Universitat Politcnica de Catalunya
Departament de Llenguatges i Sistemes Informtics
Master in Artificial Intelligence

Master Thesis

Inducted Concepts From Embbeded Classes For Automatic Interpretation In Hierarchical Clustering

Student: Esther Lozano Hontecillas
Thesis Advisor: Karina Gibert

Barcelona, June 22^{nd}, 2009

Contents

Acknowledgements

- Institute Guttmann-Hospital de Neurorehabilitaci. Badalona, Spain.
- Tribu KLASS, especially to Alejandro García and Alejandra Pérez.

Chapter 1

Introduction and motivation

1.1 Formulation of the problem

In very complex and unstructured domains, the Intelligent Decision Support Systems become very important tools for the expert, since allow to manage a quantity of information in a way that would be impossible to do manually. Inside this kind of systems, the classification tools are one of the most common, and, specifically, the clustering techniques. However, these techniques have problems when managing huge amount of variables and classes, because the interpretation of the generated classes becomes very complicate.

For this reason, in this project we want to generate an automatically conceptual interpretation of the classes generated by a clustering technique to help in the labor of the expert with a clearer vision of what is representing each class in order to understand quickly and easy what are the properties and characteristics of these data.

1.2 Document Structure

In chapter §2 the general and particular objectives of this thesis are exposed. In chapter §3 is described the state of the art, that allows to contextualize the topic of this work. In chapter §4 there is the description of soma basic concepts, necessary to the correct understanding of this thesis. In chapter §5 we found the context within this thesis is sited, that is, the description of the KLASS application. In chapter §6 there is described the methodology followed for the carrying out of this project. In chapter §7 we found more detailed information about the introduced changes in KLASS and their implementation. In chapter §8 is presented a case study and the results obtained with the new version of the application. Finally, in chapter §9 the conclusions and future work are exposed.

Chapter 2

Objectives

2.1 General objectives

1. Contribute to systematize the process of interpretation of classes coming from a hierarchical cluster, process that right now is done more or less by hand.

2. Add objectivity to the mechanisms of interpretation of classes coming from a hierarchical cluster.

3. Generate explicit knowledge directly from classes, in such a way that the expert can easily understand the main characteristics of the obtained unsupervised classification.

4. Contribute to clustering validation.

5. Contribute to the construction of integral KDD system, as defined by Fayyad, where production of explicit knowledge is as important as the analysis in itself.

6. Consolidate a methodology that enables the automatic generation of characterizers and conceptual interpretations in very complex domains, where from a knowledge base and a previous reference partition (necessarily obtained by a hierarchical structure clustering), a conceptual interpretation of classes is automaticaly generated, using the expert-recommended variables. This system will allow, to establish the corresponding class of a new object and to generate the characterization and conceptual interpretation that corresponds to that object.

2.2 Particular objectives

1. Contribute to an automatic process of interpretation of classes following the methodology presented in [3].

2. Prepare the system to easily include methods for knowledge integration, as well as implement some of these methods, as the Best Local and no Close-World Assumption simple and the Best Local and no Close-World Assumption.

3. Implement the corresponding methods to the different evaluation criteria of rules, both the criteria for one rule and the criteria for a system of rules.

4. Incorporate a new functionality to analyze the quality of a knowledge base, using for that the evaluation criteria mentioned above.

5. Apply the automatic interpretation to a case study within the medical domain using the different knowledge integration methods and compare the results with descriptions from the experts.

Chapter 3

State of the Art

3.1 Intelligent Decision Support Systems

Decision Support Systems (DSS) are a specific class of computerized information systems that supports business and organizational decision-making activities. A properly-designed DSS is an interactive software-based system intended to help decision makers compile useful information from raw data, documents, personal knowledge, and/or business models to identify and solve problems and make decisions.

Although these systems have an excellent functionality, they have also an important problem related to their design, since it requires a considerable experience about exploitation questions and a very complex human analysis to optimize operation times. Experience has revealed that there is a kind of domains with a particularly complex structure where construction of DSS is especially difficult.

An *Intelligent Decision Support System (IDSS)* [51] uses a combination of models, analytical techniques of information recovery and the necessary knowledge about the concrete domain to help to develop and evaluate adequate alternatives [1]. These systems are focused on strategic and non-operational decisions and are oriented to reduce the necessary time to take decisions in a domain, as well as to improve the coherence and quality of decisions [38].

In these types of critical domains where wrong decisions can have disastrous consequences of social, economic and ecologic type (as for example medical and environmental domains), the IDSS-aided decision-making process should be collaborative and not contradictory, and the decision-making should inform and involve those that should live with the decisions and their consequences.

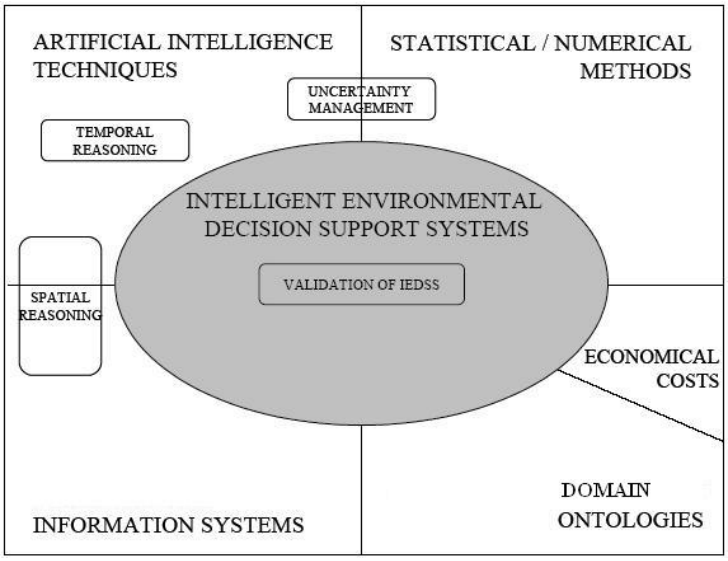

Figure 3.1: Components of an IDSS

In [6] is proposed a structure of 5 levels or layers for the IDSS:

1. The first level includes tasks involved in data collection and its storage in databases. Original data is often *defective*, what it makes necessary a series of processes for data pre-processing before it can be stored in a understandable way and be interpreted. Missing data and uncertainty are factors that should be also considered at this level. Specific data analysis techniques also match with this level.

2. The diagnostic level includes the reasoning models used to infer the *process situation* in order to have a reasonable proposal for action. This can be achieved with the help of statistical, numerical and from Artificial Intelligence models that will use the knowledge acquired in the previous level.

3. The decision support level implies the compilation and fusion of conclusions derived from the AI knowledge models and from statistical models. This level also raises the users' interaction with computer through an interactive system. When a clear and unique conclusion cannot be reached it should be presented to the user a set of ordered decisions according to the likelihood of success or the certainty grade or any other relevant criterion (usefulness, cut of error, etc.).

4. In the fourth level plans are formulated and it is presented to the user a general list of actions or strategies suggested to solve a specific problem.

5. The set of actions that are carried out to solve the problem(s) in the domain to consider is the fifth level. The system recommends not only the action,

or sequence of actions (a plan), but also a solution that must to be accepted by the person in charge of making decisions.

An IDSS not only contributes as an efficient mechanism to find and optimal or sub-optimal solution, given any set of preferences, but also as a mechanism to do all the process more open and transparent. In this context, an IDSS can be a key part in interaction of human beings and systems, as they are tools designed for facing the multidisciplinary and high complexity nature of problems. From a functional point of view, and taking into account the type of problem that the IDSS solves, there are two types of IDSS to be distinguished:

1. IDSS of *control-supervision* of a real-time process (or almost real-time process). It must guarantee robustness against noise, missing of data, typographical mistakes faced with any combination of input data. In general, the final user is responsible of accept-refine-reject the proposed solutions by the IDSS. This can reduce the user responsibility (therefore, there is an increment of confidence in the IDSS) throughout time as the system faces up to situations solved in past (validation).

2. IDSS that give support to specific decisions making. They are used mainly to justify multi-criteria decisions (to formulate transparent to users policies) more than to take daily decisions. This is interesting for final user because it gives the possibility of playing with possible scenarios, to explore the answers and stability of the solution (how sensitive our decision is to little variations in weight and value of variables), etc. Confidence does not increase according to results in front of similar situations, because these IDSS are very specific and, sometimes, they are only built to take or justify one decision.

3.2 Knowledge Discovery from Data

Fayyad defines a *Kowledge Discovery of Data* process as the *overall process of finding and interpreting patterns from data, typically interactive and iterative, involving repeated application of specific data mining methods or algorithms and the interpretation of th patterns generated by these algorithms*[11].

Obtener conocimiento de conjuntos de datos grandes o To obtain knowledge from large or small data sets, and even more, with no structure, is a very difficult task. The combination of techniques for multivariant analysis of data (i.e. clustering), inductive learning (i.e. knowledge-based systems), database management and multidimensional graphic representation must generate some benefit in this direction and in the short term. The Figure 3.2 shows a diagram of the KDD process.

These are the different stages of a KDD process: [12], [11]):

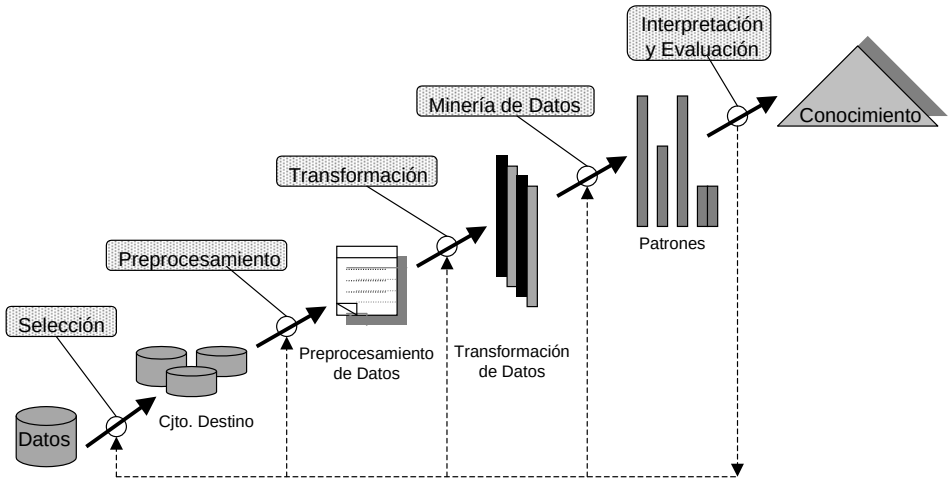

Figure 3.2: Diagram of KDD process

1. Understanding of the application domain, relevant knowledge and final user's goals.

2. Creation of an objective data set. Select a data set, or select an attributes subset or data sample over which do the analysis.

