Daniel Lemes Gribel

A clustering-based approach to detect probable outcomes of lawsuits

Rio de Janeiro

2014

Daniel Lemes Gribel

A clustering-based approach to detect probable outcomes of lawsuits

Undergraduate thesis/project presented at the Escola de Informática Aplicada of the Universidade Federal do Estado do Rio de Janeiro (UNIRIO) in order to obtain the title of Bachelor in Information Systems

Universidade Federal do Estado do Rio de Janeiro – UNIRIO

Centro de Ciências Exatas e Tecnologia

Escola de Informática Aplicada

Supervisor: Leonardo Guerreiro Azevedo

Supervisor: Maíra Athanazio de Cerqueira Gatti

Rio de Janeiro

2014

Daniel Lemes Gribel

A clustering-based approach to detect probable outcomes of lawsuits/ Daniel Lemes Gribel. – Rio de Janeiro, 2014 59 p.

Supervisor: Leonardo Guerreiro Azevedo, Maíra Athanazio de Cerqueira Gatti

Undergraduate final project – Universidade Federal do Estado do Rio de Janeiro – UNIRIO

Centro de Ciências Exatas e Tecnologia

Escola de Informática Aplicada, 2014.

1. Clustering. 2. Similarity. 3. Lawsuit. I. Leonardo Guerreiro Azevedo, Maíra Athanazio de Cerqueira Gatti. II. Universidade Federal do Estado do Rio de Janeiro. III. Escola de Informática Aplicada. IV. A clustering-based approach to detect probable outcomes of lawsuits.

CDU 02:141:005.7

Daniel Lemes Gribel

A clustering-based approach to detect probable outcomes of lawsuits

Undergraduate thesis/project presented at the Escola de Informática Aplicada of the Universidade Federal do Estado do Rio de Janeiro (UNIRIO) in order to obtain the title of Bachelor in Information Systems

Approved by the following commission in December 19th, 2014.

Leonardo Guerreiro Azevedo, D.Sc.

Maíra Athanazio de Cerqueira Gatti, D.Sc.

Adriana Cesario de Faria Alvim, D.Sc.

Sean Wolfgand Matsui Siqueira, D.Sc.

Rio de Janeiro
2014

Acknowledgements

To my parents, Rogerio and Maristerra, who have always supported me. To my supervisors, Leo and Maíra, who gave me the right direction and helped me a lot with the technical issues. To Ivar Hartmann and Gabriel Dias, for the ideas and the great help in understanding the domain of Law. To UNIRIO professors, friends and employees, for friendship, confidence and learning.

Abstract

The large amount of data coming from the numerous lawsuits in progress or already judged by the STF (Brazilian Supreme Court) consists of non-structured data, which leads to a large number of hidden or unknown information, as some relationships between lawsuits that are not explicit in the available data. These conditions also contribute to generate non-intuitive influences between variables and to increase the degree of uncertainty. However, many lawsuits can be decided based in certain patterns like: (i) the comparison to lawsuits with similar features (area of the Law, parties involved, nature of the claim, rapporteur, etc), (ii) the comparison to outcomes taken by a judge with a history of judging similar lawsuits, and (iii) the comparison to laws considered in past cases. All these parameters and some other patterns observed in past lawsuits provide a framework of non-structured data that can be transformed in useful data to predict new outcomes. This work proposes an approach to identify possible judgement outcomes that considers aspects beyond the analytical techniques. Through the use of similarity calculations and clustering mechanisms, the proposed solution was built in order to find the most similar lawsuits for a new instance that is being tested. By analysing some meta-data, it is possible to find a similar case already judged, since the amount of data provided by the judicial system are quite large. By the developing of a program that detects clusters and compiles past votes, the results shown that is possible to verify the most likely outcome and to detect its degree of uncertainty.

Key-words: clustering. similarity. lawsuit.

List of Figures

List of Tables

Contents

1 Introduction

1.1 Motivation

Modelling is one of the most effective ways to solve real problems. In many situations it is not possible to find appropriate solutions through experimentation with real objects. Build, destroy and make critical changes in real world components can be very expensive, dangerous, or just impossible, depending on the nature of the problem. Thus, an alternative consists in isolate the real world problem and build a model to represent it. This process, however, requires abstraction: it is necessary to omit the aspects considered irrelevant and consider only those that are essential for the representation of the problem. Thus, the model is often less complex than the original problem. Currently, there are several types of models. Most of the popular modelling approaches are based in formulas. The technology behind spreadsheet-based modelling is quite simple: you enter the model inputs in some cells and then view the outputs in others. The input and output values are linked by chains of formulas and – in more complex models – scripts. Various add-ons allow you to perform parameter variation, Monte Carlo, or optimization experiments (GRIGORYEV, 2012).

However, a purely analytical modelling sometimes is not enough for some problems as the one addressed in this work. According to Grigoryev (2012), for problems that present non-linear behaviour, causal and temporal dependencies, non-intuitive influences between variables and "memory" needs, it is hard to obtain the right formulas, much less put together a mental model of such system, especially if the features described above come combined with uncertainty and a large number of parameters.

This work handle the problem related to the outcome of lawsuits judged by the Brazilian Supreme Court. A large amount of non-structured data generates non-intuitive influences between variables which is also combined with some degree of uncertainty and a large number of parameters. In this context, this work proposes an approach that considers aspects beyond the analytical techniques to infer possible results through the use of pattern recognition mechanisms.

Much of the analysis involving judicial lawsuits depends on a large cognitive effort by the involved actors (lawyers, judges etc.), which demand deep domain

knowledge and ability to deal with subjective questions such as the interpretation of laws. In terms of AI (Artificial Intelligence), these conditions introduce a component of difficulty to the already existing technology and methods.

Surden (2014), the main motivation for this present work, applies machine learning techniques in the practice of law. He points how certain tasks, that are normally thought to require human skills, can sometimes be automated through the use of non-intelligent computational techniques that employ heuristics or proxies (*e.g.*, statistical correlations) capable of producing useful "intelligent" results. Then, this principle is applied to the practice of law, discussing machine-learning automation in the context of certain legal tasks currently performed by attorneys and judges including: predicting the outcomes of legal cases; finding hidden relationships in legal documents and data; electronic discovery; and, the automated organization of documents.

According to Surden (2014), some machine learning techniques are applied to law domain in order to suggest that there are a subset of legal tasks performed manually today by attorneys that can be partially automated by machine learning algorithms. However, these tasks may be partially automated, because machine learning algorithms are not used to replace crucial attorney tasks such as of determining whether certain ambiguous documents are relevant under uncertain law, or will have significant strategic value in a judicial case. Rather, in many cases, the algorithms may be able to filter out data that are likely to be irrelevant so that the attorney does not have to waste limited cognitive resources analysing them.

1.2 Problem statement

In the present work, the problem to be addressed is the large amount of data coming from the numerous lawsuits in progress or already judged by the STF (Brazilian Supreme Court). As this large amount consists of unstructured data, there are also a large number of hidden or unknown information, as some relationships between lawsuits that are not explicit in the available data. For instance, some questions that arise are: "How do we know which similar lawsuits can be a reference to a new lawsuit?"; "How do we estimate the time for taking the decisions?"; "How do we estimate a likelihood for the possible emergent results?". All these issues are important to understand the dynamics of a judicial decision and may be lost in the large volume of data generated over time. However, many lawsuits can be decided based in certain parameters, as the comparison to lawsuits with similar features (area of the Law, parties involved, nature of the claim, rapporteur, etc); the comparison

to outcomes taken by a judge with a history of judging similar lawsuits; and the comparison to laws considered in past cases. All these parameters and some other patterns observed in past lawsuits provide a framework of unstructured data that can be transformed in useful data to predict new outcomes. Moreover, this framework is also useful for finding key influencers in a particular court case.

1.3 Goals

The goal of this work is to develop an approach to suggest the possible outcomes for a given lawsuit based on modelling, similarity detection and clustering. Generally speaking, past data is analysed to detect similarities to anticipate judgements results. The solution searches for past cases that are similar and then test the behaviour of involved judges through past judgements and the degree of agreement between them. In other words, the approach is based on the analysis of past examples which should be useful enough to produce accurate results in the future, on new lawsuit instances. The proposal supports Federal government, law firms and companies on:

1. Estimating the timing of lawsuit rulings and, thus, optimize their long-term management.

2. Anticipating the likelihood of specific lawsuit developments and outcomes in order to optimize their mid/long-term management.

1.4 Text structure

This work is structured in the following chapters beyond the Introduction: Chapter 2 describes the domain and the procedure used to model the problem. Chapter 3 presents the basic concepts and terminologies used in this work. Chapter 4 describes the system built to generate the results and the main aspects of the solution. Chapter 5 presents the experiment results. Finally, Chapter 6 presents the final remarks, highlights the research contributions and suggests possibilities for future work.

2 Lawsuits modelling

In this chapter, some aspects regarding to the understanding of the domain and the rationale to model the problem is presented. The first sections provide a background on Law domain and judgements phases, while the last one focus on modelling issues, as the steps adopted to extract data and then form a dataset.

2.1 The Supreme Federal Court and its responsibilities

The Brazilian Supreme Court (STF) is an organism part of the Brazilian Judiciary System. It is responsible for the safeguarding and interpreting of the Constitution, having jurisdiction over the entire national territory. STF decides matters related to the Constitution or when there is doubt or controversy through special legal actions that work as instruments to evaluate the constitutionality of laws and matters (STF, 2011).

The Supreme Court is composed by eleven ministers nominated by the President after approval by the absolute majority of the Senate. They judge every kind of case as a last instance, *i.e.*, any case can reach the Supreme Court, but it does not go further. There is no higher court than STF in the Brazilian Justice. Among its main tasks, STF judges the following situations: the direct action of unconstitutionality of federal or state law; the declaratory action of constitutionality of law; the arguing of non fulfillment of fundamental precept deriving from the Constitution and the extradition requested by foreign state (STF, 2011).

2.2 STF judgement configuration

After several modifications in the quantity of its members, nowadays, the Supreme Federal Court is constituted by eleven Justices, who act in its Panels as well as in its Plenary. The possible judgement configuration are:

1. *Monocratic*: consists of a decision taken by a single judge. Contrasts with the collegial decision, where the court case is typically on appeal.

2. *Collegial*: consists of the results from an assessment of the case made by various judges. In this case, the rapporteur (one of them) introduces, details

and reports the case to the other judges. Then, each judge votes individually, prevailing the majority decision. A collegial decision can assume one of the following configuration:

a) *First Panel* (*Primeira Turma*, in portuguese): composed by five (5) judges.

b) *Second Panel* (*Segunda Turma*): composed by 5 judges.

c) *Plenary*: composed by all 11 judges – currently, there is an open position, so there are 10 judges.

It is important to note that each collegial configuration is independent. So, if the First Panel judged a case, there is no chance of the Second Panel to review what was decided, and vice versa. However, it is possible to have an appeal request for any judicial decision, which will be judged by the same panel.

2.3 Stages of the procedural handling

The processing period for a lawsuit can be divided into the initial stages of processing (reception and distribution), judgement, procedural notifications and final processing, as pointed out in Figure 1 (STF, 2012).

The first phase of this life cycle consists in receiving the lawsuit and classifying it in one of the following status: "Denied the follow-up": the judgment of the lawsuit does not compete to the STF; *"Sobretestado"*: the lawsuit must wait for the decision applied to another lawsuit with similar characteristics; or, "Distributed": the lawsuit will be judged by the STF and a rapporteur will be chosen to distribute the case to other ministers. In this work, only lawsuits classified as "Distributed" will be covered.

The second phase comprises the judgement itself. This is the relevant part for the present work, because it comprehends the set of decisions taken, as well as the votes of judges.

The third phase notifies the parties involved so that they can then appeal against decisions taken.

The last phase is the final milestone of the progress and represents the moment that all activities will be closed in the process.

