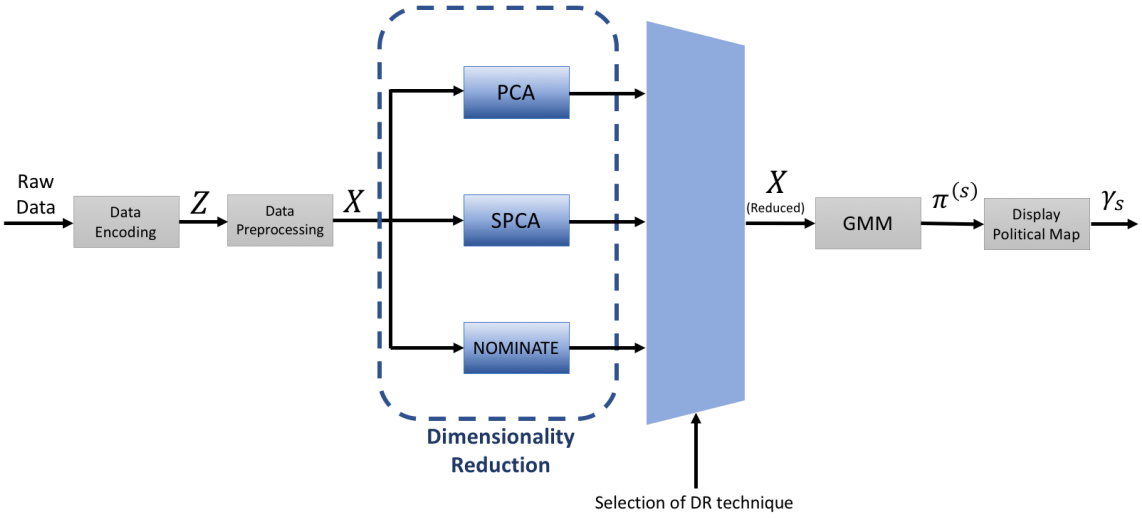


Figure 4.2: Block diagram summarizing the inference procedure of the Political DNA

The procedure applied to compute the Political DNA is summarized in Fig. 4.2. The raw data obtained at the end of the mining procedure (see Fig. 1.3) is encoded and standardized in the first block, as explained in Section 1.4.3. Then, the output is mapped on a lower-dimensional space via PCA, Sparse PCA or the NOMINATE procedure (see Chapter 2). The Gaussian Mixture Model is built using side information on membership extracted from the raw data and the Political DNA is computed via the posterior probability estimation (GMM block, Section 4.1). To produce an interpretable visualization of the results, we represent the political positions of senators in a 2-dimensional space, building what we call a *Political Map*.

More precisely, we draw a regular polytope whose vertices represent the political groups. Since the Political DNA is a vector that naturally sums to one, we express the political orientation of each Senator as a convex combination of the positions of these vertices. Formally, given the coordinates for each vertex $\{a_\ell\}_{\ell\in\{1,\dots,n_g\}} \in \mathbb{R}^2$, the Senator s is identified in the map by a point $\gamma_s \in \mathbb{R}^2$ given by

$$\gamma_s = \sum_{\ell\in\{1,\dots,n_g\}} \pi_\ell^{(s)} a_\ell. \tag{4.8}$$

The parties are denoted by a different marker with the same legend as in Table 2.2. It is worth remarking that how to place the groups on the polytope is in principle completely arbitrary. Here, parties sharing the same orientation (left-right-independent) are placed on adjacent vertices (see Table 1.1).

We discuss the plots in the following Sections and then expose some individual Senators Political DNA (identified with the techniques described in Chapter 3) in Section 4.3.

4.2.1 Political maps

The following political maps have been generated adopting the procedure illustrated in Fig. 4.2. In Fig. 4.3 are shown the maps obtained using W-NOMINATE as a dimensionality reduction technique. In particular in Fig. 4.3.(a) is shown the map obtained considering $k = 2$ dimensions and in Fig. 4.3.(b) the map obtained considering $k = 3$ dimensions. As a mean of comparison the values of APRE and GMP (see Section 2.1.2) have been specified for each plot.

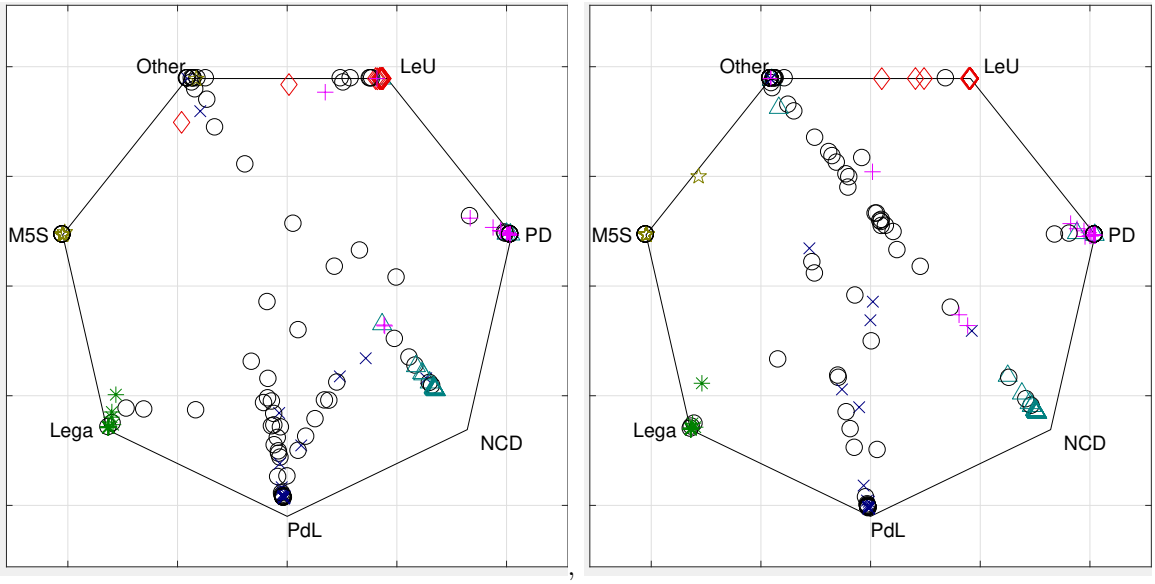
In Fig. 4.4 are shown the maps obtained using PCA as a dimensionality reduction technique. In particular in Fig. 4.4.(c) is shown the map obtained projecting the data on $k = 2$ principal components and in Fig. 4.4.(b) the map obtained projecting the data on $k = 10$ principal components. In this case the comparison is made considering the value of the expressed variance (E-Var, see Section 2.2.2) which has been specified for each plot.

Finally, in Figs. 4.5 and 4.6 are shown the map obtained using Sparse PCA as a dimensionality reduction technique. In Fig. 4.5 is shown the map obtained projecting the data on $k = 2$ sparse principal components enforcing a degree of sparsity equal to $p = 2$ (Fig. 4.5.(a)), $p = 10$ (Fig. 4.5.(b)) and $p = 50$ (Fig. 4.5.(c)). Instead in Fig. 4.6 is shown the map obtained projecting the data on $k = 10$ sparse principal components enforcing a degree of sparsity equal to $p = 2$ (Fig. 4.6.(a)), $p = 10$ (Fig. 4.6.(b)) and $p = 50$ (Fig. 4.6.(c)).

In the following sections some considerations on the plots are made, which are structurally similar to the one made in Section 2.3. This can be a good indicator of how our model correctly explain the data in a robust way.

W-NOMINATE

The political map generated starting from W-NOMINATE reduced data generally do not convey a significative amount of new information with respect to the plot shown in Section 2.3 (see Fig. 2.5). A possible reason is that, being it a MDS technique operating inside a probabilistic framework, the reduced data is already optimized to be plotted directly.



(a) $k = 2$
APRE = [0.763 0.915]
GMP = [0.841 0.924]

(b) $k = 3$
APRE = [0.764 0.914 0.959]
GMP = [0.841 0.924 0.945]

Figure 4.3: Political maps obtained with W-NOMINATE using $k = 2$ and $k = 3$ dimensions.