3. Preparation and preprocessing of data. Basic operations, if needed, as noise elimination, treatment of atypical (outliers) or missing data, etc.

4. Reduction and projection of data. Finding relevant characteristics to represent the data depends on the process goals. Use techniques for reduction of dimensionality or methods for transformation of variables to reduce the number of attributes under consideration or to find invariant representations for data.

5. Select the concrete data mining method in order to do the analysis. Depending on the goal of the KDD process, will be appropriate to treat the data with classification techniques of classification, regression, clustering, optimization, inductive reasoning, etc.

6. Select the data mining algorithm(s). Select the technique to be used at the research. This includes to decide the appropriate models and parameters and to choose a data mining method compatible with the KDD process criterion.

7. Data mining. Knowledge discovery of patterns in a formal representation or in a set of representations as: classification rules or trees, regression, clustering and so on. The user can support the data mining method by doing correctly the previous steps.

8. Interpretation of results, possible return to any previous step from 1 to 7 to following iterations.

9. Discovered knowledge consolidation. Incorporation of this knowledge to the system, or just documentation and report to the interested parts.

The KDD process is interactive and iterative and some authors emphasize especially in the interactive nature of the process [4]. It involves complex decisions and choices between the different steps of the process.

Fayyad [11] also points that the KDD process can include significant *interactions* and contain cycles between any two steps; so in every step the data miner can return to the required step to continue the work. The step where data exploitation is really done and where is done the core of knowledge discovery is known as Data Mining.

Depending on the KDD process goal, Data Mining techniques can be very different and change from the simple description of the domain with understanding purposes to the modeling with predictive purposes in all its complexity. We also have to say that there exist different commercial computing tools that deal with some of the mentioned situations (i.e. Clementine, Intelligent Manager, SPAD [44], SPSS [57], WEKA [40], DAVIS [39] are some of the most famous nowadays), which mainly present a combination of the existing techniques, allowing comparison of results.

3.3 Clustering

Cluster analysis or clustering is the assignment of a set of observations into subsets (called clusters) so that observations in the same cluster are similar in some sense. Clustering is a method of unsupervised learning, and a common technique for statistical data analysis used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics.

Besides the term clustering, there are a number of terms with similar meanings, including automatic classification, numerical taxonomy, botryology and typological analysis.

The key is to choose the best classification among all the possible ones that can be built over a set of objects, according to a certain criterion. Essentially, a clustering process could be formulated as the construction of all the possible classifications, the evaluation of some quality criterion over each of them and the selection of that partition that maximize it. Obviously, we are faced with

a NP-complete problem. For this reason clustering methods basically consist of definition of different heuristics to refine the search and avoiding building the whole search space of prohibitive dimensions. The nature of the heuristic and the criterion that allows comparison between different classifications is what changes from one method to other and according to X its nature is more statistical in statistic methods and more logical in those from Artificial Intelligence.

3.3.1 Hierrarchical clustering

Hierarchical clustering is an exploratory tool designed to reveal the natural groupings (or clusters) in a data set that would not be evident in other way. The objects in the hierarchical clustering analysis can be cases or variables, depending on if we want to classify cases or study relations between variables.

Hierarchical clustering creates a hierarchy of clusters which may be represented in a tree structure called a dendrogram. The root of the tree consists of a single cluster containing all observations, and the leaves correspond to individual observations.

Algorithms for hierarchical clustering are generally either *agglomerative*, in which one starts at the leaves and successively merges clusters together; *or divisive*, in which one starts at the root and recursively splits the clusters.

Any valid metric may be used as a measure of similarity between pairs of observations. The choice of which clusters to merge or split is determined by a linkage criteria, which is a function of the pairwise distances between observations.

Cutting the tree at a given height will give a clustering at a selected precision.

3.3.2 Clustering based on rules

The basic idea of *Clustering based on rules* is to collect the information from a Knowledge base and use it in the clustering process in a cooperative way. This knowledge is collected in rules that divide the classification space in consistent environment, so the final proposed classification must respect this first structuring that has been suggested directly by the expert. The idea is to cover three objectives: to incorporate of unknown information (as relations between attributes or restrictions), to recover of the classification goals and to guarantee the interpretability of the obtained classification [21].

In this context, the knowledge provided by the expert is formalized in a set of declarative restrictions that the final structure proposed for the domain must satisfy ($\mathcal{R}$). These restrictions will be used to induce a first *super-structure* of the domain that, even being partial, will guide all the process. The approximation to the *based on rules classification* is based on doing internal classifications, respecting the user's restrictions that can be based on arguments of a semantic nature.

Depending on the nature of the knowledge base provided by the expert $\mathcal{R}$),this process will be unsupervised (as clustering, $\mathcal{R} = \emptyset$) or supervised (as classification, $\mathcal{R}$). The method makes possible to work in any intermediate situation that use a *partial* knowledge base, what is in the context of semi-supervised methods.

Classification based on rules consists of, given a set $\mathcal{I} = \{i_1 \dots i_n\}$:

1. Build the initial knowledge base: The main goal is to make possible the introduction of knowledge from the domain, in restrictions form, to the formation of classes (basically it would lie in materializing in this knowledge base the things that can be and the things that can not be); the expert provides this knowledge in a declarative way, what results in an initial set of logical rules, given $\mathcal{R}^0$.

- Initiate the iterative process ($\xi = 1$):

2. Phase of the a priori knowledge process:

   a) Determine the partition of $\mathcal{I}$ induced by the rules: $\mathcal{P}^{\xi}_{\mathcal{R}}$ from $\mathcal{R}^{\xi}$. Include a residual class $\mathcal{C}^{\xi}_0$ in $\mathcal{P}^{\xi}_{\mathcal{R}}$ with the objects for those was provided an inconsistent knowledge or was not provided any.

   b) Phase of conflicts resolution: Analyze the objects of $\mathcal{C}^{\xi}_0$ selected by contradictory rules:

      i. If it is satisfactory, go to the classification phase.
      ii. If not, return to the construction of $\mathcal{R}^{\xi}$ and reformulate it.

3. Classification phase:

   a) Classification *intra* restrictions of the expert: $\mathcal{P}^{\xi}_{\mathcal{R}}$ will satisfy a priori the expert's requirements. Make the classification for each $\mathcal{C} \in \mathcal{P}^{\xi}_{\mathcal{R}}$. Notice that classes $\mathcal{C} \subset \mathcal{I}$ what will reduce the price of building the classes. Determine:

      i. The corresponding hierarchical trees (*dendograms*) $\tau^{\xi}_{\mathcal{C}}$,
      ii. Their prototypes $\bar{i}^{\xi}_{\mathcal{C}}$, through the summarization of the class,
      iii. Their masses $m^{\xi}_{\mathcal{C}} = card\ \mathcal{C}$ and
      iv. Their level indexes $h^{\xi}_{\mathcal{C}}$.

4. Integration phase:

   a) Extend the residual class: Add the prototypes $\bar{i}^{\xi}_{\mathcal{C}}$ to the residual class $\mathcal{C}^{\xi}_0$, as they were normal objects but taking into account the respective masses. The new data set is the known *extended residual class* $\tilde{\mathcal{I}}^{\xi}$:

$$\tilde{\mathcal{I}}^{\xi} = \left\{(\bar{i}^{\xi}_{\mathcal{C}}, m^{\xi}_{\mathcal{C}}) : \mathcal{C} \in \mathcal{P}^{\xi}_{\mathcal{R}}\right\} \cup \left\{(i, 1) : i \in \mathcal{C}^{\xi}_0\right\}$$

b) Do the integration: Classify $\tilde{\mathcal{I}}^{\xi}$ to integrate all the objects in one only hierarchy, recovering the hierarchical structure of the prototypes $\bar{i}_{\mathcal{C}}^{\xi}$ previously calculated $\tau_{\mathcal{C}}^{\xi}$, $(\mathcal{C} \in \mathcal{P}_R{}^{\xi})$) and lowering them from their root at level $h_{\mathcal{C}}^{\xi}$ in the global hierarchy. This gives raise to the hierarchy τ^{ξ}.

c) Determine the final number of classes: Analyze the dendogram τ^{ξ} para elegir el mejor corte horizontal,to choose the best horizontal cut, using heuristic criteria (manual or automatic ones) [21]. Do the cut of τ^{ξ} identifies a partition of the data in a set of classes, $\mathcal{P}^{\xi}$.Among de k best cuts (being k small), choose the one that makes possible a best interpretation.

5. Evaluation phase: The expert must also confirm that the partition $\mathcal{P}^{\xi}$ obtained with $\mathcal{R}^{\xi}$ improves the partition $\mathcal{P}^{\xi-1}$ that was obtained with $\mathcal{R}^{\xi-1}$ in the desired way. To that end it can be analyzed what terms contribute more to the differences between them or it can be compared different classifications through tables; it is even possible to prove the meaning of these differences, using a non-parametric test (δ-test) designed to this end and presented in [21]. This step can cause the ending criterion of the process:

   a) If the improvement is not significant, stop the iteration and assume the results of the last iteration as the best ones.
   b) If not, analyze the results to reformulate the knowledge base. Build $\mathcal{R}^{\xi+1}$,increase $(\xi = \xi + 1)$ and repeat.