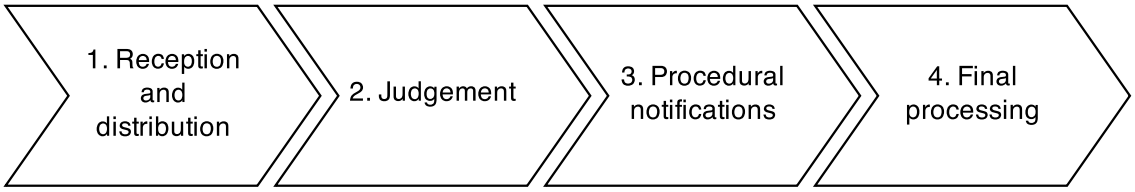

Figure 1 – The stages for a lawsuit handling

2.4 Law classes

There are several lawsuit classes in the Brazilian judicial system, as *Habeas Corpus*, *Interlocutory Appeal*, *Extraordinary Appeal*, *Direct Action of Unconstitutionality*, among others. In this work, only lawsuits belonging to the ***Appeal*** class are considered. This choice was based in two main reasons: the ***Appeal*** class corresponds to more than 50% of the lawsuits judged by the STF, which is important in terms of the heterogeneity of the data and its lawsuits have similar dynamics in their life cycles, not observed in other classes, which is important in terms of pattern detection. Thus, the choice of this class was important to constraint the boundaries of the problem and to develop a solution that is more accurate and appropriate to the reality. This decision was supported by some conversation with a professor and a student of Law School in Fundação Getúlio Vargas (FGV)[1].

2.4.1 The Appeal class

In Law, appeal is an instrument to request a change on a lawsuit being judged. According to Moacyr Amaral Santos, professor, lawyer and minister of the Supreme Court, an appeal in the judicial context is "the instrument to cause a review of a decision by the same judicial authority, or other hierarchically higher, in order to obtain their reform or modification". On the technical sense, it is the mean to review a judgement, in order to get the reform, the invalidation, the clarification or the integration of a lawsuit, and should always be provided in federal law. The Appeal class is composed by *Interlocutory Appeal*, *Extraordinary Appeal* and *Extraordinary Appeal with injury*.

[1] <http://direitorio.fgv.br/>

2.5 Mental modelling

According to FGV Law specialists, some information about a lawsuit can be extracted from metadata publicly available on internet. However, there are other relevant data from the judicial point of view that can be extracted from a deeper analysis on available documents, as the summary and the nature of the claim. In sum, the following steps were adopted in order to model a lawsuit and add it to the dataset:

1. Look for a appeal lawsuit page in the STF website[2] and identify its metadata. In a lawsuit page, it is available the lawsuit id, period (start and end date), state of origin, rapporteur, author, defendant, type (area of Law) and subjects associated to the lawsuit.

2. Identify the summary and the claim of the lawsuit. This information can be generally found in a document called *"Acórdão"*, which, for some lawsuits, is available in electronically format (PDF) in the page of the lawsuit. This document is the final decision taken collectively by the Court and represents a formal report of the decision. This document comprises a brief description of the case (with some key words addressing the summary), the understanding of the case made by the rapporteur, the citation of decisions that were references for the current decision and – maybe most important – the votes given by each judge. So, by reading this document, it may be possible to extract the summary and the claim regarding to the lawsuit. However, sometimes is not trivial to identify all of these information, once this document is extensive and is written in a difficult way to understand, due to the language of the domain.

For exemplification purpose, an already-judged lawsuit belonging to the Appeal class is shown in Annex. In these annex, the metadata[3] and the life cycle[4] of a lawsuit are shown.

Technically, a lawsuit can have many decisions before a final decision. For each decision taken by the Court, it is possible that an internal appeal is requested by a party. In this case, the Court will judge the validity of the internal appeal. That way, the conceptual diagram in Figure 2 illustrates the model, where a lawsuit

[2] <http://www.stf.jus.br/portal/cms/verTexto.asp?servico=estatistica&pagina=decisoesinicio>
[3] Annex A
[4] Annex B

is related to many decisions and a decision is related to many judges. The Enumeration `LawsuitClass` indicates the possible class assumed by a lawsuit. There are dozens of them, but only Appeal classes are considered in this work. The `DecisionType` is the modality assigned to a decision, which can be a decision taken by one judge (Monocratic) or by a collegiate (First Panel, Second Panel or Plenary). The `DecisionResult` indicates the possible results for a decision, *i.e.*, whether the judge(s) accepted or denied the appeal.

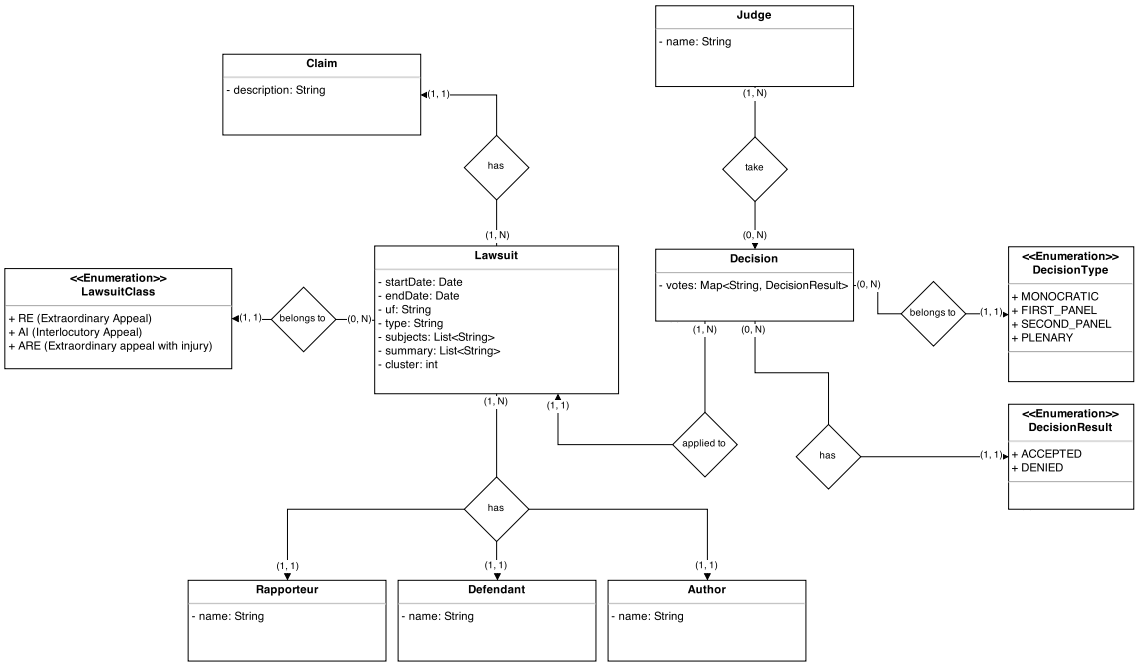

Figure 2 – A conceptual diagram illustrating how the problem was modelled

3 Main concepts and related work

The previous chapter covered some aspects regarding to the understanding of the domain and modelling details. This chapter presents the main technical concepts used in the solution proposed in this work. Concepts related to machine learning and clustering methods are presented. In addition, some related work is presented in order to compare what has been done and to understand how such solutions contributed to the development of this work.

3.1 Machine learning and statistical inferences

Many business or real-world problems that can be solved with data can be thought of as *classification* and *prediction* problems when we express them mathematically. Nowadays, many models and algorithms can be used to classify, predict or cluster. In this sense, machine learning algorithms are largely used for this purpose. In general, machine learning algorithms, such as, image recognition, speech recognition, recommendation systems, ranking and personalization of content (often the basis of data products), are not generally designed to infer the generative process (*e.g.*, to model something), but rather to predict or classify with the most accuracy. However, beyond the understanding of the machine learning algorithms operation, it is also necessary to tune them towards answering the question: "Which patterns can humans see in the data that would be automated, taking into account that complex data makes difficult to humans understand/discover the patterns?" (SCHUTT; O'NEIL, 2013).

The use of machine learning and statistical inferences are useful to test hypotheses and make estimations using sample data. These mechanisms are used to make predictions about a larger population that the sample represents.

3.2 Classification and Clustering

An important question in many areas of knowledge is how to organize observed data into meaningful structures, in order to develop taxonomies. Classification, in a broad sense of the term, can be understood as a basic human cognitive activity. Children learn very early in their lives to classify the objects in their environment and to associate the resulting classes with nouns in their language. Clas-

sification is also an important process of the practice of science since classificatory systems contain the concepts and patterns necessary for the development of theories (ALDENDERFER; BLASHFIELD, 1984).

Clustering, or cluster analysis, is the task of grouping objects of similar characteristics into categories, in such a way that objects in the same group are more similar to each other than to those in other groups (clusters). The greater the similarity (or homogeneity) within a group and the greater the difference between groups, the better or more distinct is the clustering (TAN; STEINBACH; KUMAR, 2005).

The problem of data clustering has been widely studied in the data mining and machine learning literature because of its numerous applications to summarization, learning, segmentation, and target marketing. In the absence of specific labelled information, clustering can be considered a concise model of the data which can be interpreted in the sense of either a summary or a generative model (AGGARWAL; REDDY, 2014). As pointed by Kundu (2014), clustering methods are widely used in fields like machine learning, pattern recognition, image analysis, information retrieval, and bio-informatics. Besides the term clustering, there are a number of terms with similar meanings, including automatic classification, numerical taxonomy, and typological analysis.

In other words, cluster analysis is an exploratory data analysis tool which aims at sorting different objects into groups in a way that the degree of association between two objects is maximal if they belong to the same group and minimal otherwise. Thus, cluster analysis can be used to discover structures in data without providing an explanation or interpretation. Technically, a clustering method is a multivariate statistical procedure that starts with a dataset containing information about a sample of entities and attempts to re-arrange these entities into relatively homogeneous groups (ALDENDERFER; BLASHFIELD, 1984), discovering structures in data without explaining why they exist and the possible reasons for observed co-relations.

Figure 3 exemplifies three clusters in a bi-dimensional space. The graph was obtained from the famous *Iris* dataset [1], which contains 3 classes of 50 instances each, where each class refers to a type of *iris* plant.

Most of the varied uses of cluster analysis can be subsumed under four principal goals (ALDENDERFER; BLASHFIELD, 1984):

1. Development of a typology or classification;

[1] <https://archive.ics.uci.edu/ml/datasets/Iris>

2. Investigation of useful conceptual schemes for grouping entities;
3. Hypothesis generation through data exploration;
4. Hypothesis testing, or the attempt to determine if types defined through other procedures are in fact present in a dataset.

In order to develop a typology, some structural approaches are supported by clustering method, as described in Figure 4. Basically, these approaches can be divided in Hierarchical and Partitioning.