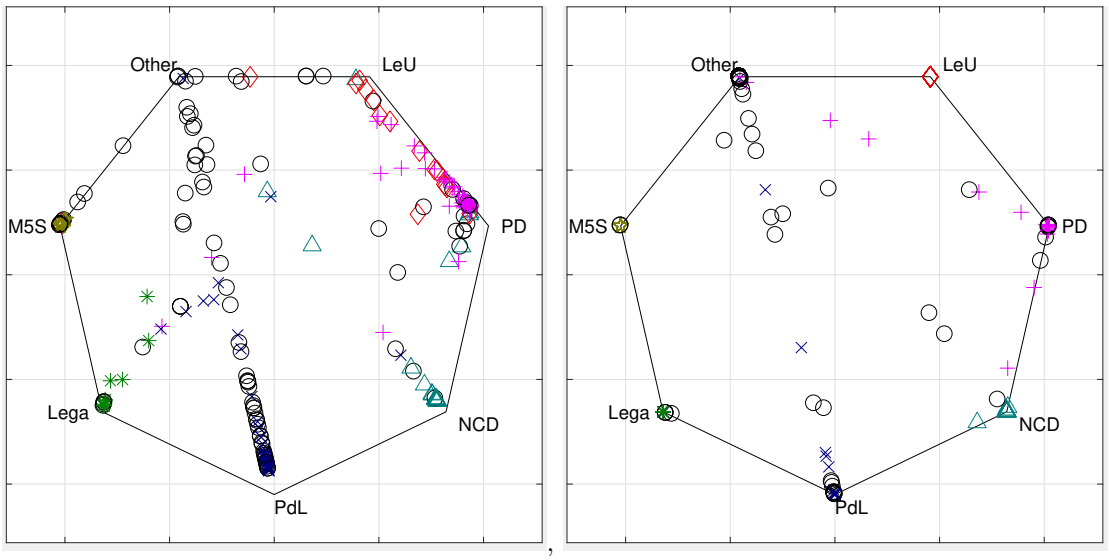
In any case, some considerations may be done:

- Both for the $k = 2$ and the $k = 3$ cases the Senators appear to be compacted towards their

nominal party, except for the Other group which, as expected, is scattered among the entire map.

- The large similarity presented by the $k = 2$ and $k = 3$ cases suggests that the extra dimension added in the latter case doesn't explain a significative amount of additional data (see also Section 2.3).

PCA



(a) PCA, $k = 2$, E-Var= 63.38%

(b) PCA, $k = 10$, E-Var= 78.79%

Figure 4.4: Political maps obtained with PCA using $k = 2$ and $k = 10$ PCs.

Regarding the maps generated from the PCA reduced data, some considerations that may hold are:

- As expected, the E-Var of the data increases as a function of the number of PCs. The small increment of about 15% with respect to the amount of PCs considered in the $k = 10$ case can be justified by the fact that the variance in the PCA components is distributed in a decreasing fashion, with the first PCs explaining the highest amount of information.

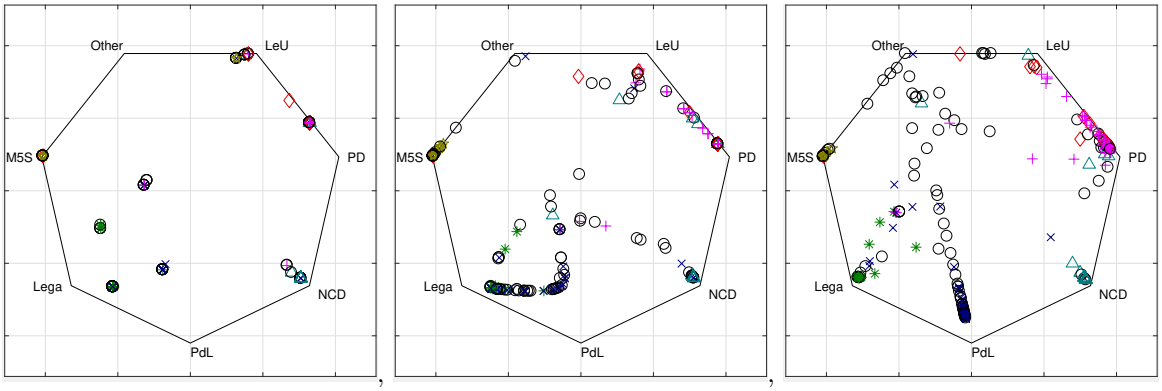
- Larger values of k lead the Senators' political positions to shift towards their nominal affiliation, hiding a possible diversification underlying the dataset.

- Also in this case, similarly to the W-NOMINATE one, the Other group is the first one to spread out when projecting the data on a low-dimensional subspace. This is to be expected

since this group is composed of a large amount of Senators of different ideologies with many of them "migrating" towards other political parties. In this case the Political DNA may help in recovering the original political identity of these Senators.

- The M5S is the most compact group i.e. is the one on which the least amount of influence is imposed by the other groups. This is coherent with the statement made in Section 2.3 regarding their internal code of conduct.

Sparse PCA

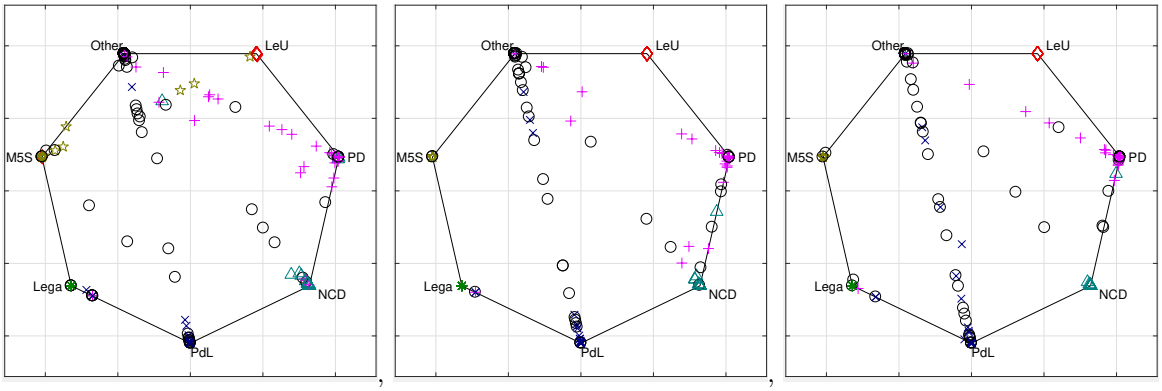


(a) Sparse PCA, $p = 2$, E-Var = 1.33%

(b) Sparse PCA, $p = 10$, E-Var = 10.01%

(c) Sparse PCA, $p = 50$, E-Var = 31.26%

Figure 4.5: Political maps obtained with Sparse PCA using $k = 2$ PCs and different level of sparsity p.



(a) Sparse PCA, $p = 2$, E-Var = 6.14%

(b) Sparse PCA, $p = 10$, E-Var = 20.73%

(c) Sparse PCA, $p = 50$, E-Var = 45.60%

Figure 4.6: Political maps obtained with Sparse PCA using $k = 10$ PCs and different level of sparsity p.

Regarding the results of the Sparse PCA analysis, many considerations may be shared with the standard PCA. Other relevant considerations are:

- For low levels of p, the Senators are less separated. This is comparable to the results obtained in Section 2.3 regarding the most important bills of the XVII Legislature (see also Table 2.3).
- As expected, the E-Var value increase proportionally with the number k of PCs and the degree of sparsity p. With respect to the PCA case, for comparable amounts of sparsity (e.g. $p = 50$) the E-Var has a more significant increase between the $k = 2$ and $k = 10$ case.
- The LeU group and PD group converge towards each other for increasing values of p. This is reasonable given that the birth of the LeU group is linked to an internal separation of the PD (see Section 2.3).
- In Fig. 4.5 we can see how a certain degree of influence is present between the M5S group and the Other group. This can be expected in accordance to a recent analysis regarding the migration of the Italian Senators to other parties [8].
- In Fig. 4.6 the PD is the first party to spread out, together with the Other group. This is reasonable since the PD was the largest political party during the XVII legislature and so the one with the most heterogeneous political composition.