Chapter 4

Basic Concepts

4.1 General Notation

Let $\mathcal{I} = \{i_1, \ldots, i_n\}$ be a set of individuals or objects that is defined by some qualitative and/or quantitative attributes $X_1 \ldots X_K$, whose values for each of the individuals $i \in \mathcal{I}$ are represented by a square matrix $\mathcal{X}$ with dimension (n, K), as it is shown in Table 4.1:

$$\mathcal{X} = \begin{pmatrix} x_{11} & x_{12} & \ldots & x_{1k-1} & x_{1k} \\ x_{21} & x_{22} & \ldots & x_{2k-1} & x_{2k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n-11} & x_{n-12} & \ldots & x_{n-1k-1} & x_{n-1k} \\ x_{n1} & x_{n2} & \ldots & x_{nk-1} & x_{nk} \end{pmatrix}$$

Table 4.1: Data matrix $\mathcal{X}$

Where x_{ik} with $1 \leq i \leq n$ and $1 \leq k \leq K$, is the value that the i-th individual gets for the k-th attribute; that is, the rows of the data matrix $\mathcal{X}$ have information related to the individual characteristics, which can be represented as a vector of attributes in the way:

$$x_i = \begin{pmatrix} x_{i1} & x_{i2} & \ldots & \ldots & x_{ik} \end{pmatrix}$$

And the column are related to the K attributes X_K.

Let $\mathcal{P}_\xi = \{C_1, ..., C_\xi\}$,be a partition in ξ classes of $\mathcal{I}$ and $\mathcal{P}_2 = \{C_1, C_2\}$, a binary partition of $\mathcal{I}$.

Let $\tau = \{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4, ..., \mathcal{P}_n\}$, be an indexed hierarchy over $\mathcal{I}$. It is important to point that given $\mathcal{P}_\xi \in \tau$ and $\mathcal{P}_{\xi+1} \in \tau$, between both there is always one class, and only one, that is subdivided exactly in two classes.

4.2 Dendogram

The hierarchy clustering process is illustrated by a dendogram (from Greek *Dendron* "tree", *-gramma* "drawing") that is a binary tree that organizes the data in subgroups joined in two by two until get the desired grouping level. In each step the two closest classes are joined and represented with a new node. The high of the new internal node of the tree is given by the distance between the two classes (this distance is larger and larger when moving forward the top of the tree).

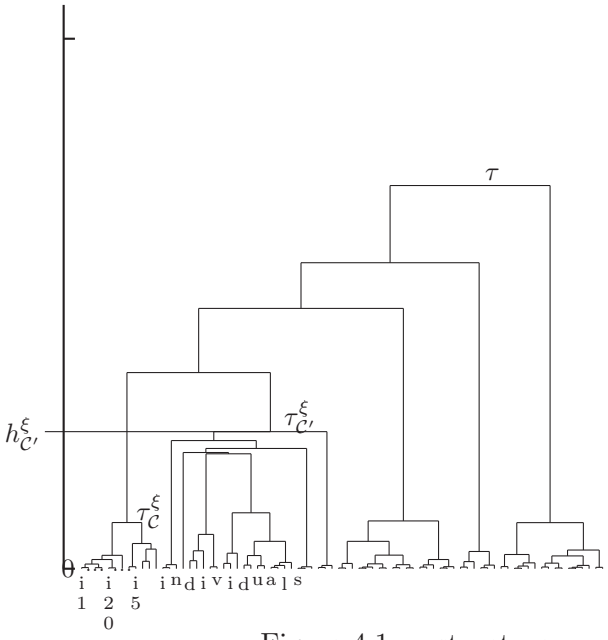

Figure 4.1: τ structure

A dendogram or hierarchical tree consists of:

- A leaf for each element in $\mathcal{I} = \{i_1, \ldots, i_n\}$,
- The internal nodes representing the different established sub groupings over the elements,
- The leaves of the respective subtree to every internal node are the elements that are part of each subclass,
- The branches of different length set the internal nodes in different levels in relation to the horizontal over elements are set out. The level of the nodes, normally coded and known as level index, indicates the similarity degree of their children, and is directly related to the distance between them and the variables space. The larger is the length of the branch that links two children with their father, the less similar are the subclasses represented by these nodes.

When cutting the tree to a determinate horizontal level, a partition $\mathcal{P}_\xi$ of $\mathcal{I}$. is defined. Modifying the level of the cut we obtain different partitions. With this, we get partitions of $\mathcal{I}$ with different degrees of abstraction and it could be chosen the one that fits the most to the user purposes. The definitive partition will be found in a later study of this tree.

The Figure 4.1 illustrates an example of hierarchical tree. The dendogram representation, as can be seen, provides more information that representation with successive partitions, as it indicates which relation of similarity exists between the nodes joined between a partition $\mathcal{P}_\xi$ and the next $\mathcal{P}_\xi + 1$.

Cut criteria: With respect to determine the adequate level of the cut, in some proves are purposed to identify the appropriate level of cut in a hierarchy. However, [58] and [43] are inclined to the construction of a graph to visualize the evolution of the level indexes of successive groupings. Marked discontinuities in this graph are interpreted as *forced* groupings of different classes (that generate large increases of the intra-class inertia). It is recommended to cut in one of these jumps as long as the resultant partition admits the interpretation. These jumps match with the discontinuity in the quotient between the variance between classes and the intra that optimizes in turn a homogeneity criterion of the classes and the separation capability between them. In fact, the graph just illustrates a phenomenon that is also perceptible, in a more veiled way, in the dendogram. The figure 4.2 shows the corresponding graph to a clustering process.

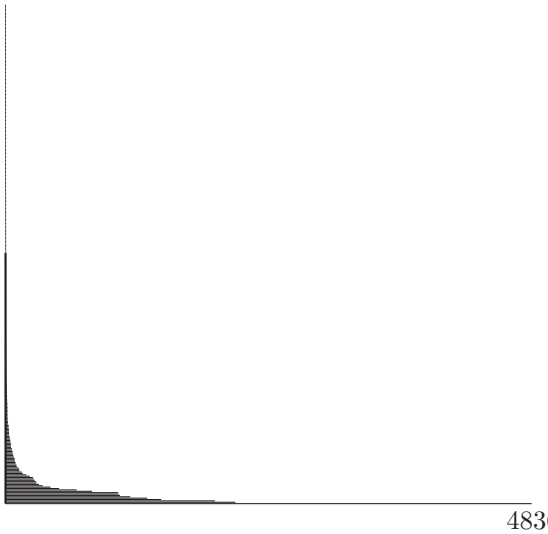

Figure 4.2: Graphic of internal inertia of the classes $[\tau^{EnW,G}_{Lj3,R2}]$

On the other hand, *KLASS* implements a heuristic to determine the level of the cut in an automatic way [15]. The heuristic is very simple and consists of starting from a list with the level indexes of each node, in descending order, then it is calculated the difference between the level index of one node and the

followings and these differences are ordered in a descendent way, associating to each one the number of classes of the cut that determines and, finally, the cut is done for the largest increase of inertia, but immediate successors are indicated to allow the user to chose another one if necessary. This is an effective way of identifying the point where the quotient of inertias is larger.

4.3 Boxplot

4.3.1 Simple Boxplot

The *simple boxplot*, Figure 4.3, is a graphic tool introduced by [53] that works as follows: the interval of values that the variable takes is visualized and the atypical observations (outliers) are marked with "*". A box is then spread from Q_1 (first quartile) until Q_3 (third quartile) and the median is marked with an horizontal sign to the center of the box. Boxes include, then, the 50% of the elements and the whiskers are spread until the minimum and the maximum value that the variable takes.

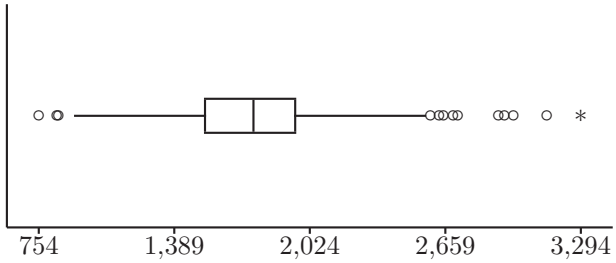

Figure 4.3: Boxplot of the variable MLSS-B

This is a graphical tool that summarizes the sufficient information about the variable distribution.

4.3.2 Multiple Boxplot

The *multiple boxplot*, Figure 4.4, visualizes distributions of a numerical variable conditioned by a set of groups (or classes) and, consequently, allows to analyze the relationship between them.

For each class, the boxplot of the numerical variable is represented for the interval of values in this class according to the introduced in §4.3.1. The boxplots of each group are juxtaposed with a common axis that keeps the same grade and allows comparisons. Juxtaposition can be displayed vertically or horizontally.

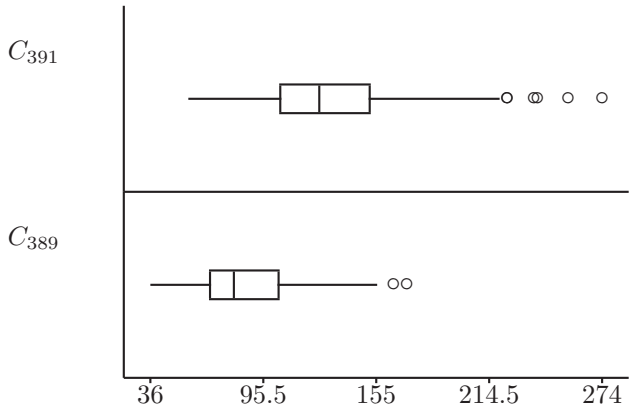

Figure 4.4: Multiple boxplot of the variable DBO-D vs $\mathcal{P}_2 = \{C_{391}, C_{392}\}$

In our case, these graphics are used to visualize the distribution of a numerical variable related with all the classes of $\mathcal{P}_\xi$ (see figure 4.4). According to the ideas introduced by Turkey in the field of descriptive analysis, it started to see the graphical representation of these conditioned distributions and it was used to extract all the relevant information about the problem. This is the way as the *multiple boxplot* fundamental in the process of identify, first visually, *chracaterizing* variables of a class and then define how to calculate them. [19].