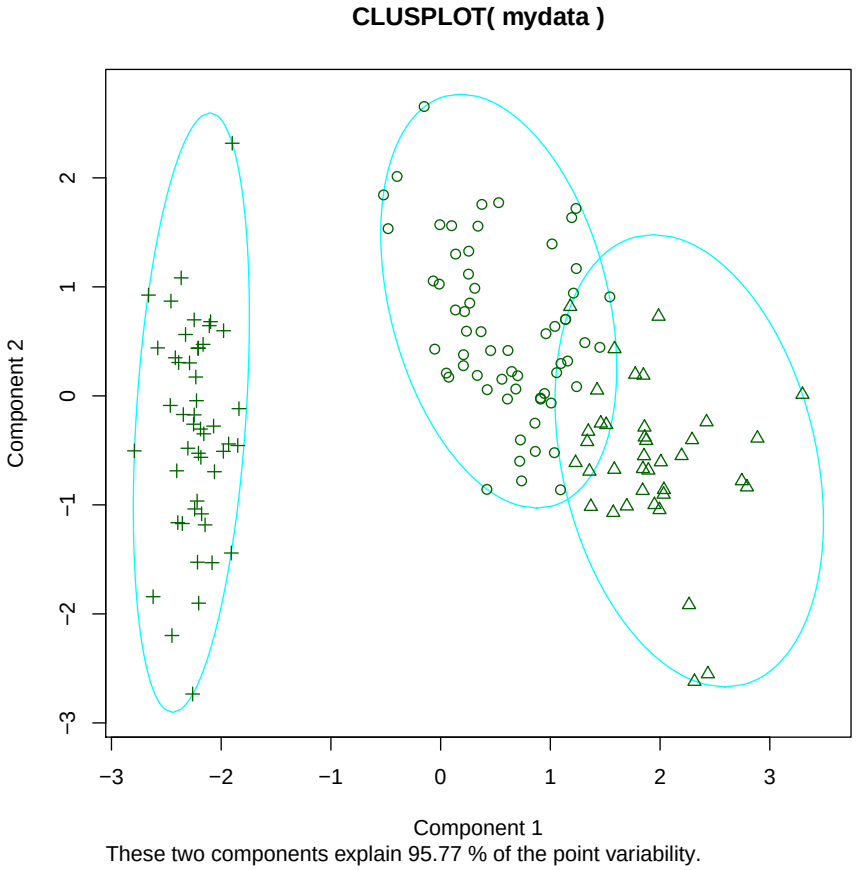

Figure 3 – A clustering example with 3 centroids in a bi-dimensional space (*Iris* dataset)

3.3 Hierarchical clustering

The hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of groups. The main reason why hierarchical clustering was chosen for experiments is that it does not require us to pre-specify the number of clusters,

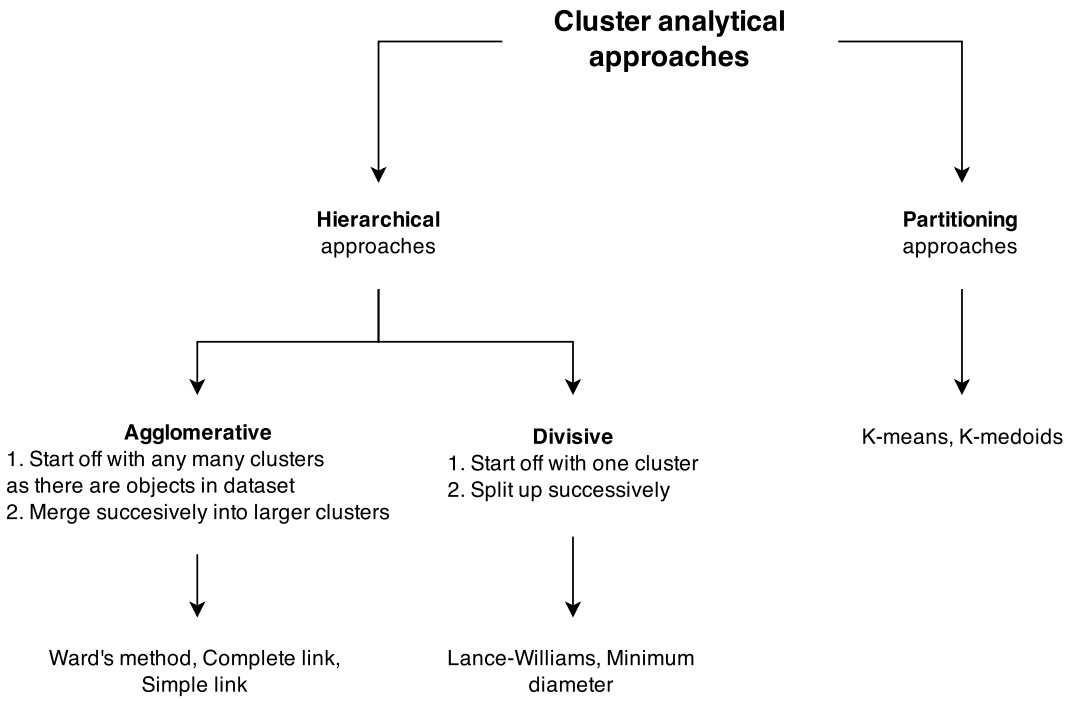

Figure 4 – Clustering analytical approaches

which is a requirement for partitioning methods. Determining the correct number of clusters is not a trivial choice. It can be ambiguous, and depends on the distribution of points in a dataset and on the desired clustering resolution. Beyond this aspect, hierarchical clustering has the distinct advantage that any valid measure of distance can be used. In fact, the observations themselves are not required. A matrix of distances is used, which is important for this case where many textual fields are considered. In addition, as described by Sander et al. (2003), hierarchical clustering is less influenced by cluster shapes and less sensitive to handle clusters with different densities.

The main disadvantage resides in its complexity, which in general is equal or higher than $O(n^2)$, which makes them too slow for large datasets (SASIREKHA; BABY, 2013).

The two main strategies for hierarchical clustering are (SASIREKHA; BABY, 2013):

(a) **Agglomerative** is a "bottom-up" approach where each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.

(b) **Divisive** is a "top-down" approach where all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

Agglomerative algorithms group data into many small clusters and few large ones, which usually makes them good at identifying small clusters but not large ones. Divisive algorithms however, have reversed characteristics, making them good at identifying large clusters in general (CHIPMAN; TIBSHIRANI, 2006). Within a domain full of uncertainty as Law, it is desirable to prioritize the quality of small clusters, once the larger ones can provide additional noise to it. Therefore, the agglomerative approach was chosen.

3.3.1 Agglomerative approach

According to Borgatti (1994), given a set of N items to be clustered, and an $N \times N$ distance (or similarity) matrix, the basic process of hierarchical agglomerative clustering is:

1. Start by assigning each item to its own cluster, so that if you have N items, you would have N clusters, each containing just one item. Let the distances (similarities) between the clusters equal the distances (similarities) between the items they contain.

2. Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one less cluster.

3. Compute distances (similarities) between the new cluster and each of the old clusters.

4. Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.

At the moment of merging clusters, it is necessary that some agglomerative method indicate which is the "shortest distance" criteria to join such clusters. The definition of this "shortest distance" is what differentiates the many agglomerative clustering methods. According to R documentation[2], different agglomerative methods are provided for hierarchical clustering. In this work, the agglomeration method chosen for the hierarchical clustering was the Ward's algorithm, which is presented in the next section.

[2] <https://stat.ethz.ch/R-manual/R-patched/library/stats/html/hclust.html>

3.3.1.1 The Ward's algorithm

Ward's method is a special case of objective function approach originally presented by Joe H. Ward Jr. He suggested a general agglomerative hierarchical clustering procedure, where the criterion for choosing the pair of clusters to merge at each step is based on the minimum of some inter-cluster distance measure (WARD, 1963).

In Ward's minimum variance criterion, which is a particularization of the Ward general method, the objective function is to minimize the total within-cluster variance. That is, at each step of the algorithm, the pair of clusters with minimum between-cluster distance are merged. To implement this method, at each step it is necessary to find the pair of clusters that leads to minimum increase in total within-cluster variance after merging (WARD, 1963). As a general result, Ward's minimum variance method leads to compact, spherical clusters.

At the first step, all clusters are singletons, which means that each cluster contains only one object. To apply a recursive algorithm under this objective function, the initial distance between individual objects must be proportional to squared Euclidean distance. The initial cluster distances in Ward's minimum variance method are therefore defined to be the squared Euclidean distance between points:

$$d_{ij} = d(\{X_i\}, \{X_j\}) = \|X_i - X_j\|^2 \tag{3.1}$$

where X_i and X_j refers respectively to points (single clusters) i and j.

3.3.1.2 Lance–Williams algorithms

Ward's minimum variance method can be defined and implemented recursively by a Lance–Williams algorithm. The Lance–Williams algorithms are a huge family of agglomerative hierarchical clustering algorithms which are represented by a recursive formula for updating cluster distances at each step. At each step, it is necessary to optimize the objective function (find the optimal pair of clusters to merge). The recursive formula simplifies finding this pair.

Suppose that clusters C_i and C_j are next to be merged. At this point, all of the current pairwise cluster distances are known. The recursive formula gives the updated cluster distances following the pending merge of clusters C_i and C_j. Let:

- d_{ij}, d_{ik}, and d_{jk} be the pairwise distances between clusters C_i, C_j, and C_k, respectively,

- $d_{(ij)k}$ be the distance between the new cluster $C_i \cup C_j$ and C_k.

In this way, an algorithm belongs to the Lance-Williams family if the updated cluster distance $d_{(ij)k}$ can be computed recursively by:

$$d_{(ij)k} = a_i d_{ik} + a_j d_{jk} + \beta d_{ij} + \gamma |d_{ik} - d_{jk}|, \tag{3.2}$$

where a_i, a_j, β, and γ are parameters, which may depend on cluster sizes, that together with the cluster distance function d_{ij} determine the clustering algorithm. Several standard clustering algorithms such as single linkage, complete linkage, and group average method have a recursive formula of the above type.

Ward's minimum variance method can be implemented by the Lance–Williams formula. For disjoint clusters C_i, C_j, and C_k with sizes n_i, n_j, and n_k respectively:

$$d(C_i \cup C_j, C_k) = \frac{n_i + n_k}{n_i + n_j + n_k} d(C_i, C_k) + \frac{n_j + n_k}{n_i + n_j + n_k} d(C_j, C_k) - \frac{n_k}{n_i + n_j + n_k} d(C_i, C_j). \tag{3.3}$$

Thus, Ward's method can be implemented as a Lance–Williams algorithm with the following values for the estimated parameters:

$$a_i = \frac{n_i+n_k}{n_i+n_j+n_k}, \qquad a_j = \frac{n_j+n_k}{n_i+n_j+n_k}, \qquad \beta = \frac{-n_k}{n_i+n_j+n_k}, \qquad \gamma = 0.$$

3.3.1.3 Single and Complete-linkage algorithm

The class of agglomerative algorithms is perhaps the most popular class of hierarchical algorithms (ACKERMAN; BEN-DAVID, 2011). As Ward's method described previously, Single-linkage and Complete-linkage also belong to the agglomerative approach and are extensively used in the literature for hierarchical clustering purposes. In the next paragraph, Single and Complete-linkage algorithms will be briefly presented, in order to compare their performances with Ward's algorithm in results section (5.5). These two algorithms were chosen because they fit well for some test instances and have a large coverage in literature.

Firstly, it is important to note that both algorithms can also be implemented as a Lance-Williams algorithm. In Single-linkage clustering, the objective function is defined by those two elements (one in each cluster) that are closest to each other. The shortest of these links causes the fusion of the two clusters whose elements are involved. The method is also known as nearest neighbour clustering. In Complete-linkage clustering, the objective function is defined by those two elements (one in

each cluster) that are farthest away from each other. The shortest of these links causes the fusion of the two clusters whose elements are involved. The method is also known as farthest neighbour clustering.

4 Automated inferring of lawsuits outcomes

As described in previous chapters, this work deals with following problem: when a new lawsuit come, it is hard to figure out what is the probable outcome and how much time the judgement will take until the final decision. The proposed solution analyses past data, calculates similarities and applies clustering techniques to then achieve lawsuit outcomes prediction. Considering a dataset already modelled, the process consists in using certain parameters – as the comparison to lawsuits with similar features (area of the Law, parties involved, nature of the claim, rapporteur, etc) and the comparison to past decisions – to predict the likely outcome that could arise for a new lawsuit. Also, the votes of the judges and the degree of agreement between them are taken into account.

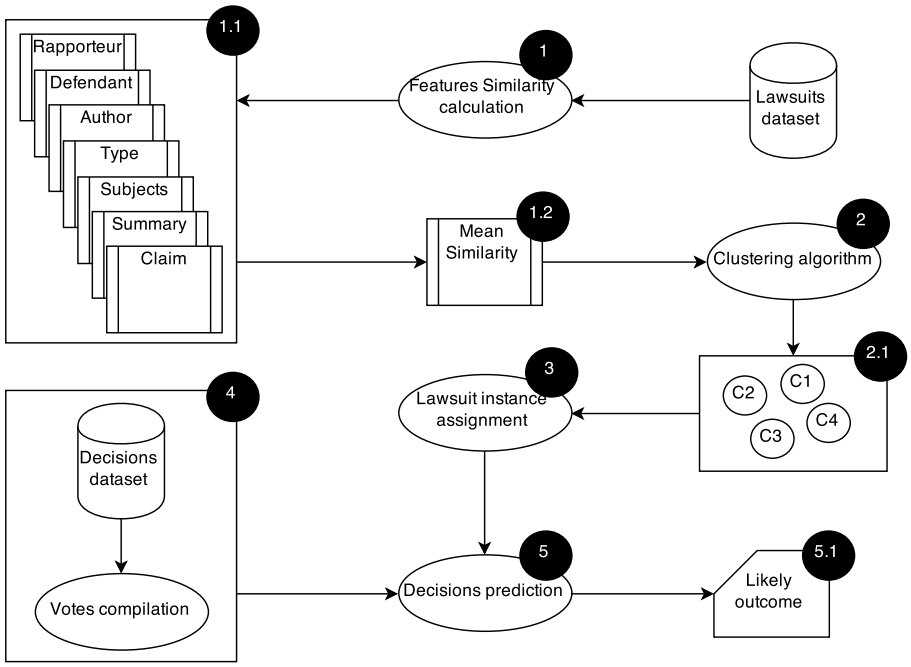

Figure 5 – The workflow proposed for lawsuits outcomes

The general workflow of the proposed solution is explained by figure 5, where each step of the process (numbers inside black bubbles) will be detailed bellow:

1. Features Similarity calculation: from the modelled dataset, calculate the simi-

larities between lawsuits. At this step, each pair of lawsuit receives a similarity coefficient regarding to a property. The output of this phase is one adjacency matrix for each property (rapporteur, defendant, author, type, subjects, summary and claim), conform item 1.1. Then, a mean matrix is obtained from each property matrix, giving final similarity result, as shown in item 1.2.