Given all these considerations, we decided to adopt the model generated with Sparse PCA using $k = 2$ and $p = 50$ as the one used to illustrate the individual Political DNA of the Senators. This choice has been made because this set of parameters represents a good comprise between the degree of sparsity and how well the model is capable of explaining the dataset.

4.3 Outliers DNA

In this section we expose the individual Political DNA of some Senators whose political behavior is not consistent with the one of their nominal political group, i.e. they present a heterogeneous vector π of political influence. This Senators have been identified with the procedures explained in Chapter 3.

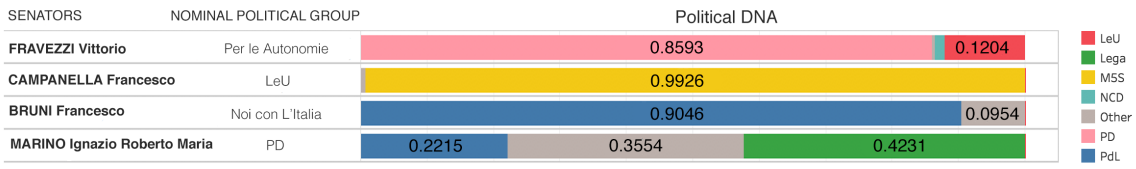


Figure 4.7: Political DNA of senators extracted via Sparse PCA with $k = 2$ and $p = 50$

In Fig. 4.7 we analyze in depth the individual Political DNA of some of outliers provided by the algorithm. The learning is performed by using the procedure described in Fig. 3.2 with $k = 2$ and $p = 50$ (see also the political map in Fig. 4.5.(c)). With these parameters the expressed variance is around 30% guaranteeing a good compromise between sparsity and data explanations. They are:

- Fravezzi Vittorio, whose nominal group is Other, appears in the first PC corresponding to PD. He has a strong influence from PD (0.855) and a small influence from LEU (0.136).
- Campanella Francesco, whose nominal group is Leu, appears in the second PC (M5S). This is coherent with the fact that he has been elected with M5S at the beginning of the legislature.
- Bruni Francesco, whose nominal group is NcI (in Other), is present in the third PC (corresponding to the PdL). He has a strong influence from PdL (0.9341) and a small influence from Other (0.0659) In fact, he was Member of PdL for almost the entire legislature.

We report also the Political DNA of Ignazio Roberto Maria Marino, whose nominal group during the entire legislature was PD, but whose Political DNA shows a prevalent component from Lega and a relevant component from PdL.

A complete list of the individual Political DNAs of the XVII Legislature is available in Appendix A. In Appendix B we also present the Political DNA for each political group. This measure has been derived by averaging the individual Political DNA of the Senators belonging to the group. This latter result is of particular interest: from it we can see how the trends that we analyzed in the previous results are also present in the analysis made by looking at the groups as a whole. The M5S expresses the maximum cohesion, as it did also on the political maps. LeU and PD share a significative amount of Political DNA, in accordance to how the LeU was formed following an internal split of the PD group. It also interesting to see how the Mixed group has in effect the most various Political DNA among all the groups. This kind of results, which well reflects the a-priori knowledge we have on the political composition of the Italian Senate, indicates that overall our model is capable of correctly interpret the data.

4.4 Testing on new data: the XVIII Legislature

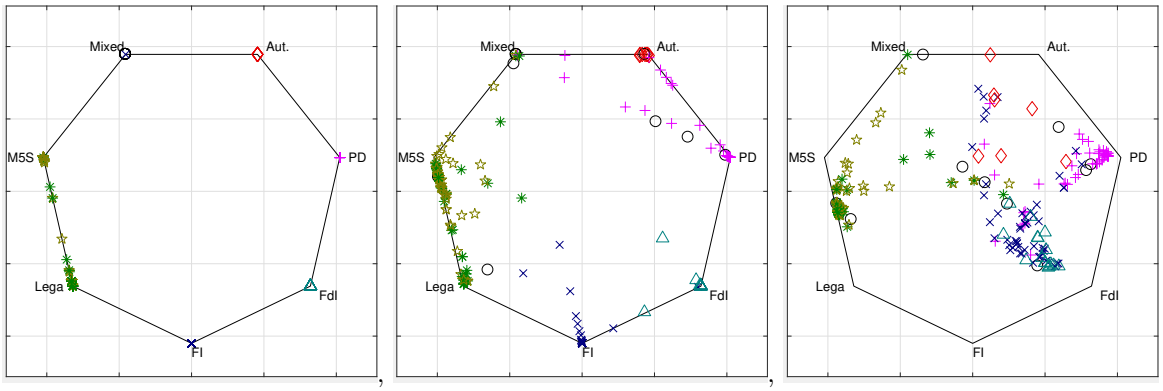
To evaluate the robustness of our model we decided to test it on the data available for the XVIII Legislature. The current Government started on the 30th of May 2018 and is still active with Giuseppe Conte as prime minister. The organization of the Italian Senate is structurally different with respect to that of the XVII Legislature. In particular, M5S and Lega are now the ruling

parties and the Government has managed to be instituted because of an alliance between these two groups. The considered groups in the analysis have changed to reflect the new composition of the Senate.

+	PD	Partito Democratico (CSX)
☆	M5S	Movimento 5 Stelle (Ind.)
✳	Lega	Lega (CDX)
×	FI	Forza Italia (CDX)
△	FdI	Fratelli d'Italia (CDX)
◇	Aut.	Per le Autonomie (CSX)
○	Mixed	Mixed group

Table 4.1: Legend of the considered political groups adopted for the XVIII Legislature

The amount of data available with respect to the XVII Legislature is relatively small. In total $n = 24$ bills were acquired to identify the Political DNA of $m = 313$ Senators (already reduced by the cleaning procedure).



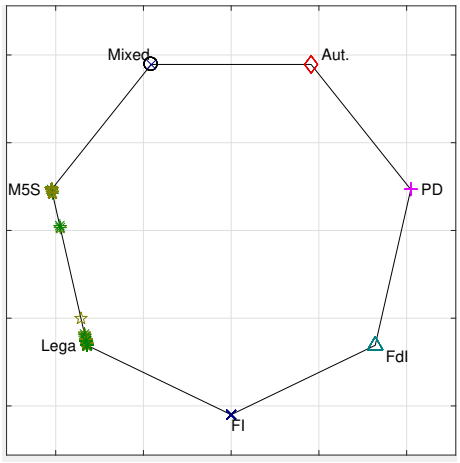
(a) PCA
$k = 20$, E-Var: 98.93%

(b) PCA,
$k = 5$, E-Var: 73.90%

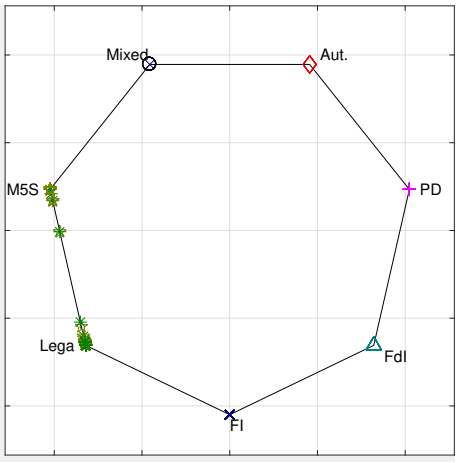
(c) PCA,
$k = 2$, E-Var: 53.04%

Figure 4.8: Political maps of the XVIII Legislature obtained with PCA using different amounts k of PCs

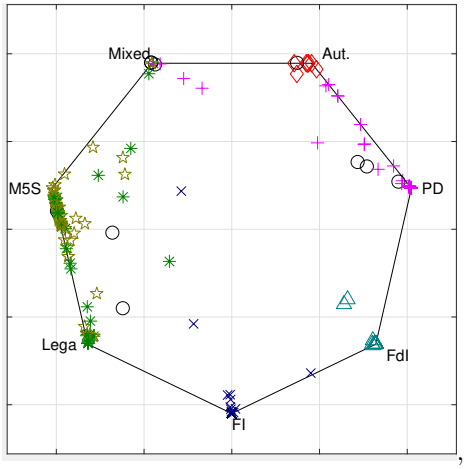
In Fig. 4.8 are shown the political maps obtained with PCA considering $k = 20$ (Fig. 4.8(a)), $k = 5$ (Fig. 4.8(b)) and $k = 2$ (Fig. 4.8(c)) PCs. The main consideration is that model is able to explain the alliance between M5S and Lega. The Senators belonging to the two currently ruling parties are heavily influenced by each other. Fig. 4.8.(b) evidently expresses the connection between the Lega and M5S groups, which almost completely meet in the middle point of Fig. 4.8.(c). It is also interesting to notice that this behavior starts to appear considering a large number of PCs (see Fig. 4.8(a)) that should instead hinder the visualization. Another notable behavior is the connection between the Senators of the PD and the Senators of the Aut., since both groups share the nominal ideology.