4.4 Characterizing variables and values

The characterizing variables were used, first of all, to define a first characterization process to detect minimum sets of variables to distinguish one class from another using just qualitative variables. This is related to the study of how classes interact.

In a second stage, variables are considered in their natural state, avoiding any arbitrary transformation over their nature that could modify the interaction sense between classes.

The numerical variables are managed by identifying all the interactions between the values of variables and the different classes by identifying the values of the variables where the arity of the overlapping classes change, so is possible to identify the different combinations of classes where the same value of a certain variable can appear and, in consequence, the characterizing values of a class emmerge.

In [18] the defined concepts are formal and general and a characterizing method based on finding the typical values of classes is proposed.

Calculus of typical values of a class rests necessary in the distributions of each variable conditioned to classes.

At first the multiple boxplot was analyzed in a visual way, as it is very easy

to see if the boxplot of a certain class does not intersect with the rest, see Figure 4.5, and graphically it is very easy to see the typical values. However, in practice it is not possible to have an automatic process based on the interpretation of a graphical representation, so this ideas evolved till a formal characterization of some basic concepts of different types of, what is named, *characterizing values* and *characterizing variables*.

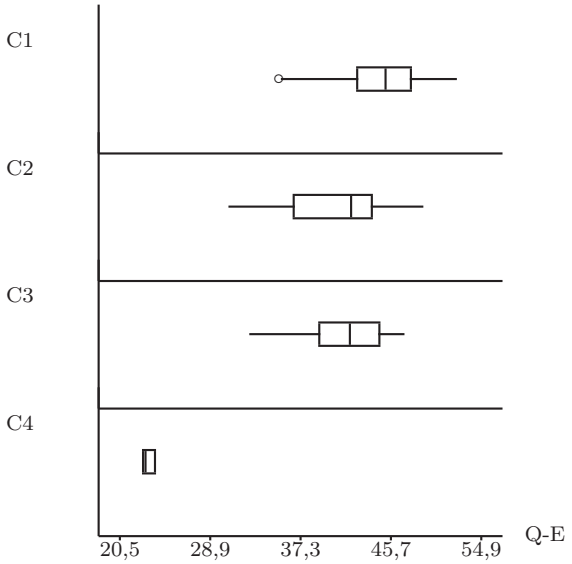

Figure 4.5: Multiple Boxplot of the variable Q_E *vs* partition in 4 classes

The idea is to identify the *characterizing variables* of the class C [26], concept that is based on the concept of *eigenvalue* of a class C. In [19] is presented the definitive formulation of these concepts:

1. Eigenvalue: A value $c_s^k \in \mathcal{D}_k$ is an *eigenvalue* of the class C, if it fulfils:

$$(\exists i \in C : x_{ik} = c_s^k) \wedge (\forall i \notin C : x_{ik} \neq c_s^k)$$

   They are values that appear exclusively in a class. These values, when appear, identify a class with all security, so they act as *chracterizing values* of C and we represent them by λ_{sc}^k.

   We call Λ_C^k to the set of *eigenvalues* of the variable X_K for the class C.

2. Characterizing value: $\lambda \in \Lambda_C^k$, and λ can be used to identify the whole class or a part of it, it depends if there exist other values of X_K in C. There are four types of values, see Table 4.2, the last two types stated in posterior works [56].

   a) λ is a *partially characterizing* value de C if $\{i \in C : x_{ik} = \lambda\} \subset C$. Let V_C^k be the set of *partially characterizing values of* C. It happens exclusively in a class but does not cover it.

b) λ is a *totally characterizing* value of C if $\{i \in C : x_{ik} = \lambda\} = C$. It happens exclusively in a class and covers it completely.

c) A *non typical characterizing* value of C is the one that happens in a class and covers it completely but it is not exclusive of this class.

d) A *generic* value of the class C happens in the class, it does not cover it and it is not exclusive of the class.

	Type of Value	Coverage of the Class	
		Covers all C	Covers part of C
Interaction with other classes	Exclusive	*Total chracterizing*	*Partial characterizing*
	Non-exclusive	*Non-typical characterizing*	*Generic*

Table 4.2: Characterizing values

3. Characterizing variables: According to [15], *characterizing variables*,are *"the ones have been the most decisive result in the creation of these classes and, eventually, allow detection of the membership of an object to a specific class, excluding it from the rest."*, see Table 4.3.

a) Partially characterizing variable: X_K is *partially characterizing* of the class $C \in \mathcal{P}$ if it has at least one typical value of the class C, $(\Lambda_C^k \neq \emptyset)$ and $(V_C^k \neq \Lambda_C^k)$,although it can share some value with other class(es).

b) Totally characterizing variable: X_K is *totally characterizing*of the class $C \in \mathcal{P}$, if all the values that X_K takes in C are *typical* of C, that is, there are not objects of other classes that take these values. Let be $Ext(\Lambda_C^k) = \{i \in \mathcal{I} \ tq \ x_{ik} \in \Lambda_C^k\}$, if $Ext(\Lambda_C^k) = C$, X_K is totally characterizing of C.

	Type of Value	Coverage of the Class	
		Covers all C	Covers part of C
Interaction with other classes	Exclusiv	$Ext(\Lambda_C^k) = C$	$\Lambda_C^k \neq \emptyset \wedge V_C^k \neq \Lambda_C^k$
	Non-exclusive	$Ext(\Lambda_C^k) \supsetneq C \wedge$ $card(\Lambda_C^k) = 1$	$Ext(\Lambda_C^k) \not\supseteq C \wedge$ $card(\Lambda_C^k) > 1$

Table 4.3: Characterizing variables

4. Characterizing grade: Let $(1-\varepsilon)$, $\varepsilon \in [0,1]$ the characterization grade of a class C, for a value. Given a variable X_K,

A *$(1-\varepsilon)$—characterizing* value of C is that *typical value* of C that only identifies $(1-\varepsilon)\%$ of C.
$Card(Ext(\Lambda_C^k)) = (1-\varepsilon)Card(C)$

With all this information in [3] the following relation is established between the belongness of an object i to a class C and its values in X_K, see Table 4.4.

	Type of value	Coverage of the Class	
		Covers all C	Covers part of C
Interaction with other classes	Exclusive	$i = c_s^k \Leftrightarrow i \in C$	$i = c_s^k \Rightarrow i \in C$ $i = c_s^k \not\Leftarrow i \in C$
	Non-exclusive	$i = c_s^k \Leftarrow i \in C$ $i = c_s^k \not\Rightarrow i \in C$	$i = c_s^k \not\Leftrightarrow i \in C$

Table 4.4: relation of characterizing values and a class C

As usually few *totally characterizing* variables, are found, in strict sense, and *partially characterizing* [56]. values are the common ones. That is, values that determine part of a class, the one has to be quantified in order to determine the characterization power of those values. There is not guarantee of finding typical values in a any class, so other proposals have to be considered when we are in such case.

4.5 Boxplot-based discretization

The *Boxplot based discretization (BbD)* is presented in Gibert and Pérez-Bonilla (2006) as an efficient way of transforming a numerical variable into a qualitative one in such a way that the cut points for discretizing identify where the set of classes with non-null intersection of X_k changes and it consists:

1. Calculate de *minimum* (m_C^k) and *maximum* (M_C^k) of X_k inside any class. Built $\mathcal{M}^k = \{m_{C_1}^k, \ldots, m_{C_\xi}^k, M_{C_1}^k, \ldots, M_{C_\xi}^k\}$, where $card(\mathcal{M}^k) = 2\xi$

2. Build the *set of cutpoints* $\mathcal{Z}^k$ by sorting $\mathcal{M}^k$ in increasing way into $\mathcal{Z}^k = \{z_i^k \ ; \ i = 1, \ldots, 2\xi\}$. At every z_i^k the set of intersecting classes changes.

3. Built the *set of intervals I^k induced by $\mathcal{P}$ on X_k* by defining an interval I_s^k between every pair of consecutive values of $\mathcal{Z}^k$. $I^k = \{I_1^k, \ldots, I_{2\xi-1}^k\}$ is the *BbD* of X_k. The I_s^k intervals have variable length and the set of intersecting classes is constant all along the interval and changes from one to another.

In Vázquez and Gibert (2001) [54] there is a proposal of building all the I_s^k following a single pattern: $I_s^k = (z_s^k, z_{s+1}^k] \ \forall s > 1$ being $I_1^k = [z_1^k, z_2^k]$.

In Gibert and Pérez-Bonilla (2005) [29] a deeper discussion about the correct way to build the intervals when the reference partition has two classes is presented. Through a case-analysis it is seen that only two patterns of intervals are suitable in this situation, which are called closed-center pattern and open-center pattern, depending if the central interval is an open interval (on the two sides) or a closed one. For the case of two classes in the reference partition:

- If $(M^k_{C_j} < m^k_{C_i})$ or $(M^k_{C_i} < m^k_{C_j})$ then generate an **open-center pattern** $\mathcal{D}^k$:
  $I^{k,\xi}_1 = [z^k_1, z^k_2]$
  $I^{k,\xi}_2 = (z^k_2, z^k_3)$
  $I^{k,\xi}_3 = [z^k_3, z^k_4]$

- If not, generate a **closed-center pattern** $\mathcal{D}^k$:
  $I^{k,\xi}_1 = [z^k_1, z^k_2)$
  $I^{k,\xi}_2 = [z^k_2, z^k_3]$
  $I^{k,\xi}_3 = (z^k_3, z^k_4]$

4.6 Boxplot-based induction rules

In [21] is described the way of characterize a classification using representatives of class from qualitative variables and the first version about the use of successive conditionings is presented, in that case using a close-world hypothesis and combining negatives in the generated concepts.