2. Clustering algorithm: from the similarities observed, run the hierarchical clustering algorithm for the lawsuits dataset. In this phase, lawsuits are assigned to groups, according to their similarities. The output of this phase are the lawsuits classified in clusters, as pointed by item 2.1.

3. Lawsuit instance assignment: from the detected clusters, calculate the similarities between the new lawsuit instance and the other lawsuits already classified. In this step, the new instance is assigned to the most similar cluster.

4. Decisions compilation: considering a list of judges that will decide the lawsuit as an input of the system, collect their past votes observed in the cluster. This allows an understanding of the likely outcome to the lawsuit being tested. In this step, the decisions dataset is used.

5. Decisions prediction: finally, past votes are computed and the degree of agreement between judges is calculated. The output of this step, which is the output of the whole process, is the likely outcome (item 5.1) – a number between 0 and 1, indicating the probable decision for the lawsuit being tested.

4.1 Similarity calculation

In this step of the process, the goal is to find the similarities between the lawsuits of the dataset. For this purpose, each property is compared taking into account that all lawsuits are compared in pairs. For binary properties (as the rapporteur name, defendant, author, type and claim), 0 or 1 is assigned for each pair of lawsuits. The logic here is to generate a distance matrix for each property (feature) of lawsuits. In this way, if two lawsuits have the same binary property, the value 0 is assigned to the pair. If the property is not the same then the value 1 is assigned to the pair. Thus, the point here is: the closer to zero the pair of lawsuits is, then more similar the lawsuits are. The closer to 1, more different they are, considering the property being compared.

When analysing non-binary properties, that is, properties that are composed by a list of items, as subjects and the summary, then the Jaccard index (JACCARD,

1912) is applied to the pair of lawsuits. The Jaccard index is a statistic used for comparing the similarity and diversity of sample sets. This coefficient measures similarity between finite sets (in this case two sets are taken for each comparison), and is defined as the size of the properties intersection divided by the size of the properties union:

$$j_x(a,b) = \frac{|a_x \bigcap b_x|}{|a_x \bigcup b_x|} \tag{4.1}$$

where a and b are two lawsuits being compared and x is the property. Thus, the Jaccard coefficient generates a number between 0 and 1. To meet the normalization adopted by the distance matrix, we consider the value $1-j(a,b)$ for each pair (a,b). So, the closer to zero the coefficient is, then more similar the lawsuits are.

4.1.1 Mean similarity matrix

After all similarities matrices are computed by properties, it is necessary to calculate a resultant matrix, which is obtained through a weighted average of each property similarity matrix. This resultant matrix is required for the clustering algorithm, which expects a single similarity object as input. However, it is essential to distinguish the weights for each property. For instance, to calculate the final similarity between lawsuits, it is important to consider that the summary is more relevant than the rapporteur, as it concerns to the nature of the lawsuit, while a rapporteur participates in the judgement of many different lawsuits. As the goal in this step is to find lawsuits of the same nature and with similar features, the weight assigned to summary is greater then the weight assigned to rapporteur. The table 1 shows the arbitrary weights chosen for each property, based on a mental modelling that considers the importance of each one of them for clustering purposes.

Property	Weight
Rapporteur	1
Defendant	1
Author	1
Type	2
Subjects	3
Summary	4
Claim	3

Table 1 – Weights arbitrarily assigned to each lawsuit property

So, to generate a mean similarity matrix the following mean calculation is performed:

$$m(a,b) = \frac{\sum_{x=0}^{n} j_x(a,b) \cdot w_x}{\sum_{x=0}^{n} w_x} \tag{4.2}$$

where a and b are two lawsuits being compared and x is one of the properties. Note that x varies from 0 to n, where n represents the number of properties taken into account, which in this case is 7. The $j_x(a,b)$ is the Jaccard index computed for lawsuits a and b, considering the x property, as described in equation 4.1. The w_x is the arbitrary weight assigned to the property x in question, as defined in table 1.

4.2 Lawsuits clustering

From the mean similarity matrix computed in the previous step, it is needed to generate a distance matrix, in order to have a valid object for the hierarchical clustering operation, which is the next step to be executed.

A distance refers to a positive, symmetric mapping of a pair of observation vectors onto a positive real which determines the degree of similarity (distance) between pair values observed. To achieve the proposed hierarchical clustering results, it was developed the algorithm 1, which will be explained in the next paragraphs (see also the R script [1] which runs the algorithm).

Algorithm 1 Assign lawsuits to clusters

Require: *similarityMatrix*, *agglomerativeMethod* {Similarity coefficients between all lawsuits, Agglomerative algorithm to guide hierarchical clustering}
Ensure: Lawsuits assigned to clusters

```
distanceMatrix ← calculate distance matrix from similarityMatrix
clusters ← run hierarchical clustering on distanceMatrix with
    agglomerativeMethod
S ← size of similarityMatrix(number of distinct entries)
for i ← 2; i ≤ √S do
    cuts ← apply tree cut on clusters into i groups
    print cuts to standard output
end for
```

Although the hierarchical clustering does not require a pre-defined number of clusters, as it groups all items until a single cluster of size N, it is important to

[1] Appendix A

determine the "cut tree", a mechanism to cut a tree resulting from the hierarchical clustering by specifying the desired number(s) of groups. This mechanism is desirable because the process continues beyond the clustering algorithm, with the step that calculates the similarity between a new lawsuit and the clustered lawsuits. Thus, to avoid further unnecessary comparisons (as comparing the instance to clusters of size 1 or N), the result of the hierarchical clustering here implemented is guided by the "cut tree" specification. So, in algorithm 1 is defined a maximum number of clusters that will be evaluated posteriorly. It varies from 2 to $\sqrt{S}$, where S is the number of entries in the lawsuits dataset. The cut tree is then performed inside the *for*.

Determining the number of clusters – called from now by k – in a dataset is a common problem in data clustering, and is a distinct issue from the process of actually solving the clustering problem (CAO et al., 2012). This reveals an important issue to be addressed as future work: it may be important to define a more efficient clustering method than the hierarchical clustering. That is, the hierarchical clustering could be a reasonable escape from the solution quality point of view, but it is bad in terms of performance, once its complexity is, in the general case, $O(n^3)$. Instead of generating intermediate solutions in the search space by calculating the number of clusters ranging from N to 1, which certainly demands too much processing time on a large dataset, it could be defined a criterion for determining the number of clusters according to the dataset size and other characteristics, as the variance of the points in the dataset.

However, the correct choice of the number of clusters is often ambiguous (CAO et al., 2012), with interpretations depending on the distribution of points in a dataset and on the desired clustering resolution of the user. By increasing k without penalty will always reduce the amount of error in the resulting clustering, to the extreme case of zero error if each data point is considered its own cluster (i.e., when k equals the number of data points, N) (VIRVOU; ALEPIS; TROUSSAS, 2012), which does not seem reasonable. On the other hand, a very low value for k leads to very general and imprecise clusters. Another possibility is to define a certain search space (which would be a sub-space of the hierarchical space) and then develop optimization heuristics to determine a stopping point when the algorithm finds a good solution. A stopping point, in this case, could be when the algorithm finds a cluster that is similar enough to the instance been tested and has difficulties to improve this best rate found.

Returning to algorithm 1, the hierarchical clustering uses a set of similarities, which in this case was generated in the previous step, the similarity calculation. Then, this matrix is transformed in a distance matrix. The agglomeration method

is passed according to the agglomeration algorithm adopted. For these experiments, Ward's, Single-linkage and Complete-linkage were used (to remember details about these methods, see subsection 3.3.1). The final output of the algorithm is a tuple <lawsuit id, assigned cluster> for each cut (total number of clusters).

4.3 Lawsuit instance assigning

After all lawsuits of the dataset were properly allocated to a cluster, it is time to check in which cluster a lawsuit test instance should be allocated. For this purpose, it is sufficient to use a method already been implemented: the similarity calculation, described in section 4.1.

The instance is compared to each classified lawsuit and then a coefficient from 0 to 1 is generated, conform the similarity average based on the properties weights. Finally, with the similarity coefficients and the information of the clusters, an average is applied by cluster. Then, the new instance is assigned to the most similar cluster – that one with the smallest average of coefficients.

4.4 Votes compilation

The last step of the proposed solution consists in compiling the judges votes. For this purpose, two aspects are taken into account:

1. The decisions taken by each judge for each lawsuit in the cluster, *i.e.*, if the judge denied or accepted the judicial review.
2. The degree of agreement between the judges, *i.e.*, if the judges agree in the decisions taken. This is important to detect some divergent behaviour in the judgement.

It is important to be said that only the votes of a set of judges is evaluated. This set of judges is part of the system input, and represents the judges that would participate in the judgement of the instance being tested. In the STF case, this set can be composed by a single judge (in case of a monocratic decision), by five judges (in case of First Panel or Second Panel) or by eleven judges (in case of Plenary).

To compute the decisions (votes) taken by each judge (item 1 above), the degree of acceptance of each judge in the cluster is collected. That is, it corresponds to the ratio $\frac{A_i}{A_i+D_i}$, where A_i is the number of lawsuits in the cluster i favourably judged (accepted) by the judge and D_i is the number of lawsuits unfavourably judged

(denied) in the same cluster. Thus, 0 means that the judge denied all lawsuits in the cluster and 1 means that all lawsuits were accepted by him/her. So, if the judge judged 15 cases belonging to the cluster in question and accepted 6 of them, his/her votes coefficient is $\frac{6}{15} = 0.4$. If the judge did not participate in any lawsuit belonging to the cluster, then "n/a" is the result. Algorithm 2 explains this procedure (see also the Java function [2] which runs the algorithm).

To compute the degree of agreement between the judges (item 2 above), the following rationale is adopted: for each lawsuit belonging to the cluster where the judge voted and another judge composing the input also voted, it is computed if they agreed or disagreed (1 or 0 is assigned, respectively). Then, the average is calculated for each judge. If the judge did not participate in any lawsuit belonging to the cluster, then "n/a" is the result. Algorithm 3 explains this procedure (see also the Java function [3] which runs the algorithm).

Algorithm 2 Judges votes on cluster

Require: $judgesList, c$ {the judges that would participate in the judgement, the evaluated cluster}
Ensure: Mean votes given by each judge in the cluster

```
1:  for all judge j in judgesList do
2:      numDecisionsParticipated = 0
3:      numDecisionsAccepted = 0
4:      for all decision d taken by judge j on cluster c do
5:          numDecisionsParticipated++
6:          if d == ACCEPTED then
7:              numDecisionsAccepted++
8:          end if
9:      end for
10:     if numDecisionsParticipated == 0 then
11:         return n/a
12:     else
13:         return numDecisionsAccepted / numDecisionsParticipated
14:     end if
15: end for
```

[2] Appendix B
[3] Appendix C

Algorithm 3 Judges agreement on cluster

Require: $judgesList, c$ {the list of judges that would participate in the judgement, the evaluated cluster}

Ensure: The agreement degree between judges votes in the cluster

1: **for all** judge j in $judgesList$ **do**
2: $numCommonVotes = 0$
3: $numAgreements = 0$
4: **for all** lawsuit l within cluster c **do**
5: **for all** decision d_j taken by j in lawsuit l **do**
6: **for all** decision d not taken by j in lawsuit l **do**
7: **if** $d_j == d$ **then**
8: $numAgreements++$
9: **end if**
10: $numCommonVotes++$
11: **end for**
12: **end for**
13: **end for**
14: **if** $numCommonVotes == 0$ **then**
15: return n/a
16: **else**
17: return $\frac{numAgreements}{numCommonVotes}$
18: **end if**
19: **end for**

5 Results

In this chapter, the key results will be presented, indicating the main results for outcomes and the interpretation we can extract from them. In order to give practical effect to the workflow described in Chapter 4, a solution composed by a Java program and a R script for clustering was developed.