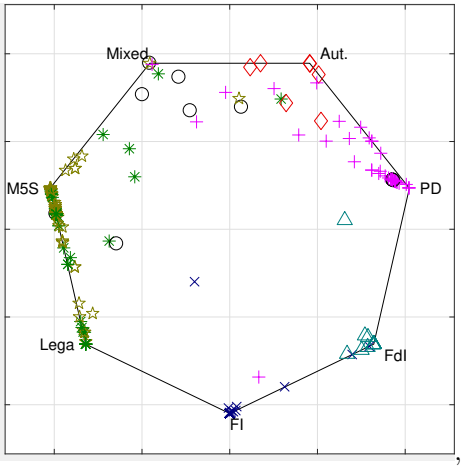
(a) Sparse PCA, $k = 20$, $p = 10$, E-Var: 88.51%



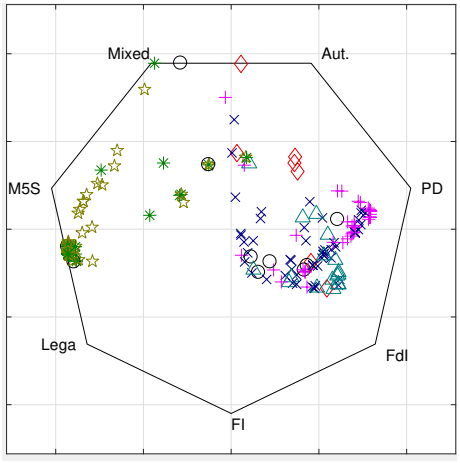
(b) Sparse PCA, $k = 20$, $p = 5$, E-Var: 70.48%



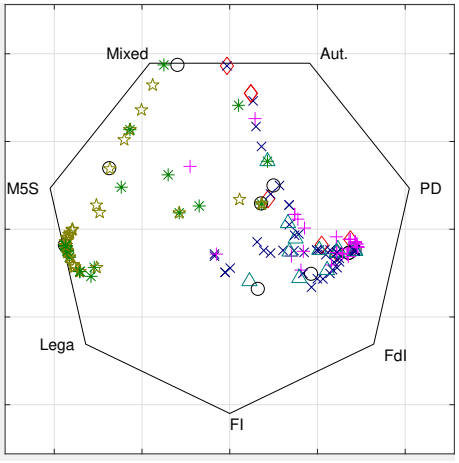
(c) Sparse PCA, $k = 5$, $p = 10$, E-Var: 64.04%



(d) Sparse PCA, $k = 5$, $p = 5$, E-Var: 44.61%



(e) Sparse PCA, $k = 2$, $p = 10$, E-Var: 43.41%



(f) Sparse PCA, $k = 2$, $p = 5$, E-Var: 26.20%

Figure 4.9: Political maps of the XVIII Legislature obtained with Sparse PCA using different amounts k of PCs and levels of sparsity p.

A similar consideration can be made for the Sparse PCA case, shown in Fig. 4.9. In this case, considering that only 24 bills were available, for each k we considered only two levels of sparsity $p = 5$ and $p = 10$. It is also interesting to notice that some of the members of the M5S group seems to migrate towards the Mixed group, mirroring the behavior already present in the XVII Legislature. Also, the PD group seems to have an affinity with the Aut. group. This is reasonable since they are the only center-left parties present on the map.

The five most important bills identified by the algorithm are exposed in Table 4.2.

Date	Description
18-07-2018	Electronic billing of gas stations Decree
26-07-2018	Bari court Decree
06-08-2018	[Milleproroghe 2018] "Thousand-extension" Decree-Law n.717. Final vote
07-08-2018	Dignity Decree
20-09-2018	[Milleproroghe 2018] "Thousand-extension" Decree-Law n.717-B. Final vote

Table 4.2: Bills of the XVIII Legislature identified by Sparse PCA ($k = 20$, $p = 5$).

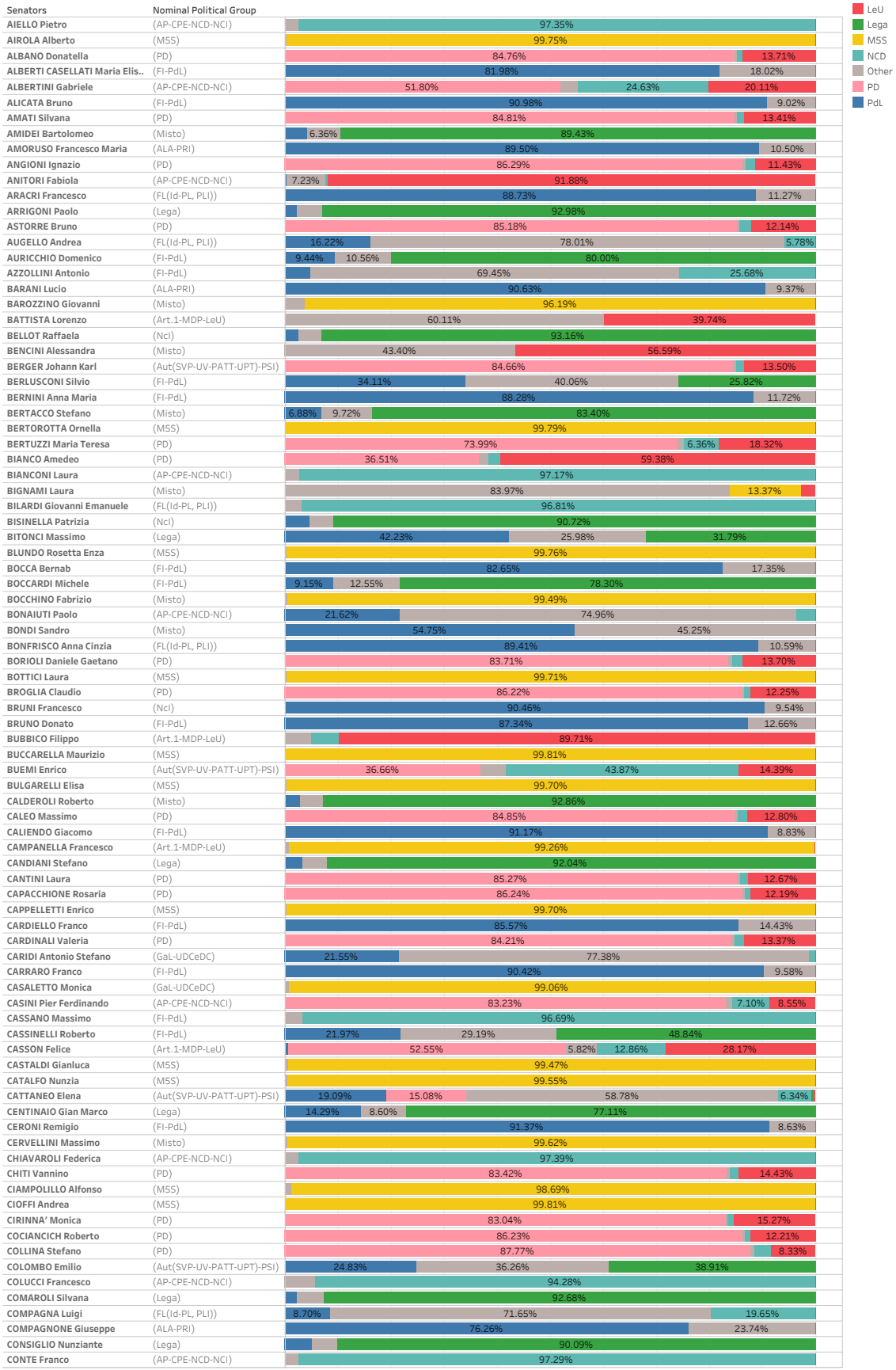
Taking into considerations these results, the model seems to be able to explain the political structure of the Italian Senate even with a limited amount of training data.