Previous versions of BbIR are already in use and is combined the induction of all the numerical variables. Some previous works as [17] and [33] present the general method using numerical variables versus a partition, where any object can be associated with its belonging grade to a class by means of probabilization of the generated rules; and in [32] the method is used also with numerical variables obtaining the identification of totally characterizing variables to get a knowledge base that allows to generate a first interpretation of a 4-classes partition validated by the expert.

Boxplot-based Induction Rules presented in [19], is based on a very simple idea that imitates quite well what experts really do when interpret a multiple boxplot.

In Gibert (2004) [19] the formulation of the methodology *boxplot based induction rules (BbiR)* is presented. It is a method for generating probabilistic concepts with a minimum number of attributes on the basis of the *boxplot based discretization (BbD)* of X_k.. A brief description of the method is the following:

1. For all the numerical variables of $C \in \mathcal{P}_\xi$, obtain with the *BbD* the system of intervals $\mathcal{D}^k = \{I^k_1, \ldots, I^k_{2\xi-1}\}$:

   a) Build $\mathcal{M}^k = \{m^k_{c_1}, \ldots, m^k_{c_\xi}, M^k_{c_1}, \ldots, M^k_{c_\xi}\}$, being the $card(\mathcal{M}^k) = 2\xi$

   b) Build the *set of cut points*
   $\mathcal{Z}^k$ ordering $\mathcal{M}^k$ in ascendent way $\mathcal{Z}^k = \{z^k_i \ ; \ i = 1 : 2\xi\}$, so:
   i) $z^k_1 = min\mathcal{M}^k$
   ii) $z^k_i = min(\mathcal{M}^k \setminus \{z^k_j \ ; \ j < i\}), \ i = \{2, \ldots, 2\xi\}$

$\mathcal{Z}^k = \{z_i^k\}$ is a set that: $\mathcal{Z}^k = \{z_j^k | z_{j-1}^k < z_j^k;\ 1 < j \leq 2\xi\}$

c) Build the *system of intervals* I^k *induced by* $\mathcal{P}_\xi$ *over* X_K, defining the interval I_s^k between each two consecutive values of $\mathcal{Z}^k$ in the following way:

i. If $(M_{C_j}^k < m_{C_i}^k)$ or $(M_{C_i}^k < m_{C_j}^k)$ then generate an open-center pattern $\mathcal{D}^k$ (in the case that the reference partition has two classes and the Pérez-Bonilla and Gibert approach is adopted.):
$I_1^{k,\xi} = [z_1^k, z_2^k]$
$I_2^{k,\xi} = (z_2^k, z_3^k)$
$I_3^{k,\xi} = [z_3^k, z_4^k]$

ii. If not, generate a closed-center pattern $\mathcal{D}^k$ (under the general approach proposed in [54]):
$I_1^{k,\xi} = [z_1^k, z_2^k)$
$I_2^{k,\xi} = [z_2^k, z_3^k]$
$I_3^{k,\xi} = (z_3^k, z_4^k]$

d) Define the new categorical variable I^k of which set of values is $\mathcal{D}^k = \{I_1^k, \ldots, I_{2\xi-1}^k\}$, with *card*$(\mathcal{D}^k) = 2\xi - 1$ y hacer $x_{iI^k} = I_s^k \ \ tq \ \ x_{ik} \in I_s^k$.

2. For all variables:

a) If X_K is a numerical variable, $\mathcal{D}^k = \{I_1^k, \ldots, I_{2\xi-1}^k\}$
build the table of frequencies for the classes conditioned to the intervals:

$\mathcal{I}^k \vert \mathcal{P}_\xi$	C_1	C_2	$\ldots$	C_ξ
I_1^k	p_{11}	p_{12}		
I_2^k				
$\vdots$			p_{sc}	
$I_{2\xi-1}^k$				$p_{(2\xi-1)\xi}$
	1	1		1

donde $p_{sc} = \frac{card\{i : i \in C \wedge x_{ik} \in I_s^k\}}{card\{i : x_{ik} \in I_s^k\}}$

and the total characterizing is such that $p_{sc} = 1$

b) If X_K is a categorical variable, $\mathcal{D}^k = \{I_1^k, \ldots, I_{n_k}^k\}$, where n_k is the number of modalities of the categorical variable X_K and I_s^k will be a modality of X_K.

Build the table of frequencies for the classes conditioned to the categories of the variable:

$\mathcal{I}^k \vert \mathcal{P}_\xi$	C_1	C_2	$\dots$	C_ξ
I_1^k	p_{11}	p_{12}		
I_2^k				
$\vdots$			p_{sc}	
$I_{n_k}^k$				$p_{n_k\xi}$
	1	1		1

where $p_{sc} = \frac{card\{i : i \in C \wedge x_{ik} = I_s^k\}}{card\{i : x_{ik} = I_s^k\}}$

and the total characterizing is such that $p_{sc} = 1$

3. Identify the empiric conditioned frequencies with the certainty grades. This can be represented graphically and can be used as a support tool for the interpretation.

   The figure 4.6shows the belonging grade of a variable X_K in a 4-classes partition.

4. For each non-empty cell of the table $\mathcal{P}_\xi|\mathcal{I}^k$ build a probabilistic rule such as: $\mathcal{R} = \{r : x_{ik} \in I_s^k \xrightarrow{p_{sc}} C : \forall p_{sc} > 0\}$

5. Finally, if crisp decisions are required, decide the uncertainty level α and cut from $\mathcal{R}$ all the rules with an uncertainty grade lower than α to have an automatic interpretation of $\mathcal{P}_\xi$, *at level* α.

to have an automatic interpretation of *characterizing* values [54]. So, we have that:

- A *totally characterizing* value of C is: $I_s^k \ tq \ p_{sc} = 1, \ p_{s'c} = 0, \ \forall \ s' \neq s$.
- A *partially characterizing* value of C is: $I_s^k \ tq \ p_{sc} = 1, \ p_{s'c} > 0, \ \forall \ s' \neq s$.
- A *non-typical characterizing* value of C is: $I_s^k \ tq \ p_{sc} \in (0,1), \ p_{s'c} = 0, \ \forall \ s' \neq s$.
- A *generical value* of the class C is: $I_s^k \ tq \ p_{sc} \in (0,1)$ and $\exists \ s'$ such that $p_{s'c} > 0, \ s' \neq s$ and $\exists \ c'$ such that $p_{sc'} > 0, \quad c' \neq c$. These values can be interpreted as the subset of individuals i of the class C that share their value I_s^k and also with the rest of classes, existing in turn, in the same class C some other elements that belongs to other intervals.

	Type of Value	Coverage of the Class	
		Covers all C	Covers part of C
Interaction with other classes	Exclusive	$I_s^k \ tq \ p_{sc} = 1 \wedge p_{s'c} = 0, \ \forall \ s' \neq s$	$I_s^k \ tq \ p_{sc} = 1 \wedge p_{s'c} > 0, \ \forall \ s' \neq s$
	Non-exclusive	$I_s^k \ tq \ p_{sc} \in (0,1) \wedge p_{s'c} = 0, \ \forall \ s' \neq s$	$I_s^k \ tq \ p_{sc} \in (0,1) \wedge p_{s'c} > 0, \ \forall \ s' \neq s$

Table 4.5: Relation between characterizing values and p_{sc}

Diagram of belonging grades. The idea of associate to any object its belonging grade to each class from the table of frequencies for the classes conditioned to the intervals gives rise to a graphic of belonging grades that could be adapted easily to the diffuse paradigm [33] for each class and for each variable as can be seen in the Figure 4.6 and that leads to elements of approximate reasoning. It could be mentioned that the area under these functions is not 1 any more, because they consist of probabilities from different conditioned distributions (those from $C|\mathcal{I} = I_s^k,\ \forall s$).

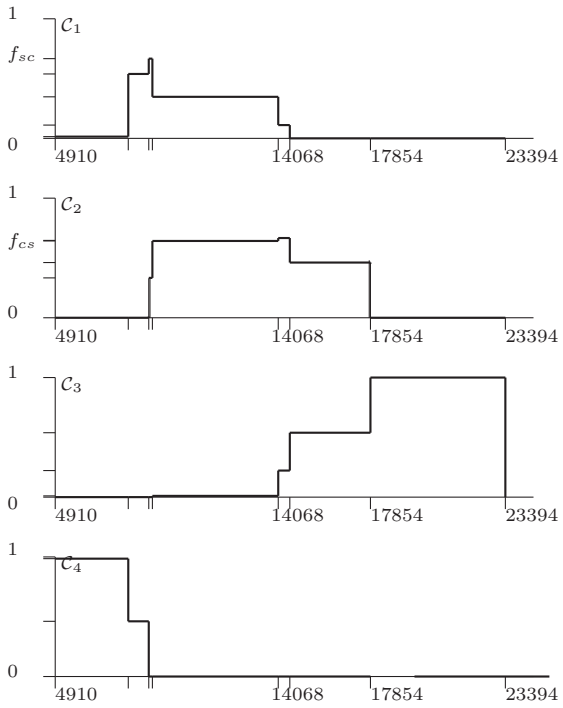

Figure 4.6: Belonging grades of X_K to a partition $\mathcal{P}_4$ in 4 classes.

4.7 Propositional Logics

Propositions A proposition in classical logic, for the purpose of this work, is a declaration that can evaluate to true or false certain state of the universe is modeling, for example: $5 > 4$, $2+2 = 5$, "Peter ate at three", "I like soup". Sometimes is more difficult than others to determine if the declaration (or proposition) is true or false, in other words, if it takes the value of truth or falseness. Propositional logic is a formal language that can be described with a BNF grammar. The Backus-Naur form (BNF) (also known as Backus-Naur formalism, Backus normal form or Panini-Backus Form) is a metasyntax used to express context-free grammars: that is, a formal way to describe formal languages.