5.1 Datasets

First of all, let's describe the datasets used in experiments. There are two datasets: one composing the set of extracted lawsuits and another composing the decisions taken in each lawsuit. It is important to remember that each lawsuit can have one or more decisions, once we are treating only lawsuits already judged, *i.e.*, lawsuits with a final decision. For the last experiments performed, datasets `lawsuit_16.csv` and `decision_16.csv` – composed by 16 lawsuits and 24 decisions, respectively – were used. Actually, these datasets represent the mental modelling applied to lawsuits and decisions available on internet, which corresponds to the identification of metadata and the extraction of *"Acórdão"* useful information. Both datasets, lawsuits[1] and decisions[2], are available as annex in the final part of this work.

Each lawsuit composing the lawsuits dataset provides the following information: lawsuit id, start/end date of lawsuit, state of origin, rapporteur, defendant, author, type (area of Law), subjects, summary and claim.

Each decision composing the decisions dataset provides the following information: associated lawsuit id, decision id, type of decision (Monocratic, First Panel, Second Panel or Plenary), date, votes tuple <judge name, vote (accepted or denied)> and resultant decision (accepted or denied).

All lawsuits and decisions of the datasets were extracted directly from the STF website[3].

[1] Annex C
[2] Annex D
[3] <http://www.stf.jus.br/portal/cms/verTexto.asp?servico=estatistica&pagina=decisoesinicio>

5.2 Similarity analysis

In order to achieve the similarities between lawsuits, each property is compared taking into account that all lawsuits are compared in pairs. For binary properties, 0 (same property) or 1 (different property) is assigned. For non-binary properties, Jaccard index is used. Thus, an adjacency matrix representing similarities is generated for each property. Figures 6 and 7 show the similarity matrices for a binary property (rapporteur) and a non-binary property (summary), respectively. Each row and each column represents a lawsuit of the dataset, which in both cases is composed by 16 lawsuits. For obvious reasons, these similarities matrices are symmetric and their main diagonal are zero.

After all similarities matrices are computed by properties, it is necessary to calculate a resultant matrix, which is obtained through a weighted average of each property similarity matrix. Figure 8 shows the final result for the similarities compilation between lawsuits. This mean similarity matrix is an important result to determine how similar is a lawsuit to another. Beyond this, it is the input necessary to perform clustering analysis.

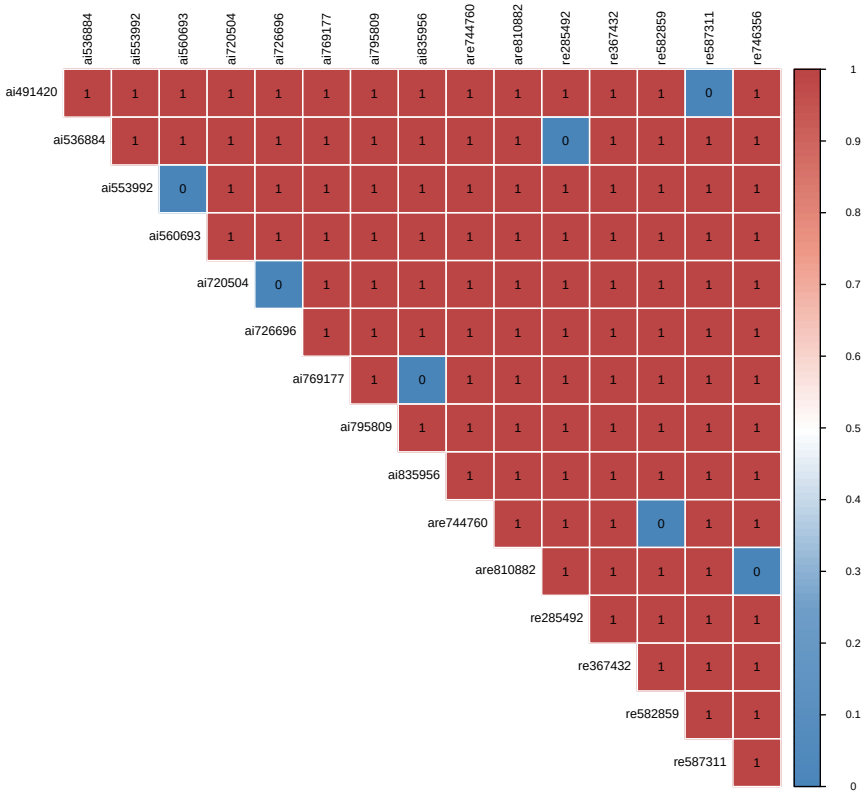

Figure 6 – A similarity matrix (binary) for the rapporteur property

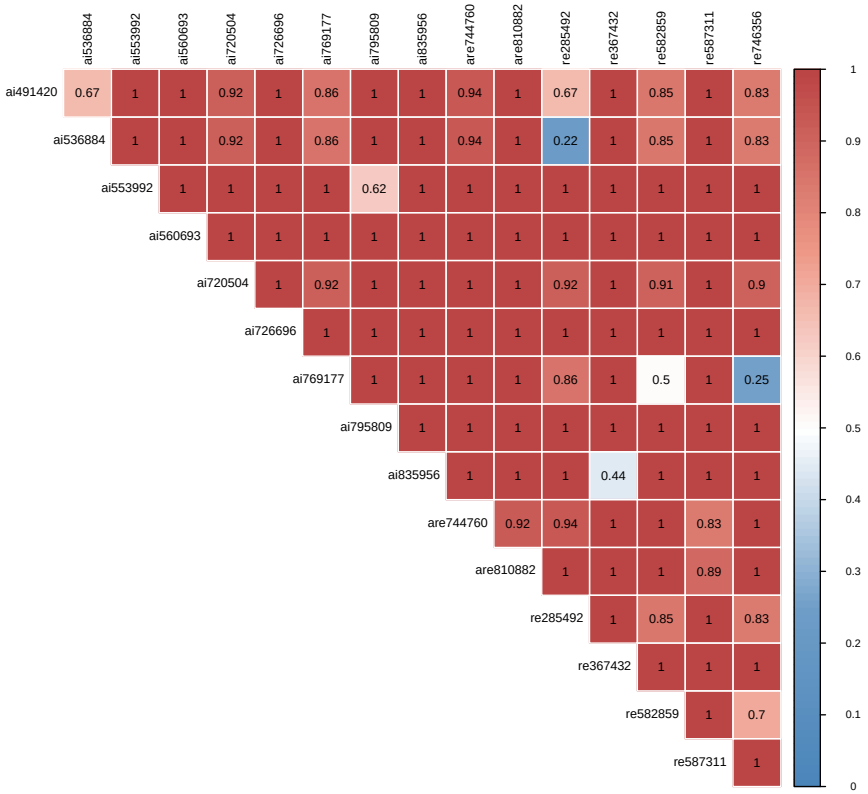

Figure 7 – A similarity matrix (non-binary) for the summary property

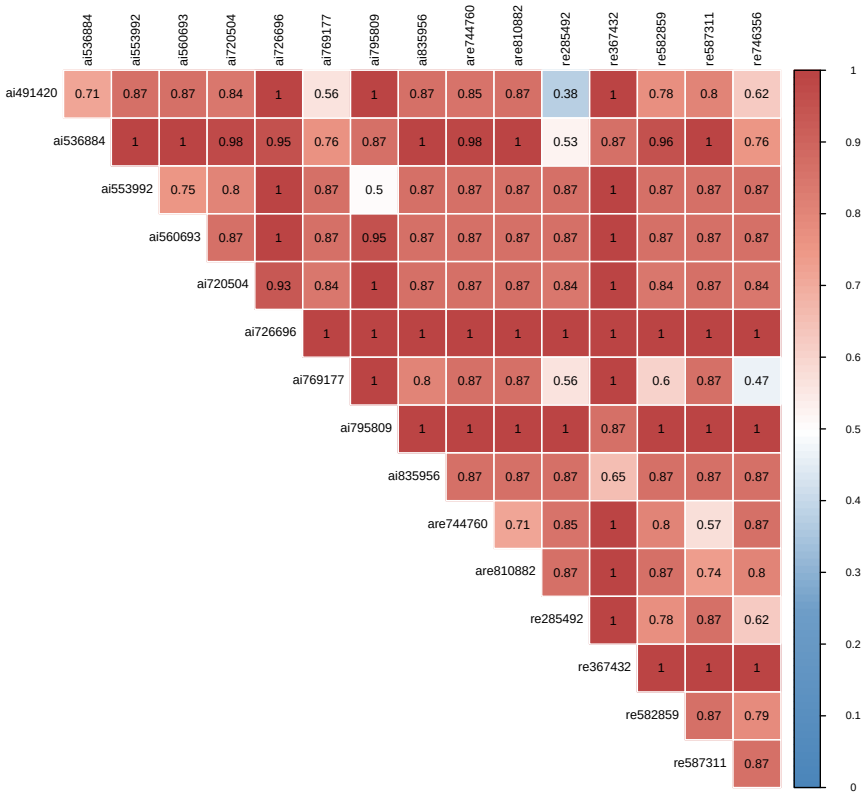

Figure 8 – The mean similarity matrix generated according to weights assigned

5.3 Clustering analysis

Before describing input data and discussing the results achieved by agglomerative algorithms and their effects on the final assignment, let's see some visualizations for clustering operation. Figure 9 illustrates a hierarchical clustering for a dataset of 16 lawsuits, by using Ward's method, showing that by joining the branches leads to a final cluster of size N. Each item in the tree corresponds to a lawsuit in dataset, which is identified by its key number (ID). Is is important to note that by cutting the tree intermediate results can be achieved.

In figure 10, the similarity matrix is presented by a heat map. A heat map is a two-dimensional visual representation of data that provides an immediate visual summary of information. It uses colors to communicate relationships between data values that would be much harder to understand if presented numerically in a spreadsheet. In this case, Each row and column entry corresponds to a lawsuit in dataset. The color of a cell within the matrix represents the degree of similarity

between two lawsuits. The darker, then more similar the lawsuits are. The heat map visualization is provided by `gplots` R library, and the adaptation for this work was based in Raschka (2013).

Beyond the heat map visualization itself, figure 10 display clusters being formed from the similarity matrix. The rows (columns) of the matrix are ordered such that similar rows (columns) are near each other (FRIENDLY; WILKINSON, 2009). On the left margin of the matrix are hierarchical cluster trees.