Conclusions

Over the recent years the interest in the study of social systems by means of techniques drawn from the areas of control theory and machine learning domain has risen significantly. In particular, the analysis of social influence and opinion dynamics is one of the most representative in this field of study. The building of models for the description of opinion dynamics and the identification of the complex networks regulating the social interactions between several agents participating in a collective behavior is of increasing interest in the academic community. Following this trend, in this work we presented an automated numerical technique that, based on publicly available voting data, is able to produce explanatory maps of hidden interconnections among voters nominally belonging to a given number of political or ideological groups. The thesis has been inspired also by an existing and solid research active on the U.S. Congress and the aim is to gain insight on the social composition of the Italian Senate starting from the voting data of the XVII legislature. With respect to the U.S. case, this has been proven challenging for the higher number of both political members and parties involved in the Italian case. Our method is based on a Gaussian mixture generative model that we use as a prior to compute a voter's posterior influences (what we have denominated the Political DNA), given evidence of its votes. We applied our method to a data set pertaining to the votes of 335 members of the Italian Senate on 155 bills during the XVII Legislature and tested the model on the available data of the XVIII Legislature, obtaining consistent results. These results have also been compared with the one obtained applying the NOMINATE procedure, which currently represent the state-of-the-art technique in the context of political analysis. A possible goal for a future development of this work is to infer also the underlying graph modeling the social interactions and the dynamics guiding the formation of a political opinion between the Italian Senators. While the DNA approach is here presented in a political analysis context, we believe that the kind of interpretability it offers makes it suitable to broader application endeavors, such as in the qualitative and quantitative analysis of behaviors, influence and preferences in a marketing context.

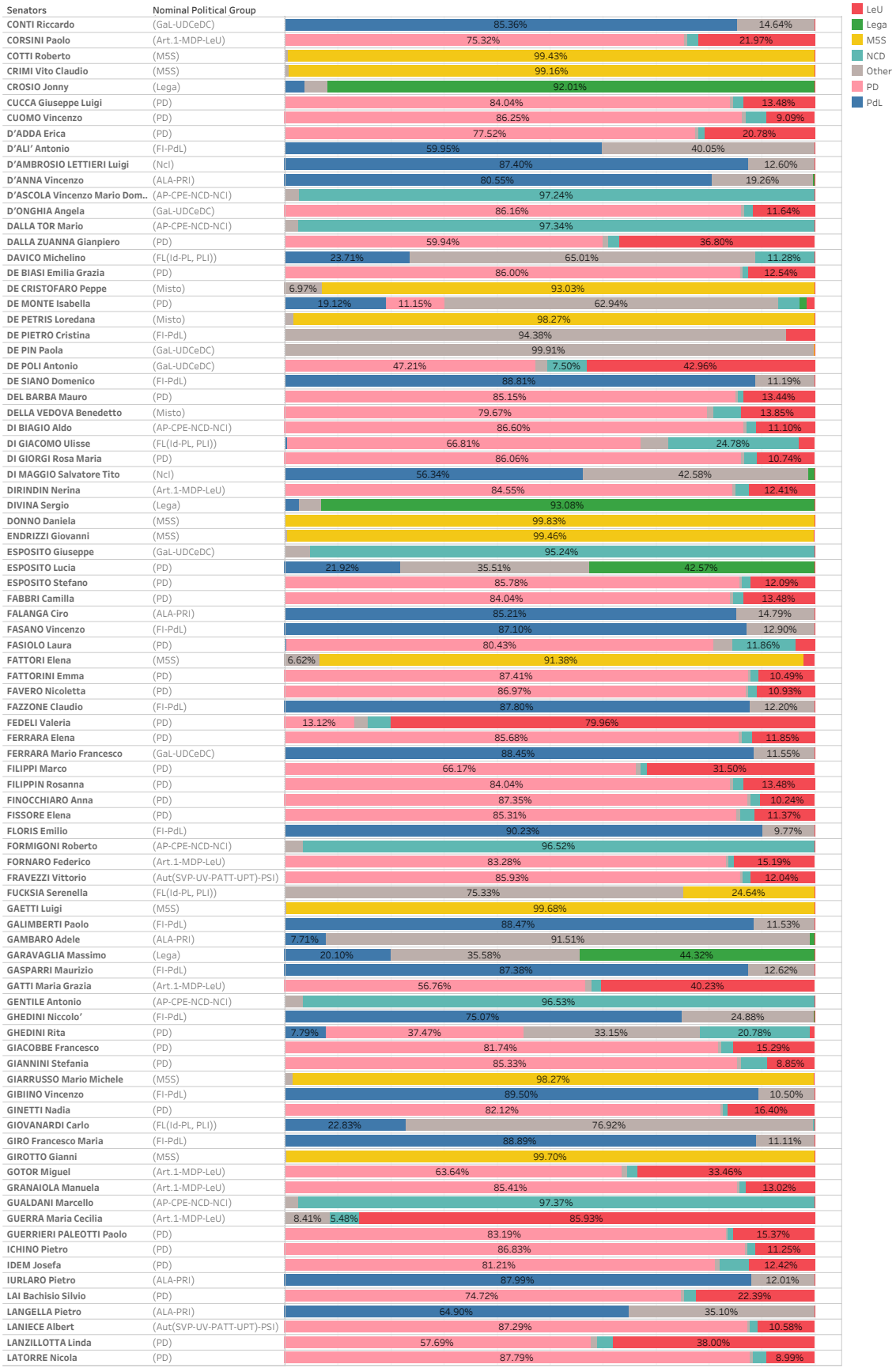
Appendix A - Political DNA of the XVII Legislature by Senators