Syntaxis and notation

1. Syntax: The first step in the study of a language is to define the basic symbols that constitute it (alphabet) and how can they be combine to form sentences. The theory over a domain in propositional logic consists of:

   - Symbols of veracity: $\top$ for true and $\bot$ for false. Alternatively it can be used V for true and F for false.
   - Symbols of variables: $p, q, \ldots, z$
   - Symbols of connectives: Negation ($\neg$), Conjunction ($\wedge$), Disjunction ($\vee$), Implication ($\longrightarrow$), Coimplicacin ($\longleftrightarrow$).
   - Symbols: brackets (), square brackets [] and brace brackets {} to avoid ambiguities.

2. Formation rules: Classes of sentences well-formed are defined by purely syntactic rules, known as formation rules, they are:

   - A propositional variable is a sentence (also known as formula) well-formed. Well-formed sentences are: $\neg p$, $p \vee q$, $p \wedge q$, $p \longrightarrow q$, $p \longleftrightarrow q$. If p and q are in turn well-formed sentences.
   - In conjunctions and disjunctions more than two arguments can be allowed.

Truth tables Truth tables allow evaluating composed and well-formed sentences from values of the variables they consist of.

p	$\neg p$	p	q	$p \vee q$	$p \wedge q$	$p \longrightarrow q$	$p \longleftrightarrow q$
V	F	V	V	V	V	V	V
F	V	F	V	V	F	V	F
		V	F	V	F	F	F
		F	F	F	F	V	V

Table 4.6: Negation ($\neg$), Disjunction ($\vee$), Conjunction ($\wedge$), Conditional ($\longrightarrow$) and Biconditional ($\longleftrightarrow$)

Morgan's Laws (1806-1871)

1. Morgan's law 1: $\neg(A \wedge B) \equiv \neg A \vee \neg B$
2. Morgan's law 2: $\neg(A \vee B) \equiv \neg A \wedge \neg B$

Axioms and rules Axioms for propositional calculus [10] are:
Let P, Q and R be sentences, then;

1. Idempotence axiom: $(P \vee P) \longrightarrow P$.
2. Adjunction axiom: $P \longrightarrow (P \vee Q)$.
3. Commutativity axiom: $(P \vee Q) \longrightarrow (Q \vee P)$.
4. Addition axiom: $(P \longrightarrow Q) \longrightarrow [(R \vee P) \longrightarrow (R \vee Q)]$

From these axioms and applying the two following transformation rules any theorem can be proved:

1. Substitution rule: the result of replace any variable in a theorem by a well-formed sentence is a theorem.
2. Separation rule: if S and $(S \longrightarrow R)$ are theorems, then R is a theorem.

   Relating to a validation criterion, an axiomatic system should fulfill the next properties to be a perfect system:

   - It should be logical or reasonable, in the sense that every theorem is either an axiom or the last sequence of a deduction that is followed by deductive logical operations in accordance with the specified rules.
   - Complete: every valid well-formed sentence is a theorem and should be proved from axioms.
   - Sound: well-formed sentences that are not tautologies can be proved as theorems.
   - They should be independent: any axiom should be derivable from the others. However, the Gdel's theorem proves that such a perfect system is not possible.

4.8 On the quality of rules

In the presented proposal the best rules are selected at every iteration of the process. The criteria to decide which is the best rule from a Knowledge Base, is a combination of some of the following criteria, some of them used to evaluate the quality of a rule.

In [3], they are formulated in the following way:

4.8.1 Evaluation criteria for one rule

Support (Sup)

Given a rule $r : A_C(i) \xrightarrow{p} C$,the support of r is the proportion of objects in $\mathcal{I}$ that satisfy the antecedent of the rule, [45].

$$Sup(r) = \frac{card\{i \ \in \ \mathcal{I} \ \ tq \ \ A_C(i) = true\}}{n} \tag{4.1}$$

It measures how many times the rule r is activated in the database.

If support reaches the 100% means that all the objects of the database satisfy the rule.

Confidence (p)

Given a rule r, the confidence of r is the proportion of objects of antecedent $(A_C(i) = true)$ that are in C, $\forall \ C \in \mathcal{P}_\xi$ [45].

$$p(r) = \frac{card\{i \ \in \ C \ \ tq \ \ A_C(i) = true\}}{card\{A_C(i) = true\}} \tag{4.2}$$

where $r : A_C(i) \xrightarrow{p} C$, $A_C(i)$ is true if i satisfies the antecedent $A_C(i)$, whatever the form of the antecedent of the rule (simple or compound).

It allows to measure how many a rule $r : A_C(i) \xrightarrow{p} C$ would be mistaken when assigning an object i to a class C. If $p(r) = 0$ means that it is always mistaken and if $p(r) = 1$ means it is always true.

Relative Covering $(CovR)$

Given a rule $r : A_C(i) \xrightarrow{p} C$, the relative covering is the proportion of objects of class C that satisfy the antecedent of the rule.

$$CovR(r) = \frac{card\{i \ \in \ C \ \ tq \ \ A_C(i) = true\}}{n_c} \tag{4.3}$$

Relative covering measures how many timesthe rule $r : A_C(i) \xrightarrow{p} C$ would be mistaken when describing the class C with the antecedent $A_C(i)$.

4.8.2 Evaluation criteria for a system of rules

Global Covering ($CovG_{lobal}$)

Given a knowledge base that consists of a set of rules of the type $r : A_C(i) \xrightarrow{p} C_j$, the global covering is the proportion of objects of $\mathcal{I}$ that activate correctly the rules of the knowledge base KB^{ξ}.

$$CovG_{lobal}(\mathcal{R}) = \frac{\sum_{\forall C \in \mathcal{P}_{\xi}} card\{i \ \in \ C \ \ tq \ \ A_C(i) = true\} \times n_c}{n} \tag{4.4}$$

Total Support (Sup_T)

It is the total support of the partition that is being interpreted and it is the addition of the support values of each composed rule associated to each one of the classes that forms the final partition. It is frequently used in the literature [45] and represents the percentage of objects that activate some rule of the knowledge base included in the final partition.

$$Sup_T(\mathcal{R}) = \sum_{\forall r \in \mathcal{R}} Sup(r) = \sum_{\forall r \in \mathcal{R}} \frac{card\{i \ \in \ \mathcal{I} \ \ tq \ \ A_C(i) = true\}}{n} \tag{4.5}$$

Mean Confidence ($\overline{p}$)

Mean confidence of the knowledge base of a system of rules $\mathcal{R}(\mathcal{P}_{\xi})$ is the average of the certainty values of each one of the rules [45].

$$\overline{p}(\mathcal{R}) = \frac{\sum_{\forall r \in \mathcal{R}(\mathcal{P}_{\xi})} p(r)}{n_{\mathcal{R}}} = \frac{\sum_{\forall r \in \mathcal{R}(\mathcal{P}_{\xi})} \frac{card\{i \ \in \ C \ \ tq \ \ A_C(i) = true\}}{card\{A_C(i) = true\}}}{n_{\mathcal{R}}} \tag{4.6}$$

Chapter 5

Context of the Research

5.1 The framework project description

This work is within the frame of a research project directed by Dr. Karina Gibert which the objective of developing hybrid support methodologies to the Knowledge Discovery and Data Mining in unstructured domains [24] to solve decision support problems, mainly in medical and environmental domains. This project was started in 1995 with the idea of combining Statistical techniques with the ones from Artificial Intelligence to overcome the limitations of classical techniques I the different steps of the analysis of these kinds of domains [20], [19].

The first proposal constitutes the Karina Gibert's degree thesis [13] and PhD thesis [15] that resulted in the formaulation of the *m*ethodology of classification based on rules and a first version of the computer system that implements it, called *KLASS* [15] and that has been used in different real applications [22, 23, 27, 25, 34, 32].

All the methodologies developed within the framework project are integrated in a master tool, that nowadays is *java.KLASS* [28] and that joins tolls of very different nature, offering the necessary interface to communicate the different modules and transfer the necessary information in each moment of the analysis.

Within this framework, different PhD thesis, master thesis and degree thesis have been developed both in Statistics degree and Computer engineering. Nowadays there is a group of people doing research and working as a team.

Nowadays, the main goals are centered in the development of tools and methodologies for the support to the clustering interpretation and the modeling and conceptualization of dynamic systems in medical and environmental domains.

5.2 KLASS' Chronology

- Feb. 1991 **K**LASS v0. Karina Gibert's dissertation. "KLASS. Study of an assistance system for statistical treatment of large databases". It classifies data matrixes of heterogeneous data with mixed distance. [14]

- Nov. 1994 **K**LASS v1. Karina Gibert's thesis. "Use of symbolic information in the automatization of the statistic treatment of unstructured domains". It is an extension of **K**LASS v0. It includes classification based in rules. [16]

- Jul. 1996 **K**LASS v1.1. PFC Xavier Castillejo. It incorporates to an independent windows interface with a system that enables the use of KLASS from a SUN and from a PC to users that don't know Lisp and UNIX. Let's call **x**cn.KLASS to the Lisp kernel of this new version and **x**cn.i to the C interface. [5]

- Oct. 1997 **jj**.KLASS. PFC Juan Jos Marquez and Juan Carlos Martn. It incorporates to the **K**LASS.v1 version new options for the treatment of missing values, the possibility of working with weighted objects and implements a non-parametric test for the comparison of classifications [46].