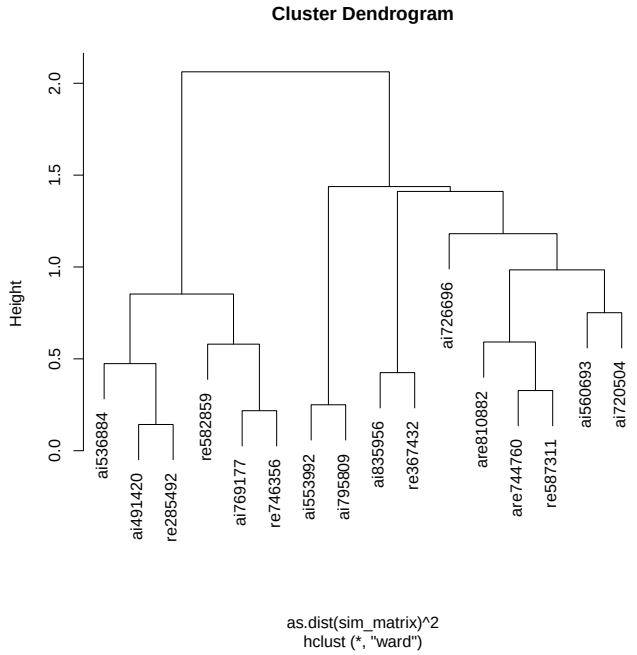

Figure 9 – A dendrogram (tree diagram) illustrating a hierarchical clustering for lawsuits

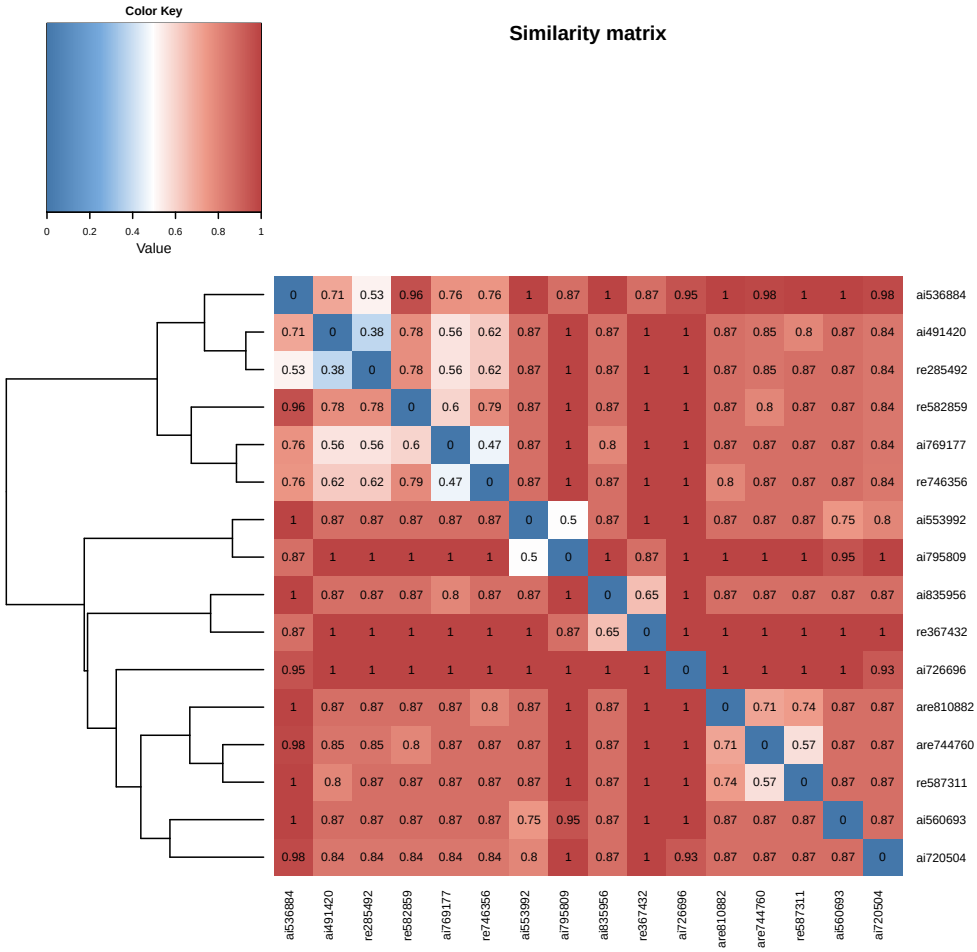

Figure 10 – Heatmap illustrating clusters formation from lawsuits similarities

5.4 Input

Once the clustering analysis was made and the dataset is classified, it is time to evidence what input structure is used, in order to understand the achieved results. The input is basically a properties file that can be easily customizable for test purposes, and contains the following information:

- Agglomeration method: defines which agglomeration algorithm will be used in the hierarchical clustering. Here the proper value for the `method` argument in the R `hclust` command is expected (`ward`, `single`, `complete`, `average`, etc). For tests purpose, the methods Ward's, Single-linkage and Complete-linkage were used. The comparison between their performances is described in section 5.5.

- Instance: describes a lawsuit instance that will have the outcome predicted. The format is the same as a lawsuit data modelled in the dataset. The fields taken into account are rapporteur, defendant, author, type, subjects, summary and claim. Table 2 exemplifies it.

- Judges: is the list of ministers that would judge the lawsuit. It can be composed by a single judge (in case of a monocratic decision), by five judges (in case of First Panel or Second Panel) or by eleven judges (in case of Plenary).

For exemplification purpose and to address the results obtained in the next section, the test instances presented in table 2 were used.

Test instance 1	
Rapporteur	Joaquim
Defendant	Company LTDA
Author	João da Silva
Type	Direito Administrativo
Subjects	atos administrativos, licenças
Summary	lei municipal,edificações,telefonia,direito de construir, competencia legislativa, uso do solo urbano
Claim	competência legislativa
Test instance 2	
Rapporteur	Carmen
Defendant	Estado do Rio de Janeiro
Author	José das Neves
Type	Direito Administrativo
Subjects	cumprimento,execução de sentença,inadimplência
Summary	desapropriação,liquidação,precatório,juros compulsórios, juros compensatórios,intervenção do estado na propriedade
Claim	dispensa da expedição de precatório
Test instance 3	
Rapporteur	Toffoli
Defendant	Município de Cabo Frio
Author	Maria da Silva
Type	Direito Administrativo
Subjects	responsabilidade da administração,indenização por dano material, indenização por dano moral
Summary	responsabilidade civil objetiva,omissão na conservação pública, comprovação de danos,dano moral configurado
Claim	responsabilidade objetiva

Table 2 – Description of three test instances

5.5 Agglomerative algorithms performances for clustering

The tables 3, 4 and 5 display the similarity rate for each test instance, considering the number of clusters varying from 2 to $\sqrt{S}$, as mentioned previously. The similarity rate is obtained by calculating the similarity average between all lawsuits in a cluster and the lawsuit instance.

	Ward's best cluster		Single-linkage best cluster		Complete-linkage best cluster	
k	Similarity rate	Size	Similarity rate	Size	Similarity rate	Size
2	**0.53**	6	0.75	15	0.74	14
3	**0.53**	6	0.69	11	0.72	13
4	**0.53**	6	0.69	11	0.62	7

Table 3 – Clustering results obtained for instance 1

	Ward's best cluster		Single-linkage best cluster		Complete-linkage best cluster	
k	Similarity rate	Size	Similarity rate	Size	Similarity rate	Size
2	**0.84**	10	**0.84**	15	0.84	14
3	**0.66**	2	0.75	4	0.83	13
4	**0.66**	2	0.71	3	0.83	6

Table 4 – Clustering results obtained for instance 2

	Ward's best cluster		Single-linkage best cluster		Complete-linkage best cluster	
k	Similarity rate	Size	Similarity rate	Size	Similarity rate	Size
2	**0.79**	10	0.81	15	0.81	14
3	**0.75**	8	0.78	11	0.80	13
4	0.71	6	0.78	11	**0.68**	6

Table 5 – Clustering results obtained for instance 3

In general, as can be seen in the tables above, Ward's method was the most efficient for this problem. However, for the third test instance, the Complete-linkage method was the best. It is interesting to note that the Ward's method favours the formation of clusters whose sizes do not differ too much, when compared to Simple and Complete-linkage methods. This naturally leads to clusters more heterogeneous internally.

5.6 Prediction results

The output shown in tables 6, 7 and 8 corresponds to the final results for each instance test, once the most similar cluster was chosen in each case. The *Votes* vector indicates the degree of acceptance of each judge in the cluster. That is, it corresponds to the ratio $\frac{A_i}{A_i+D_i}$, where A_i is the number of lawsuits in the cluster i favourably judged (accepted) by the judge and D_i is the number of lawsuits unfavourably judged (denied) in the same cluster. Thus, 0 means that the judge denied all lawsuits in the cluster and 1 means that all lawsuits were accepted by him/her. So, if the judge judged 15 cases belonging to the cluster in question and accepted 6 of them, his/her votes coefficient is $\frac{6}{15} = 0.4$. It is important to note that each element of the vector represents a judge. In this way, for results shown in tables 6, 7 and 8, we have a vector of size 5, which means that we are testing the instance considering a Collegiate judgement. In addition, is important to note that the prediction results for these test instances were achieved considering a configuration with the following judges participating: Toffoli, Gilmar, Fux, Lewandowski and Marco. However, the list of judges can be easily changed, by just editing the properties file.

The *Judges agreement* vector corresponds to the degree of agreement for each judge, *i.e.*, for each lawsuit belonging to the cluster where the judge voted and other judges composing the input also voted, it is computed if they agreed or disagreed (1 or 0 is assigned, respectively). Then, the average is calculated for each judge. If the judge did not participate in any lawsuit belonging to the cluster, then "n/a" is displayed. As well as the *Votes* vector, it is important to note that each element of the *Judges agreement* vector represents a judge.

The *Likely outcome* column is the average applied in the *Votes* vector. The *Judges agreement* vector is not used for outcome calculation purpose, but is an extra and useful information to reinforce the degree of uncertainty of the outcome and to detect whether a certain judge usually has a discrepant behaviour within the cluster context.

Finally, the results were obtained from the input described in Section 5.4 compared to datasets `lawsuit_16.csv` and `decision_16.csv`, described in subsection 5.1.

As can be seen in the tables 6, 7 and 8, the outcomes prediction for the first and the third instances seems to be well established. However, for the second instance there is a big uncertainty. This bring out two factors. The first is related to the fact that the cluster has a similar proportion of favourable (accepted) and not favourable (denied) decisions, which demands a deep analysis in lawsuits composing

k	Best cluster	Similarity rate	Votes	Judges agreement	Likely outcome	Lawsuits mean duration (days)
2	1	0.53	<0, 0, 0, 0, 0>	<1, 1, 1, 1, 1>	**0**	1254.33
3	1	0.53	<0, 0, 0, 0, 0>	<1, 1, 1, 1, 1>	**0**	1254.33
4	1	0.53	<0, 0, 0, 0, 0>	<1, 1, 1, 1, 1>	**0**	1254.33

Table 6 – Outcomes for instance 1, considering best solution found (Ward's algorithm)

k	Best cluster	Similarity rate	Votes	Judges agreement	Likely outcome	Lawsuits mean duration (days)
2	2	0.84	<0.25, 0, 0.29, 0, 0.43>	<0.93, 1, 0.93, 1, 0.86>	**0.19**	1463.1
3	2	0.66	<0.5, n/a, 0.5, n/a, 0.5>	<1, n/a, 1, n/a, 1>	**0.5**	1862
4	2	0.66	<0.5, n/a, 0.5, n/a, 0.5>	<1, n/a, 1, n/a, 1>	**0.5**	1862

Table 7 – Outcomes for instance 2, considering best solution found (Ward's algorithm)

k	Best cluster	Similarity rate	Votes	Judges agreement	Likely outcome	Lawsuits mean duration (days)
2	1	0.81	<0.25, 0, 0.29, 0, 0.38>	<0.93, 1, 0.93, 1, 0.86>	**0.18**	1313
3	1	0.80	<0.14, 0, 0.17, 0, 0.29>	<0.92, 1, 0.92, 1, 0.83>	**0.12**	1299.31
4	2	0.68	<0, 0, 0, 0, 0.33>	<0.83, 1, 0.83, 1, 0.67>	**0.07**	1286.5

Table 8 – Outcomes for instance 3, considering best solution found (Complete-linkage algorithm)

the cluster to then take more accurate conclusions about the lawsuit being tested. But it can still be a useful result, since such result is an important filter in the whole dataset. Another factor to have in mind when such uncertainty emerges, is that some judges included in the input did not participated in any lawsuit of the cluster, which is the case of instance in table 7. This tends to disappear as the dataset size increases.

6 Conclusions and Future work

Much of the analysis applied in judicial context demand deep domain knowledge and ability to deal with subjective questions, such as, the interpretation of laws. However, the exploration of similarity and clustering methods can be a key contribution in this field, since they provide results that bring useful information for outcome predictions.