Political DNA - Key votes - *k = 10, p = 50*



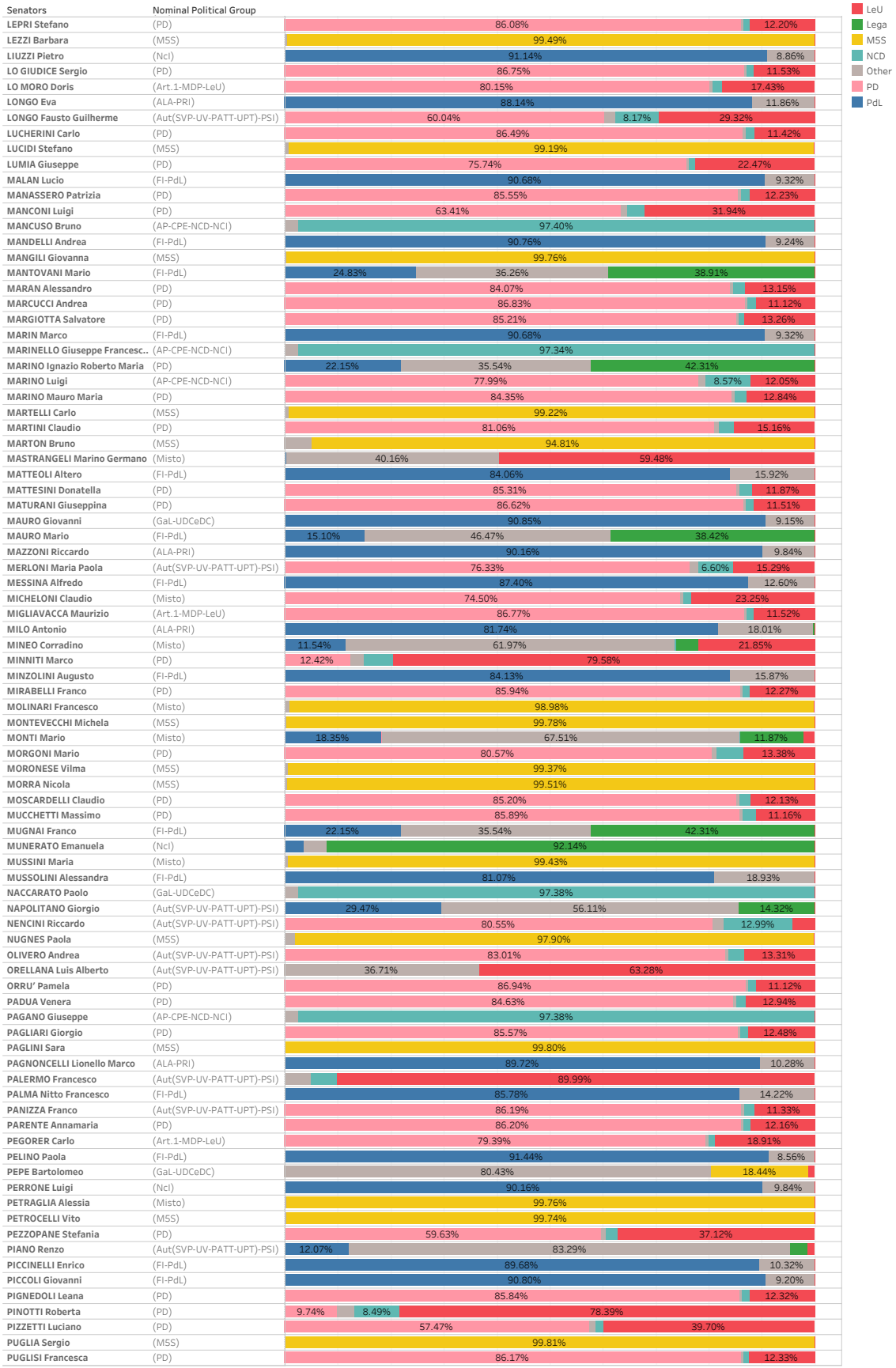
Appendix A - Political DNA of the XVII Legislature by Senators

Political DNA - Key votes - *k = 10, p = 50*



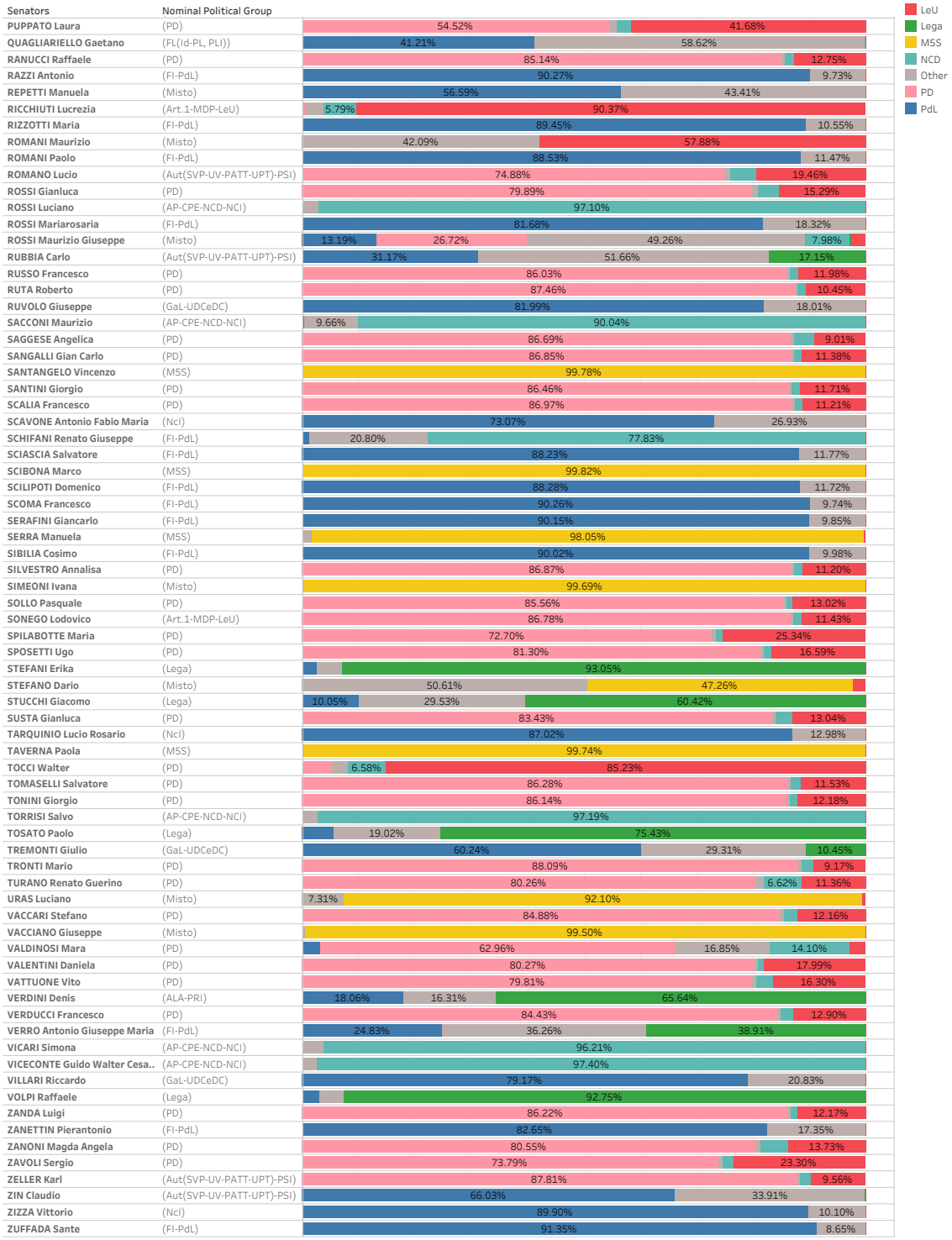
Appendix A - Political DNA of the XVII Legislature by Senators

Political DNA - Key votes - *k = 10, p = 50*



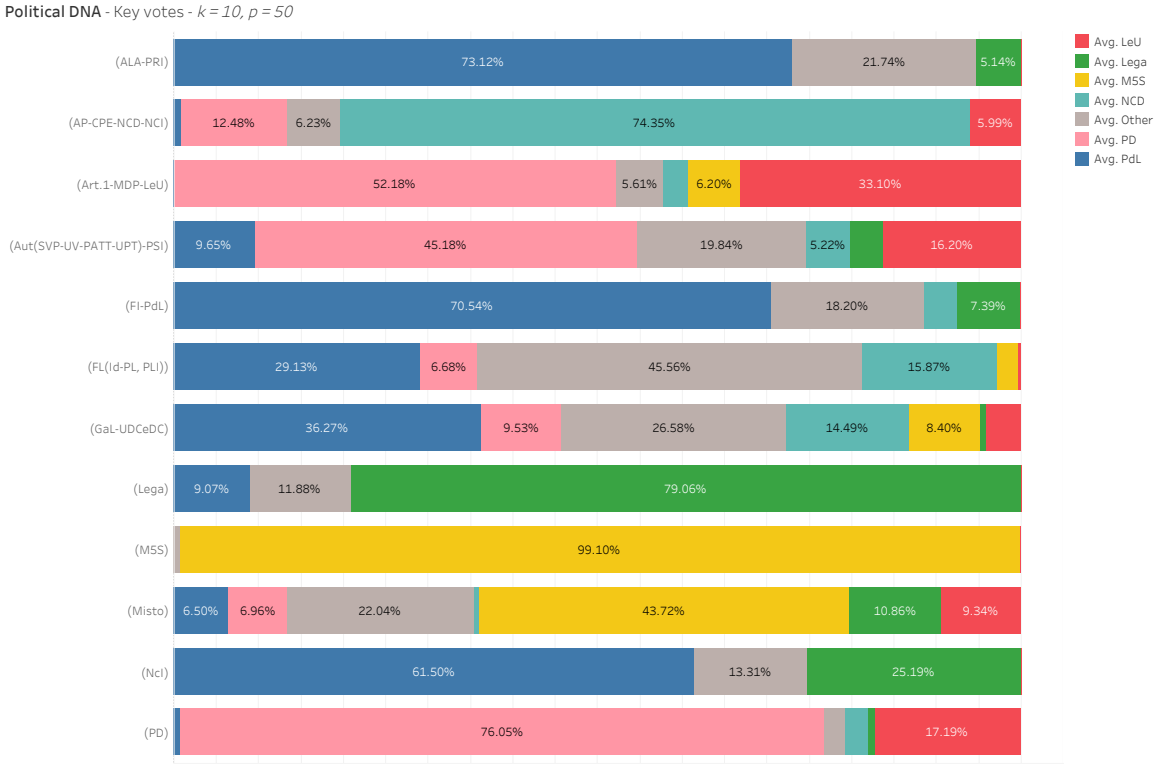
Appendix A - Political DNA of the XVII Legislature by Senators