- Set. 1999 **K**LASS v1.2. PFC Xavier Tubau (β version). It incorporates to the **x**cn.KLASS version the comparison of classifications module of **jj**.KLASS,the Ralambondrainy's mixed metric [48] [49]and prepares the formulation of three more for their later implementation. Let's call **x**t.KLASS to the Lips kernel of this new version and X to the associated C interface. [52]

- 1999-2000 **K**LASS+ v1. PFC Slvia Bayona. Definitive fusion of versions **x**t.KLASS and **jj**.KLASS. It incorporates a new module of data descriptive analysis, as well as the resultant classes, reorienting KLASS to a more general proposal and less specialized. Let's call **s**bh.KLASS to the Lisp kernel of this new version and **s**bh.i to the associated C interface. [2]

- 2000-2002 **K**LASS+ v2. PFC Josep Oliveras. It incorporates to **sbh.KLASS** the pending mixed metrics (Gower [37] [35] [36], Gowda-Diday [8] [7] and Ichino-Yaguchi [41]). Let's call **joc.KLASS** to this new version. [47]

- 2000-2003 **j**r.KLASS+. Jorge Rodas's thesis. Integrates **KLASS+ v.2** and **Columbus**, that is later presented. [50]

- 2000-2003 Anna Salvador and Fernando Vzquez research. Developement of **CIADEC**, that is later presented. [55]

- 2002-2003 **J**ava-KLASS v0. PFC M del Mar Colillas. Java version of the descriptive analysis module and integration with **CIADEC** and **Columbus**.
  - 2003-2005 **J**ava-KLASS v0.22. In collaboration with Mar Colillas. Extension of the descriptive analysis module and introduction of tools for data management *(definition of orders in the informs, possibility of simultaneously different matrixes of objects in the system, change of active matrix).*
  - 2005-2006 **J**ava-KLASS v1.0. In collaboration with Mar Colillas. It includes the reading and visualization of isolated dendograms, as well as the generation of partitions from them.
- 2006-2007 **J**ava-KLASS v2.0. PFC Jose Ignacio Mateos. Extension of **J**ava-KLASS with a module of calculation of distances for different types of data matrixes, including the ones that combine qualitative and quantitative information, treatment of missing values and creation of submatrixes.
- 2006-2007 **J**ava-KLASS v3.0. PFC Roberto Tuda. It includes a module of automatic classification by hierarchical methods, using all the distances implemented in v2.0 and an option for studying aggregations of objects step by step. The option of selecting the default work directory is created. The option of adding and recording weighted objects is included.
- 2006-2007 **J**ava-KLASS v4.0. PFC Laia Riera Guerra. Introduction, management and evaluation of Knowledge Bases. Extension of **J**ava-KLASS with a module of transformation of variables that allows discretizations, recodifications and arithmetic calculations with numerical variables. Finally, this version includes the definition of submatrixes through logical filters over the objects, the edition of metainformation of the matrix variables, elimination of variables and importation of files in .dat standard format.
- 2007 **J**ava-KLASS v5.0. PFC Andreu Raya. It includes the embedded classification, classification based on rules and functionalities for division of the database and for management of classification trees *(or dendograms)* associated to the different data matrixes.
- 2007 **J**ava-KLASS v6.0. Alejandro Garca's thesis. Exogenous classification based on rules. Internationalization and location to three languages (Catalan, English, Spanish). Matrix merger.
- 2008 **J**ava-KLASS v6.4. master thesis of Alfons Bosch Sansa, Patricia Garca Gimnez, Ismael Sayyad Hernando. Boxplot-based discretization, Boxplot-based Induccion rules.

- 2008. Alejandra Perez's thesis. Characterization by embedded conditionings, methodology that induces automatically associated concepts to the discovered classes.
- 2008. Gustavo Rodriguez's thesis. Classification based on rules by states that allows analysis of dinamical systems.

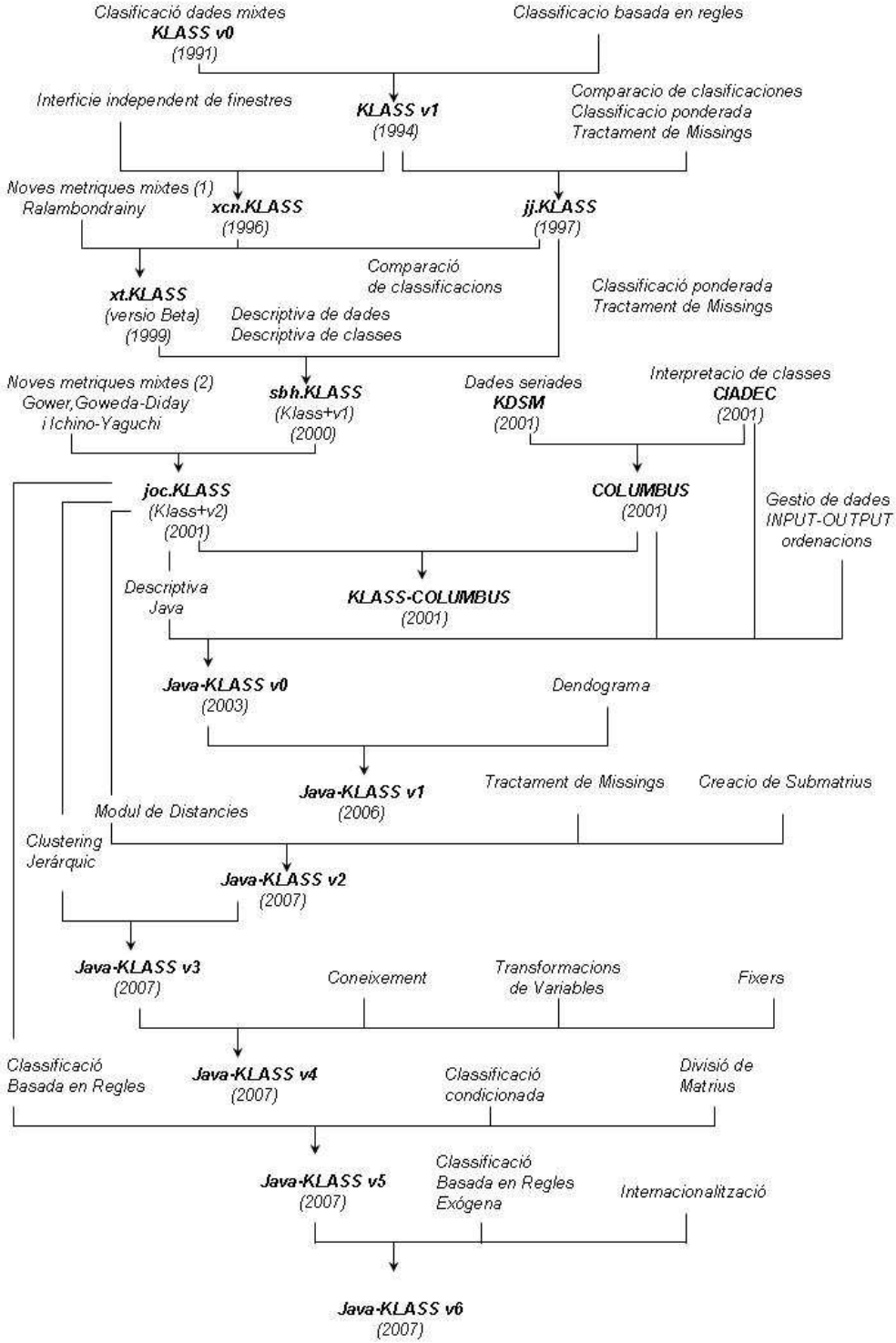

Figure 5.1: KLASS' Chronology

Chapter 6

The CCEC Methodology

6.1 Introduction

In this chapter is introduced the methodology that represents the basis of this project, the *Methodology of conceptual characterization by embedded conditioning CCEC*, oriented to the automatic generation of conceptual descriptions of classifications that can support later decision-making and that was firstly formulated in [3].

6.2 Methodology

CCEC takes advantage of the existence of τ, and uses the property of any binary hierarchical structure that $\mathcal{P}_{\xi+1}$ has the same classes of $\mathcal{P}_{\xi}$ except one, which splits in two subclasses in $\mathcal{P}_{\xi+1}$. Binary hierarchical structure will be used by *CCEC* to discover particularities of the final classes step by step also in hierarchical way. The *CCEC* [30] allows generation of automatic interpretations of a given partition $\mathcal{P} \in \tau$.

1. Cut the tree at highest level (make $\xi = 2$ and consider $\mathcal{P}_2 = \{C_1, C_2\}$).

2. Use for *the boxplot based discretization* presented in [19] and revised in [31], to find (total or partial) characteristic values for numerical variables [21].

3. Use for *boxplot based induction rules (BbIR)*, to generate the knowledge Base for both classes.

4. For classes in $\mathcal{P}_2$, determine concepts A_1^{ξ,X_k} : "$[X_k \in I_s^k]$", $A_2^{\xi,X_k} : \neg A_1^{\xi,X_k}$ associated to C_1, C_2, by taking the intervals provided by a totally characteristic variable or the partial one with greater relative covering and $p_{sc} = 1$.

5. Go down one level in the tree, by making $\xi = \xi + 1$ and so considering $\mathcal{P}^{\xi+1}$. As said before $\mathcal{P}^{\xi+1}$ is *embedded* in $\mathcal{P}^{\xi}$ in such a way that there is a class of $\mathcal{P}^{\xi}$ split in two new classes of $\mathcal{P}^{\xi}$, namely $C_i^{\xi+1,X_k}$ and $C_j^{\xi+1,X_k}$ and all other classes are common to both partitions.

   Since in the previous step $C_i^{\xi+1,X_k} \cup C_j^{\xi+1,X_k}$ were conceptually separated from the rest, at this point it is only required to find the variables which separate (or distinguishes) $C_i^{\xi+1,X_k}$ from $C_j^{\xi+1,X_k}$, by repeating steps 2-4. Suppose $B_i^{\xi+1,X_k}$ and $B_j^{\xi+1,X_k}$ the concepts induced from $C_i^{\xi+1,X_k}$ and $C_j^{\xi+1,X_k}$, in the step $\xi + 1$.