In this work, the efforts were directed in order to use similarity calculations and clustering algorithms to find the most similar lawsuits when compared to a new lawsuit being tested. By analysing some metadata, as the subjects, summary and claim of a lawsuit, it is possible that this test instance had a similar case already judged, since the amount of data provided by the judicial system are quite large. Thus, the challenge is to find the best way to detect such similarities and evaluate how close the past emerging results are from the lawsuit being tested. By the developing of a program that detects clusters and compiles past votes – and with the addition of some tunings – the results shown that was possible to verify the most likely outcome and to detect the degree of uncertainty of the outcome, by checking the degree of voting agreement between judges in a certain scope.

In this sense, prediction results were satisfied when analysing them manually. Indeed, lawsuit instances were correctly assigned to clusters and similarity comparison revealed a good coefficient between lawsuits. But these aspects reveals a limitation of the work. It is currently using artificial instances for test, which can introduce some bias. As an improvement, it would be better to collect real instances, predict them and then verify the accuracy rate through an evaluation metric, as *R-squared* or *Cross-validation*. This is possible because most of lawsuits available in STF site were already judged, so final results are available for some of them.

As future work, some improvements and modifications are already addressed. Firstly, the use of more sophisticated machine learning techniques are required, in order to improve the results accuracy and to detect other relationships that are currently hidden, by analysing full and unstructured texts existing in many judicial documents. Secondly, it is important to investigate a more efficient clustering method than the hierarchical clustering. This technique is good if we assume that such method considers different scenarios, but it is not so good in terms of performance, once it finds many intermediate results that are not relevant. Then, for future work it could be defined a criterion for determining the number of clusters

according to the dataset size and other characteristics, as the variance of the points in the dataset. However, the correct choice of the number of clusters is not a trivial decision, it depends on the distribution of points in a dataset and on the desired clustering resolution. Thirdly, it is necessary to discriminate decisions by type. It means that a mechanism to simulate and analyse monocratic decisions separately of internal appeals is a relevant aspect from the judicial point of view. The current solution handle all kind of decision as having the same characteristics and dynamics and contribute with the same weight for the votes compilation. Finally, a better mechanism to find lawsuits properties weights should be developed. For instance, SVM (Support vector machines) provides learning algorithms to detect feature weights from data relationships. One limitation, however, is that SVM works well when values are described by numerical series for each entry. In this problem, the extracted properties are mostly literals, then the way to generate numerical values is creating an adjacency matrix from a similarity metric (Jaccard, in this case), where each coefficient represents the similarity degree of a pair of entries (lawsuits). To SVM accepts this, it is necessary to apply some normalization to generate coefficients per lawsuit, not per pair.

Another direction of this work is the modelling and simulation of lawsuit decision process on which multiple variables are predicted, multiple scenarios are explored and *what-if* analyses could be done, in order to provide a more flexible solution. Other approaches, as the one adopted in (GATTI et al., 2013), consider the use of stochastic modelling and simulation to evaluate the emergent behaviour originated from the observed relationships. In this case, the key target is to be able to predict human behaviour like posting, forwarding or replying a message with regard to topics and sentiments, and to analyse the emergent behaviour of such actions in a social network, as Twitter. To tackle this problem, a method to model and simulate interactive behaviour was presented. The use of a stochastic multi-agent based approach was useful to demonstrate that with this engineering method that first models each network user individually as an agent it is possible to develop social media simulators using a bottom-up approach (micro level) to evaluate the emergent behaviour (macro level). One could evolve the solution herein proposed using the same rationale of the (GATTI et al., 2013) approach.

Bibliography

ACKERMAN, M.; BEN-DAVID, S. *Discerning Linkage-Based Algorithms Among Hierarchical Clustering Methods*. 2011. Available from internet: <https://www.aaai.org/ocs/index.php/IJCAI/IJCAI11/paper/view/3116>. Mentioned on page(s) 25.

AGGARWAL, C. C.; REDDY, C. K. (Ed.). *Data Clustering: Algorithms and Applications*. [S.l.]: CRC Press, 2014. Available from internet: <http://www.charuaggarwal.net/clusterbook.pdf>. ISBN 978-1-46-655821-2. Mentioned on page(s) 20.

ALDENDERFER, M. S.; BLASHFIELD, R. K. *Cluster Analysis*. Beverly Hills: Sage, 1984. (Sage University Paper Series on Quantitative Applications in the Social Sciences, 07-044). ISBN 0 8039 2376 7. Mentioned on page(s) 20.

BORGATTI, S. P. How to explain hierarchical clustering. 1994. Available from internet: <http://www.analytictech.com/networks/hiclus.htm>. Mentioned on page(s) 23.

CAO, L. et al. *Agents and Data Mining Interaction: 7th International Workshop, ADMI 2011, Taipei, Taiwan, May 2-6, 2011, Revised Selected Papers*. [S.l.]: Springer, 2012. (LNCS sublibrary. SL 7, Artificial intelligence). Available from internet: <https://books.google.com.br/books?id=zQyZm2drScMC>. ISBN 9783642276088. Mentioned on page(s) 31.

CHIPMAN, H.; TIBSHIRANI, R. Hybrid hierarchical clustering with applications to microarray data. *Biostatistics*, Oxford University Press, v. 7, n. 2, abr. 2006. ISSN 1465-4644. Mentioned on page(s) 23.

FRIENDLY, M.; WILKINSON, L. The history of the cluster heat map. *The American Statistician*, 2009. Mentioned on page(s) 40.

GATTI, M. et al. Large-scale multi-agent-based modeling and simulation of microblogging-based online social network. In: *Multi-Agent-Based Simulation XIV - International Workshop, MABS 2013, Saint Paul, MN, USA, May 6-7, 2013, Revised Selected Papers*. [S.l.: s.n.], 2013. Available from internet: <http://dx.doi.org/10.1007/978-3-642-54783-6_2>. Mentioned on page(s) 47.

GRIGORYEV, I. *AnyLogic 6 in three days - A quick course in simulation modeling*. [S.l.]: AnyLogic North America, 2012. ISBN 0615705677. Mentioned on page(s) 11.

JACCARD, P. The distribution of the flora in the alpine zone. *New Phytologist*, v. 11, n. 2, p. 37–50, fev. 1912. Mentioned on page(s) 29.

KUNDU, M. *Advanced Computing, Networking and Informatics, Volume 1: Advanced Computing and Informatics Proceedings of the Second International*

Conference on Advanced Computing, Networking and Informatics (Icacni-2014). [S.l.]: Springer, 2014. (Smart Innovation, Systems and Technologies). ISBN 9783319073538. Mentioned on page(s) 20.

RASCHKA, S. A short tutorial for decent heat maps in r. 2013. Available from internet: <http://sebastianraschka.com/Articles/heatmaps_in_r.html>. Mentioned on page(s) 40.

SANDER, J. et al. Automatic extraction of clusters from hierarchical clustering representations. In: *Proc. of 7th Pacific-Asia Conf. of Advances in Knowledge Discovery and Data Mining, PAKDD 2003, Proceedings*. [S.l.]: Springer, 2003. p. 75–87. Mentioned on page(s) 22.

SASIREKHA, K.; BABY, P. Agglomerative Hierarchical Clustering Algorithm- A Review. *International Journal of Scientific and Research Publications (IJSRP)*, v. 3, mar. 2013. Mentioned on page(s) 22.

SCHUTT, R.; O'NEIL, C. *Doing Data Science: Straight Talk from the Frontline*. [S.l.]: O'Reilly Media, Inc., 2013. ISBN 1449358659, 9781449358655. Mentioned on page(s) 19.

STF. Institucional. 2011. Available from internet: <http://www.stf.jus.br/portal/cms/verTexto.asp?servico=sobreStfConhecaStfInstitucional>. Mentioned on page(s) 14.

STF. Fases da tramitação processual. 2012. Available from internet: <http://www.stf.jus.br/portal/cms/verTexto.asp?servico=estatistica&pagina=explicafases>. Mentioned on page(s) 15.

SURDEN, H. Machine Learning and Law. *Social Science Research Network Working Paper Series*, mar. 2014. Available from internet: <http://ssrn.com/abstract=2417415>. Mentioned on page(s) 12.

TAN, P. N.; STEINBACH, M.; KUMAR, V. *Introduction to Data Mining, (First Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005. ISBN 0321321367. Mentioned on page(s) 20.

VIRVOU, M.; ALEPIS, E.; TROUSSAS, C. Centroid-based clustering for student models in computer-based multiple language tutoring. In: CABELLO, E. et al. (Ed.). [S.l.]: SciTePress, 2012. p. 198–203. ISBN 978-989-8565-25-9. Available from internet: <http://dblp.uni-trier.de/db/conf/sigmap/sigmap2012.html>. Mentioned on page(s) 31.

WARD, J. H. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, v. 58, n. 301, p. 236–244, 1963. Available from internet: <http://www.jstor.org/stable/2282967>. Mentioned on page(s) 24.

Appendix

APPENDIX A – R clustering script

```
run_clustering <- function(num_clusters, agglomeration_method) {
    #as.dist try to force a object to a distance matrix
    sim_matrix <- as.matrix(read.table("similarity.csv", header=TRUE, sep
    = "\t", as.is=TRUE))
    h_cluster <- hclust(as.dist(sim_matrix)^2, method=agglomeration_method
    )
    for(i in 2:num_clusters) {
        c_tree <- cutree(h_cluster, k=i)
        write.table(c_tree, sep="\t")
    }
}
```

APPENDIX B – Java function to compile judges votes

```
public void getJudgesAcceptedVotes(List<String> judgesList) {
    System.out.println("#" + cluster + " Votes");
    for(String j : judgesList) {
        int numDecisionsParticipated = 0;
        int numDecisionsAccepted = 0;

        for(Lawsuit l : lawsuits.values()) {
            if(l.getCluster() == cluster) {
                for(Decision d : l.getDecisions()) {
                    if(d.getVotes().containsKey(j)) {
                        numDecisionsParticipated++;
                        if(d.getVotes().get(j) == DecisionResult.ACCEPTED)
     {
                            numDecisionsAccepted++;
                        }
                    }
                }
            }
        }

        DecimalFormat decimalFormat = new DecimalFormat("#.##");

        if(numDecisionsParticipated == 0) {
            System.out.println(j + ": n/a");
        } else {
            System.out.println(j
                    + ": "
                    + decimalFormat.format((1.0 * numDecisionsAccepted)
                            / numDecisionsParticipated));
        }
    }
}
```

APPENDIX C – Java function to detect judges agreement

```
public void getJudgesAgreement(List<String> judgesList) {
    System.out.println("#" + cluster + " Agreement");
    for(String j : judgesList) {
        int numCommonVotes = 0;
        int numAgreements = 0;

        for(Lawsuit l : lawsuits.values()) {
            if(l.getCluster() == cluster) {
                for(Decision d : l.getDecisions()) {
                    if(d.getVotes().containsKey(j)) {
                        for(String other : d.getVotes().keySet()) {
                            if(d.getVotes().containsKey(other) && !j.
    equals(other) && judgesList.contains(other)) {
                                if(d.getVotes().get(j) == d.getVotes().get
    (other)) {
                                    numAgreements++;
                                }
                                numCommonVotes++;
                            }
                        }
                    }
                }
            }
        }

        DecimalFormat decimalFormat = new DecimalFormat("#.##");

        if(numCommonVotes == 0) {
            System.out.println(j + ": n/a");
        } else {
            System.out.println(j
                    + ": "
                    + decimalFormat.format((1.0 * numAgreements)
                            / numCommonVotes));
        }

    }
}
```

APPENDIX D – Git repository

To access the program code and the datasets used in this project, please send a message to `daniel.gribel@uniriotec.br` requesting access to the `lawsim` Git repository hosted at Bitbucket.