Political DNA - Key votes - *k = 10, p = 50*



Senators	Nominal Political Group	
PUPPATO Laura	(PD)	54.52% 41.68%
QUAGLIARIELLO Gaetano	(FL(Id-PL, PLI))	41.21% 58.62%
RANUCCI Raffaele	(PD)	85.14% 12.75%
RAZZI Antonio	(FI-PdL)	90.27% 9.73%
REPETTI Manuela	(Misto)	56.59% 43.41%
RICCHIUTI Lucrezia	(Art.1-MDP-LeU)	5.79% 90.37%
RIZZOTTI Maria	(FI-PdL)	89.45% 10.55%
ROMANI Maurizio	(Misto)	42.09% 57.88%
ROMANI Paolo	(FI-PdL)	88.53% 11.47%
ROMANO Lucio	(Aut(SVP-UV-PATT-UPT)-PSI)	74.88% 19.46%
ROSSI Gianluca	(PD)	79.89% 15.29%
ROSSI Luciano	(AP-CPE-NCD-NCI)	97.10%
ROSSI Mariarosaria	(FI-PdL)	81.68% 18.32%
ROSSI Maurizio Giuseppe	(Misto)	13.19% 26.72% 49.26% 7.98%
RUBBIA Carlo	(Aut(SVP-UV-PATT-UPT)-PSI)	31.17% 51.66% 17.15%
RUSSO Francesco	(PD)	86.03% 11.98%
RUTA Roberto	(PD)	87.46% 10.45%
RUVOLO Giuseppe	(GaL-UDCeDC)	81.99% 18.01%
SACCONI Maurizio	(AP-CPE-NCD-NCI)	9.66% 90.04%
SAGGESE Angelica	(PD)	86.69% 9.01%
SANGALLI Gian Carlo	(PD)	86.85% 11.38%
SANTANGELO Vincenzo	(M5S)	99.78%
SANTINI Giorgio	(PD)	86.46% 11.71%
SCALIA Francesco	(PD)	86.97% 11.21%
SCAVONE Antonio Fabio Maria	(NcI)	73.07% 26.93%
SCHIFANI Renato Giuseppe	(FI-PdL)	20.80% 77.83%
SCIASCIA Salvatore	(FI-PdL)	88.23% 11.77%
SCIBONA Marco	(M5S)	99.82%
SCILIPOTI Domenico	(FI-PdL)	88.28% 11.72%
SCOMA Francesco	(FI-PdL)	90.26% 9.74%
SERAFINI Giancarlo	(FI-PdL)	90.15% 9.85%
SERRA Manuela	(M5S)	98.05%
SIBILIA Cosimo	(FI-PdL)	90.02% 9.98%
SILVESTRO Annalisa	(PD)	86.87% 11.20%
SIMEONI Ivana	(Misto)	99.69%
SOLLO Pasquale	(PD)	85.56% 13.02%
SONEGO Lodovico	(Art.1-MDP-LeU)	86.78% 11.43%
SPILABOTTE Maria	(PD)	72.70% 25.34%
SPOSETTI Ugo	(PD)	81.30% 16.59%
STEFANI Erika	(Lega)	93.05%
STEFANO Dario	(Misto)	50.61% 47.26%
STUCCHI Giacomo	(Lega)	10.05% 29.53% 60.42%
SUSTA Gianluca	(PD)	83.43% 13.04%
TARQUINIO Lucio Rosario	(NcI)	87.02% 12.98%
TAVERNA Paola	(M5S)	99.74%
TOCCI Walter	(PD)	6.58% 85.23%
TOMASELLI Salvatore	(PD)	86.28% 11.53%
TONINI Giorgio	(PD)	86.14% 12.18%
TORRISI Salvo	(AP-CPE-NCD-NCI)	97.19%
TOSATO Paolo	(Lega)	19.02% 75.43%
TREMONTI Giulio	(GaL-UDCeDC)	60.24% 29.31% 10.45%
TRONTI Mario	(PD)	88.09% 9.17%
TURANO Renato Guerino	(PD)	80.26% 6.62% 11.36%
URAS Luciano	(Misto)	7.31% 92.10%
VACCARI Stefano	(PD)	84.88% 12.16%
VACCIANO Giuseppe	(Misto)	99.50%
VALDINOSI Mara	(PD)	62.96% 16.85% 14.10%
VALENTINI Daniela	(PD)	80.27% 17.99%
VATTUONE Vito	(PD)	79.81% 16.30%
VERDINI Denis	(ALA-PRI)	18.06% 16.31% 65.64%
VERDUCCI Francesco	(PD)	84.43% 12.90%
VERRO Antonio Giuseppe Maria	(FI-PdL)	24.83% 36.26% 38.91%
VICARI Simona	(AP-CPE-NCD-NCI)	96.21%
VICECONTE Guido Walter Cesa..	(AP-CPE-NCD-NCI)	97.40%
VILLARI Riccardo	(GaL-UDCeDC)	79.17% 20.83%
VOLPI Raffaele	(Lega)	92.75%
ZANDA Luigi	(PD)	86.22% 12.17%
ZANETTIN Pierantonio	(FI-PdL)	82.65% 17.35%
ZANONI Magda Angela	(PD)	80.55% 13.73%
ZAVOLI Sergio	(PD)	73.79% 23.30%
ZELLER Karl	(Aut(SVP-UV-PATT-UPT)-PSI)	87.81% 9.56%
ZIN Claudio	(Aut(SVP-UV-PATT-UPT)-PSI)	66.03% 33.91%
ZIZZA Vittorio	(NcI)	89.90% 10.10%
ZUFFADA Sante	(FI-PdL)	91.35% 8.65%

Appendix B - Political DNA of the XVII Legislature averaged by groups



Bibliography

[1] Govtrack platform: https://www.govtrack.us/.

[2] Openparlamento platform: https://parlamento17.openpolis.it/.

[3] Italian Senate website: http://www.senato.it/.

[4] Scrapy framework: https://scrapy.org/.

[5] MongoDB: Open Source Document Database: https://www.mongodb.com/.

[6] "Il percorso di una legge" [Online]. Available: http://leg16.camera.it/716.

[7] "Codice etico." [Online]. Available: https://s3-eu-west-1.amazonaws.com/associazionerousseau/documenti/codice_etico_MoVimento_2017.

[8] "Cambio partito: il record di 526." [Online]. Available: https://www.corriere.it/politica/17_settembre_26.

[9] Author: Chris Hare. No changes made to the picture. [Online]. Available: https://en.wikipedia.org/wiki/NOMINATE_(scaling_method)#/media/File:House_111_X_plot.jpg

[10] Peng, Sancheng, Guojun Wang, and Dongqing Xie. "Social Influence Analysis in Social Networking Big Data: Opportunities and Challenges." *IEEE Network 31.1*: 11-17, 2017.

[11] Proskurnikov, Anton V., and Roberto Tempo. "A tutorial on modeling and analysis of dynamic social networks. Part II." *Annual Reviews in Control*, 2018.

[12] Friedkin, Noah E., and Eugene C. Johnsen. "Social positions in influence networks.", *Social Networks* 19.3: 209-222, 1997.

[13] DeGroot, Morris H. "Reaching a consensus." *Journal of the American Statistical Association* 69.345: 118-121, 1974.

[14] Acemoglu, Daron, et al. "Persistence of disagreement in social networks.", 2010.

[15] Heaney, Michael T., and Scott D. McClurg. "Social networks and American politics: Introduction to the special issue." *American Politics Research* 37.5: 727-741, 2009.

[16] Castellano, Claudio, Santo Fortunato, and Vittorio Loreto. "Statistical physics of social dynamics." *Reviews of modern physics* 81.2: 591, 2009.

[17] Gould, Roger V. "Collective action and network structure." *American Sociological Review*: 182-196, 1993.

[18] Fowler, James H. "Connecting the Congress: A study of cosponsorship networks." *Political Analysis* 14.4: 456-487, 2006.

[19] Fowler, James H. "Legislative cosponsorship networks in the US House and Senate." *Social Networks* 28.4: 454-465, 2006.

[20] Nasrabadi, Nasser M. "Pattern recognition and machine learning." *Journal of electronic imaging* 16.4: 049901, 2007.

[21] Smola, Alex, and S. V. N. Vishwanathan. "Introduction to machine learning." *Cambridge University* UK 32: 34, 2008.

[22] Mohri, Mehryar, Afshin Rostamizadeh, and Ameet Talwalkar. "Foundations of machine learning". MIT press, 2012.

[23] Tauberer, Joshua. "Observing the unobservables in the us congress" *Law Via the Internet* 487, 2012.

[24] Clinton, Joshua, Simon Jackman, and Douglas Rivers. "The statistical analysis of roll call data" *American Political Science Review* 98.2: 355-370, 2004.

[25] S. Wu, H. Wai, and A. Scaglione, "Data mining the underlying trust in the us congress," *2016 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2016 - Proceedings*. United States: Institute of Electrical and Electronics Engineers Inc. 4: pp. 1202–1206, 2017.

[26] Downs, Anthony. "An economic theory of political action in a democracy." Journal of political economy 65.2: 135-150, 1957.

[27] K. T. Poole and H. Rosenthal, "A spatial model for legislative roll call analysis," *American Journal of Political Science*, pp. 357–384, 1985.

[28] Poole, Keith T. *Spatial models of parliamentary voting*. Cambridge University Press, 2005.

[29] Lo, James. "Using W-NOMINATE in R.", 2018.

[30] W. Krzanowski, *Principles of multivariate analysis*. OUP Oxford vol. 23, 2000.

[31] I. Jolliffe, "Principal component analysis," in *International encyclopedia of statistical science*. Springer, 2011, pp. 1094–1096.

[32] J. E. Jackson, *A user's guide to principal components*. John Wiley & Sons, 2005, vol. 587.

[33] G. C. Calafiore and L. El Ghaoui, *Optimization models*. Cambridge University Press, 2014.

[34] Brigadir, Igor, *et al.* "Dimensionality Reduction and Visualisation Tools for Voting Record." *24th Irish Conference on Artificial Intelligence and Cognitive Science (AICS'16), University College Dublin, Ireland, 20-21 September 2016*. CEUR Workshop Proceedings, 2016.

[35] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Journal of computational and graphical statistics*, vol. 15, no. 2, pp. 265–286, 2006.

[36] K. Sjöstrand, L. H. Clemmensen, R. Larsen, G. Einarsson, and B. K. Ersbøll, "Spasm: A matlab toolbox for sparse statistical modeling," *Journal of Statistical Software*, vol. 84, no. 10, 2018.

[37] A. d'Aspremont, L. El Ghaoui, M. Jordan, and G. Lanckriet, "A direct formulation of sparse PCA using semidefinite programming," *SIAM Review*, vol. 49, no. 3, 2007.

[38] d'Aspremont, Alexandre, Francis Bach, and L. El Ghaoui. "Optimal solutions for sparse principal component analysis." *Journal of Machine Learning Research* 9: 1269-1294, 2008.

[39] K. T. Poole, "Nonparametric unfolding of binary choice data," *Political Analysis*, vol. 8, no. 3, pp. 211–237, 2000.

[40] Borg, Ingwer, and P. Groenen. "Modern multidimensional scaling: theory and applications" *Journal of Educational Measurement* 40.3: 277-280, 2003.

[41] McCarty, Nolan. "Measuring legislative preferences." *The Oxford Handbook of the American Congress*. 2011.

[42] Bailey, Ken. "Numerical taxonomy and cluster analysis." *Typologies and taxonomies 34*: 24, 1994.

[43] Cheeseman, Peter C., *et al.* "Bayesian Classification." *AAAI*. Vol. 88, 1988.

[44] Evers, Frederick Thomas, *et al. Fuzzy cluster analysis: methods for classification, data analysis and image recognition.* John Wiley & Sons, 1999.

[45] Shipp, Margaret A., *et al.* "Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning." *Nature medicine* 8.1: 68, 2002.

[46] Lloyd, Stuart. "Least squares quantization in PCM." *IEEE transactions on information theory* 28.2: 129-137, 1982.

[47] Steinhaus, Hugo. "Sur la division des corp materiels en parties." *Bull. Acad. Polon. Sci* 1.804: 801, 1956.

[48] Hans-Hermann, B. O. C. K. "Origins and extensions of the k-means algorithm in cluster analysis.", *Journal Electronique d'Histoire des Probabilités et de la Statistique Electronic Journal for History of Probability and Statistic* 4.2, 2008.

[49] Hinneburg, Alexander, Charu C. Aggarwal, and Daniel A. Keim. "What is the nearest neighbor in high dimensional spaces?", *26th Internat. Conference on Very Large Databases*, 2000.

[50] Blomer, Johannes, *et al.* "Theoretical analysis of the k-means algorithm - A survey.", *Algorithm Engineering*. Springer, Cham: 81-116, 2016.

[51] Garey, M. R., D. Johnson, and Hans Witsenhausen. "The complexity of the generalized Lloyd-max problem (corresp.).", *IEEE Transactions on Information Theory 28.2*: 255-256, 1982.

[52] Celebi, M. Emre, Hassan A. Kingravi, and Patricio A. Vela. "A comparative study of efficient initialization methods for the k-means clustering algorithm." *Expert systems with applications* 40.1: 200-210, 2013.

[53] Arthur David, and Sergei Vassilvitskii. "k-means++: The advantages of careful seeding." *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms.* Society for Industrial and Applied Mathematics, 2007.

[54] Rousseeuw, Peter J. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis." *Journal of computational and applied mathematics* 20: 53-65. 1987.

[55] K. Löwner. "Über monotone Matrixfunktionen.". *Math. Z.* 38: 177-216, 1934.

[56] Sun, Peng, Robert M. Freund. "Computation of minimum-volume covering ellipsoids." *Operations Research* 52.5: 690-706, 2004.

[57] Zhang, Yin, and Liyan Gao. "On numerical solution of the maximum volume ellipsoid problem." *SIAM Journal on Optimization* 14.1: 53-76, 2003.

[58] Toh, Kim-Chuan. "Primal-dual path-following algorithms for determinant maximization problems with linear matrix inequalities." *Computational Optimization and Applications* 14.3: 309-330, 1999.

[59] Shioda, Romy, and Levent Tunçel. "Clustering via minimum volume ellipsoids." *Computational Optimization and Applications* 37.3: 247-295, 2007.

[60] Lyon, Aidan. "Why are normal distributions normal?", *The British Journal for the Philosophy of Science* 65.3: 621-649, 2013.

[61] Hinton, Geoffrey. "CSC321: Introduction to Neural Networks and Machine Learning." Lecture 10, 2010.

[62] J.-M. Marin, K. Mengersen, and C. P. Robert, "Bayesian modelling and inference on mixtures of distributions," in *Bayesian Thinking*, ser. Handbook of Statistics, D. Dey and C. Rao, Eds. Elsevier, 2005, vol. 25, pp. 459 – 507.

[63] Blume, Moritz. "Expectation maximization: A gentle introduction." *Technical University of Munich Institute for Computer Science*, 2002.

[64] Lin, Xiaodong, and Yu Zhu. "Degenerate Expectation-Maximization Algorithm for Local Dimension Reduction", *Classification, Clustering, and Data Mining Applications.* Springer, Berlin, Heidelberg: 259-268, 2004.