Annex

ANNEX A – Lawsuit example (metadata)

RE 582859 - RECURSO EXTRAORDINÁRIO

Número do Protocolo:	**2008/44254**
Data de Entrada no STF:	**01/04/2008**

PROCEDÊNCIA

Número:	**ADI 70016528374**
Orgão de Origem:	**TRIBUNAL DE JUSTIÇA ESTADUAL**
Origem:	**RIO GRANDE DO SUL**

Volume: **1** Apensos:**0** Folhas:**77** Qtd.juntada linha: **0**

SUPREMO TRIBUNAL FEDERAL

Ramo do Direito	**DIREITO ADMINISTRATIVO E OUTRAS MATÉRIAS DE DIREITO PÚBLICO** |
Assunto	**Atos Administrativos** | **Licenças** | **Funcionamento de Estabelecimentos Empresariais**
Folhas	**77**
Data de Autuação	**02/04/2008**

PARTES

Categoria	Nome
RECTE.(S)	MINISTÉRIO PÚBLICO DO ESTADO DO RIO GRANDE DO SUL
PROC.(A/S)(ES)	PROCURADOR-GERAL DE JUSTIÇA DO ESTADO DO RIO GRANDE DO SUL
RECDO.(A/S)	CÂMARA MUNICIPAL DE PASSO FUNDO
ADV.(A/S)	EUCLIDES SERÁPIO FERREIRA

ANNEX B – Lawsuit example (life cycle)

RE 582859 - RECURSO EXTRAORDINÁRIO

Origem:	**RS - RIO GRANDE DO SUL**
Relator:	**MIN. CÁRMEN LÚCIA**
RECTE.(S)	**MINISTÉRIO PÚBLICO DO ESTADO DO RIO GRANDE DO SUL**
PROC.(A/S)(ES)	**PROCURADOR-GERAL DE JUSTIÇA DO ESTADO DO RIO GRANDE DO SUL**
RECDO.(A/S)	**CÂMARA MUNICIPAL DE PASSO FUNDO**
ADV.(A/S)	**EUCLIDES SERÁPIO FERREIRA**

Data	Andamento	Órgão Julgador	Observação	Documento
11/12/2013	Baixa definitiva dos autos, Guia nº		41824/2013 - TRIBUNAL DE JUSTIÇA DO ESTADO DO RIO GRANDE DO SUL	
10/12/2013	Transitado(a) em julgado		Em 6.12.2013.	
06/11/2013	Juntada de AR		ao Ministério Público do Estado do Rio Grande do Sul. JL373623910BR	
23/10/2013	Expedida intimação via postal		ao Ministério Público do Estado do Rio Grande do Sul. JL373623910BR	
21/10/2013	Publicado acórdão, DJE		DATA DE PUBLICAÇÃO DJE 21/10/2013 - ATA Nº 158/2013. DJE nº 208, divulgado em 18/10/2013	Inteiro teor do acórdão
03/10/2013	Ata de Julgamento Publicada, DJE		ATA Nº 26, de 24/09/2013. DJE nº 194, divulgado em 02/10/2013	
27/09/2013	Juntada		certidão de julgamento	
24/09/2013	Agravo regimental não provido	SEGUNDA TURMA	Decisão: A Turma, por unanimidade, negou provimento ao agravo regimental, nos termos do voto da Relatora. 2ª Turma, 24.09.2013.	Decisão de Julgamento
24/09/2013	Apresentado em mesa para julgamento		2ª Turma em 24/09/2013 11:04:42 - RE-AgR	
06/09/2013	Conclusos ao(à) Relator(a)			
06/09/2013	Interposto agravo regimental		Juntada Petição: 42447/2013	
30/08/2013	Petição		Agravo Regimental - Petição: 42447 Data: 30/08/2013 15:49:15.653 GMT-03:00	
30/08/2013	Juntada de AR		ao Ministério Público do Estado do Rio Grande do Sul, na pessoa do Procurador-Geral de Justiça, ou na de quem as suas vezes fizer. JL373588594BR	
19/08/2013	Expedida intimação via postal		ao Ministério Público do Estado do Rio Grande do Sul, na pessoa do Procurador-Geral de Justiça, ou na de quem as suas vezes fizer. JL373588594BR	
15/08/2013	Publicação, DJE		DJE nº 159, divulgado em 14/08/2013	Decisão monocrática
14/08/2013	Juntada de AR		ao Ministério Público do Estado do Rio Grande do Sul, na pessoa do Procurador-Geral de Justiça, ou na de quem as suas vezes fizer. JL373551166BR	
13/08/2013	Negado seguimento	MIN. CÁRMEN		

ANNEX C – Modelled lawsuits input file

re285492 28/07/2000,28/11/2000 ms joaquim banco do brasil diretor de posturas e controle ambiental do município de campo grande direito administrativo atos administrativos,fiscalização lei municipal,competencia legislativa,município,bancos,segurança,portas de acesso,interesse local,conforto e rapidez no atendimento competencia legislativa
ai491420 22/01/2004,08/05/2006 sp peluso febraban josé lavelli de lima direito administrativo atos administrativos,fiscalização lei municipal,competência legislativa,município,edificações,bancos,equipamentos de segurança,portas eletrônicas,exigência de equipamentos de segurança competencia legislativa
ai536884 16/03/2005,20/09/2012 rs joaquim banco real município de passo fundo direito processual civil e do trabalho jurisdição,competência legislativa lei municipal,competência legislativa,município,bancos,segurança,câmeras,interesse local,conforto e rapidez no atendimento competência legislativa
ai769177 07/10/2009,28/03/2014 sp toffoli claro municipio de são josé dos campos direito administrativo atos administrativos,licenças lei municipal,direito de construir,competência legislativa,construção irregular,ordenamento urbano,uso do solo urbano,telefonia,antena competencia legislativa
are744760 19/04/2013,11/11/2013 rj carmen município do rio de janeiro ricardina goncalves valle direito administrativo responsabilidade da administração,indenização por dano moral danos morais,indenização,responsabilidade civil objetiva,município,violência física,estudante,professor,dever de segurança,instituição de ensino responsabilidade objetiva
are810882 06/05/2014,19/08/2014 rj gilmar município de niteroi daniel leitão ribeiro direito administrativo responsabilidade da administração,indenização por dano material,indenização por dano moral responsabilidade civil objetiva,omissão na conservação pública,queda de transeunte em bueiro,comprovação de danos,dano moral configurado responsabilidade subjetiva
re587311 20/05/2008,01/04/2011 rj peluso município de nova iguacu maria izabel fernandes da silva direito administrativo responsabilidade da administração,indenização por dano material,acidente de trânsito danos materiais,indenização,responsabilidade civil objetiva,estado,acidente de trânsito responsabilidade objetiva
re746356 25/04/2013,14/08/2013 sp gilmar tim municipio de ribeirão preto direito administrativo controle de constitucionalidades,institucionalização material lei municipal,direito de construir,competencia legislativa,uso do solo urbano,telefonia,antena competencia legislativa
ai560693 19/09/2005,16/10/2012 sp marco município de são paulo espólio de arcilio loverri direito administrativo intervenção do estado na propriedade,desapropriação,liquidação,cumprimento,execução de sentença,precatório juros,moratórios,compensatórios,débito da fazenda exclusão de juros moratórios
ai720504 30/06/2008,25/02/2013 sp rosa estado de são paulo carlos eduardo doria da silva chaves direito administrativo servidor público civil,aposentadoria,sistema remuneratório,benefícios complementação de aposentadoria,complementação de benefício,aplicação de legislação local,ferroviário,competencia legislativa tempo insuficiente para aposentadoria integral
ai835956 26/01/2011,15/08/2013 ma toffoli estado do maranhão ministerio publico do estado do maranhão direito administrativo serviços,defensoria pública ação civil pública,ampliação de atuação,defensoria pública,legitimidade,relevância institucional,implementação de políticas públicas,princípio da separação dos poderes violação do princípio da separação dos poderes
re367432 28/11/2002,10/09/2010 pr grau estado do paraná ministerio publico do estado do paraná direito processual civil e do trabalho formação do processo,suspensão do processo,extinção do processo ação civil pública,segurança pública,defensoria pública,legitimidade,implementação de políticas públicas,princípio da separação dos poderes,omissão administrativa violação do princípio da separação dos poderes
re582859 02/04/2008,11/12/2013 rs carmen câmara municipal de passo fundo ministerio publico do estado do rio grande do sul direito administrativo atos administrativos,licenças,funcionamento de estabelecimentos empresariais lei municipal,instalação de supermercados,limitação horizontal de edificação,direito de construir,construção irregular,competencia legislativa,ordenamento urbano contrariedade a livre iniciativa e concorrência
ai726696 11/09/2008,11/10/2012 al rosa estado de alagoas berckmam de almeida nunes direito processual civil e do trabalho jurisdição,competência,administração pública impugnação às razões de decidir,superação da decisão agravada,novo exame,fgts,contrato individual de trabalho,contrato nulo novo exame das razões recursais
ai795809 12/04/2010,18/03/2013 rs fux união celso pereira goulart direito processual civil e do trabalho cumprimento,execução de sentença,obrigações,inadimplência liquidação,pagamento em desacordo,precatório,crédito complementar,juros de mora exclusão de juros
ai553992 26/07/2005,31/10/2012 sp marco estado de são paulo paulo franco direito administrativo cumprimento,execução de sentença,obrigações,inadimplência desapropriação,liquidação,precatório,juros de mora,intervenção do estado na propriedade,juros compensatórios exclusão de juros

ANNEX D – Modelled decisions input file

```
re-m 746356     re746356        monocratica     07/05/2013      gilmar:denied   denied
re-agr 746356   re746356        segunda_turma   28/05/2013
gilmar:denied;carmen:denied;celso:denied;lewandowski:denied;teori:denied        denied
ai-m 769177     ai769177        monocratica     06/11/2013      toffoli:denied  denied
ai-agr 769177   ai769177        primeira_turma  18/02/2014
toffoli:denied;marco:denied;fux:denied;rosa:denied;barroso:denied       denied
ai-agr 491420   ai491420        primeira_turma  21/02/2006
peluso:denied;sepulvida:denied;marco:denied;britto:denied;grau:denied   denied
ai-agr 536884   ai536884        segunda_turma   26/06/2012
joaquim:denied;lewandowski:denied;celso:denied;gilmar:denied;peluso:denied      denied
re-agr 285492   re285492        segunda_turma   28/08/2012
joaquim:denied;lewandowski:denied;celso:denied;gilmar:denied;peluso:denied      denied
are-m 744760    are744760       monocratica     16/05/2013      carmen:denied   denied
are-agr 744760  are744760       segunda_turma   15/10/2013
carmen:denied;gilmar:denied;teori:denied;barroso:denied denied
re-m 587311     re587311        monocratica     17/11/2009      peluso:denied   denied
re-agr 587311   re587311        segunda_turma   16/11/2010
gilmar:denied;celso:denied;gracie:denied;britto:denied;joaquim:denied   denied
are-m 810882    are810882       monocratica     22/05/2014      gilmar:denied   denied
are-agr 810882  are810882       segunda_turma   27/05/2014
gilmar:denied;teori:denied;lewandowski:denied   denied
ai-agr 560693   ai560693        primeira_turma  21/08/2012
marco:denied;toffoli:denied;fux:denied;rosa:denied      denied
ai-agr 720504   ai720504        primeira_turma  28/08/2012
rosa:denied;toffoli:denied;marco:accepted;carmen:denied;fux:denied      denied
ai-ed 720504    ai720504        primeira_turma  11/12/2012
rosa:denied;toffoli:denied;marco:denied;carmen:denied;fux:denied        denied
ai-m 835956     ai835956        monocratica     05/03/2013      toffoli:denied  denied
ai-agr 835956   ai835956        primeira_turma  07/05/2013
toffoli:denied;fux:denied;marco:denied;rosa:denied      denied
re-agr 367432   re367432        segunda_turma   20/04/2010
grau:denied;peluso:denied;joaquim:denied;gracie:denied;celso:denied     denied
re-m 582859     re582859        monocratica     15/08/2013      carmen:denied   denied
re-agr 582859   re582859        segunda_turma   24/09/2013
carmen:denied;celso:denied;gilmar:denied;lewandowski:denied;teori:denied        denied
ai-agr 726696   ai726696        primeira_turma  28/08/2012
rosa:accepted;toffoli:accepted;marco:accepted;carmen:accepted;fux:accepted      accepted
ai-agr 795809   ai795809        primeira_turma  18/12/2012
toffoli:accepted;marco:accepted;fux:accepted;rosa:accepted      accepted
ai-agr 553992   ai553992        primeira_turma  25/09/2012
toffoli:denied;marco:denied;fux:denied;rosa:denied      denied
```