6. Integrate the extracted knowledge of the iteration $\xi + 1$ with that of the iteration ξ, by determining the compound concepts finally associated to the elements of $\mathcal{P}_{\xi+1}$. The concepts for the classes of $\mathcal{P}_{\xi+1}$ will be: $A_q^{\xi+1,X_k} = A_q^{\xi,X_k}$, $A_i^{\xi+1,X_k} = \neg A_q^{\xi,X_k} \wedge B_i^{\xi+1,X_k}$ and $A_j^{\xi+1,X_k} = \neg A_q^{\xi,X_k} \wedge B_j^{\xi+1,X_k}$

7. Make $\xi = \xi + 1$, and return to the step 2) repeating until $\mathcal{P}_{\xi} = \mathcal{P}$.

6.3 Knowledge Integration

6.3.1 Best local concept and no Close-World Assumption

It consists of choosing among all the rules of $\mathcal{S}(\mathcal{P}^*_{\xi+1})$ that go to a same class, the one that has a higher relative coverage and not doing the CWA assumption. In this way, negation of the chosen concept is not used to define the concept of the complementary class, but for each class is used the certain concept with a higher relative coverage.

1. Restrict the search to the best rule of the knowledge base $\mathcal{S}(\mathcal{P}^*_{\xi+1})$ for the restricted partition $P^*_{\xi+1} = \{C_i^{\xi+1}, C_j^{\xi+1}\}$.

2. Consider for each class $C_i^{\xi+1}$ and $C_j^{\xi+1}$ of $\mathcal{P}^*_{\xi+1}$ a subsystem of rules that satisfies the rules pointed to a same class:

   $\mathcal{S}_{C_i}(\mathcal{P}^*_{\xi+1}) = \{r^k_{s,c} : C = C_i \ \wedge \ r^k_{s,c} \in \mathcal{S}(\mathcal{P}^*_{\xi+1})\}$ where $\mathcal{S}_{C_i}(\mathcal{P}^*_{\xi+1}) \subseteq \mathcal{S}(\mathcal{P}^*_{\xi+1})$ and

   $\mathcal{S}_{C_j}(\mathcal{P}^*_{\xi+1}) = \{r^k_{s,c} : C = C_j \ \wedge \ r^k_{s,c} \in \mathcal{S}(\mathcal{P}^*_{\xi+1})\}$ where $\mathcal{S}_{C_j}(\mathcal{P}^*_{\xi+1}) \subseteq \mathcal{S}(\mathcal{P}^*_{\xi+1})$

3. Choose the concept linked to the rule of higher relative coverage of $\mathcal{S}_{C_i}(\mathcal{P}^*_{\xi+1})$ and the one of $\mathcal{S}_{C_j}(\mathcal{P}^*_{\xi+1})$.

   > Determine k_i, s_i such as the concept "$X_{k_i} \in I_{s_i}^{k_i,\xi+1}$" has $p_{s_i c_i} = 1$ and the relative coverage of the rule $r^{k_i}_{s_i,c_i}$ is maximum at $\mathcal{S}_{C_i}(\mathcal{P}^*_{\xi+1})$
   > y k_j, s_j such as the concept "$X_{k_j} \in I_{s_j}^{k_j,\xi+1}$" has $p_{s_j c_j} = 1$ such as

the relative coverage of the rule $r^{k_j}_{s_j,c_j}$ is maximum at $\mathcal{S}_{C_j}(\mathcal{P}^*_{\xi+1})$.
$C_i \neq C_j$ y $C_i, C_j \in \mathcal{P}^*_{\xi+1}$.
Let,

$\mathcal{K}_i = \{k \quad tq \quad$ "$X_k \in I^{k,\xi+1}_s$" corresponds to a certain rule of maximum coverage in $\mathcal{S}_{C_i}(\mathcal{P}^*_{\xi+1})$.

$\mathcal{K}_j = \{k \quad tq \quad$ "$X_k \in I^{k,\xi+1}_s$" corresponds to a certain rule of maximum coverage in $\mathcal{S}_{C_j}(\mathcal{P}^*_{\xi+1})$.

4. Determine concepts $A^{*\xi+1}_i$ and $A^{*\xi+1}_j$ induced for $C^{\xi+1}_i$ and $C^{\xi+1}_j$ in the next way:

   a) If only one rule by class is identified:
   let be

   $$A^{\xi+1,k_i}_i = \text{“}X_{k_i} \in I^{k_i,\xi+1}_{s_i}\text{”} \quad then, \quad A^{*\xi+1}_i = A^{\xi+1,k_i}_i \tag{6.1}$$

   and

   $$A^{\xi+1,k_j}_j = \text{“}X_{k_j} \in I^{k_j,\xi+1}_{s_j}\text{”} \quad then, \quad A^{*\xi+1}_j = A^{\xi+1,k_j}_j \tag{6.2}$$

   For each class:

   b) If there is more than one rule $r^k_{s,c}$ with $p_{sc} = 1$ and maximum relative coverage, all of them are considered in the construction of the concept and it is done in a different way depending on if it is a totally or partially characterizing variable.

   - If X_k is totally characterizing (it generates rules $p_{sc} = 1$ and $CovR = 100\%$). It is built in the following way:

     $$A^{*\xi+1}_i = \bigwedge_{\forall\ k_i \in \mathcal{K}_i} A^{\xi+1,k_i}_i \tag{6.3}$$

     and

     $$A^{*\xi+1}_j = \bigwedge_{\forall\ k_j \in \mathcal{K}_j} A^{\xi+1,k_j}_j \tag{6.4}$$

   - Si X_k is partially characterizing (it generates rules $p_{sc} = 1$ and $CovR < 100\%$). It is built in the following way:

     $$A^{*\xi+1}_i = \bigvee_{\forall\ k_i \in \mathcal{K}_i} A^{\xi+1,k_i}_i \tag{6.5}$$

and

$$A_j^{*\xi+1} = \bigvee_{\forall\ k_j \in \mathcal{K}_j} A_j^{\xi+1,k_j} \tag{6.6}$$

6.3.2 Best local concept and Close-World Assumption

It consists of choosing among all the rules of $\mathcal{S}(\mathcal{P}^*_{\xi+1})$ that go to a same class the one that has a higher relative coverage and doing a strong *Close-World* assumption *CWA*.

In this way, for each class, negation of the chosen concept is used to define the other class in logical disjunction ($\vee$) with the certain concept obtained by the maximum relative coverage.

1. Restrict the search to the best rules of the knowledge base $\mathcal{S}(\mathcal{P}^*_{\xi+1})$ for the restricted partition $P^*_{\xi+1} = \{C_i^{\xi+1}, C_j^{\xi+1}\}$.

2. Consider for each class $C_i^{\xi+1}$ and $C_j^{\xi+1}$ of $\mathcal{P}^*_{\xi+1}$ a subsystem of rules that satisfies the rules pointed to a same class:

   $\mathcal{S}_{C_i}(\mathcal{P}^*_{\xi+1}) \subseteq \mathcal{S}(\mathcal{P}^*_{\xi+1})$ and $\mathcal{S}_{C_j}(\mathcal{P}^*_{\xi+1}) \subseteq \mathcal{S}(\mathcal{P}^*_{\xi+1})$

3. Choose the concept linked to the rule with a higher relative coverage of $\mathcal{S}_{C_i}(\mathcal{P}^*_{\xi+1})$ and the one with $\mathcal{S}_{C_j}(\mathcal{P}^*_{\xi+1})$.

   Determine k_i, s_i such as the concept "$X_{k_i} \in I_{s_i}^{k_i,\xi+1}$" has $p_{s_i c_i} = 1$ and the relative coverage of the rule $r_{s_i,c_i}^{k_i}$ is maximum in $\mathcal{S}_{C_i}(\mathcal{P}^*_{\xi+1})$ and k_j, s_j tales que el concepto "$X_{k_j} \in I_{s_j}^{k_j,\xi+1}$" has $p_{s_j c_j} = 1$ such as the relative coverage of the rule $r_{s_j,c_j}^{k_j}$ is maximum in $\mathcal{S}_{C_j}(\mathcal{P}^*_{\xi+1})$. $C_i \neq C_j$ and $C_i, C_j \in \mathcal{P}^*_{\xi+1}$.

   Let,

   $\mathcal{K}_i = \{k \quad tq \quad$ "$X_k \in I_s^{k,\xi+1}$" corresponds to a certain rule with maximum coverage in $\mathcal{S}_{C_i}(\mathcal{P}^*_{\xi+1})$.

   $\mathcal{K}_j = \{k \quad tq \quad$ "$X_k \in I_s^{k,\xi+1}$" corresponds to a certain rule with maximum coverage in $\mathcal{S}_{C_j}(\mathcal{P}^*_{\xi+1})$.

4. Do a strong hypothesis of Close-World to describe the similar class depending on the complementary concept.

5. Determinar los conceptos $A_i^{*\xi+1}$ y $A_j^{*\xi+1}$ inducidos para $C_i^{\xi+1}$ y $C_j^{\xi+1}$ de la siguiente forma:

a) If only one rule by class is identified:
Let $A_i^{\xi+1,k_i}$ = "$X_{k_i} \in I_{s_i}^{k_i,\xi+1}$" and
$A_j^{\xi+1,k_j}$ = "$X_{k_j} \in I_{s_j}^{k_j,\xi+1}$"

Finally;

$$A_i^{*\xi+1} = A_i^{\xi+1,k_i} \vee \neg A_j^{\xi+1,k_j} \tag{6.7}$$

$$A_j^{*\xi+1} = A_j^{\xi+1,k_j} \vee \neg A_i^{\xi+1,k_i} \tag{6.8}$$

b) If there are more than one rule $r_{s,c}^k$ with $p_{sc} = 1$ and maximum relative coverage, all of them are considered in the construction of the concept and it is done in a different way depending on if it is a totally or partially characterizing variable.

- If X_k is totally characterizing (generates rules $p_{sc} = 1$ and $CovR = 100\%$). It is built in the following way:

$$\begin{aligned} A_i^{*\xi+1} = & \bigwedge_{\forall\ k_i \in \mathcal{K}_i,\ k_j \in \mathcal{K}_j} (A_i^{\xi+1,k_i} \vee \neg A_j^{\xi+1,k_j}) = \\ & (\bigwedge_{\forall\ k_i \in \mathcal{K}_i} A_i^{\xi+1,k_i}) \vee (\bigwedge_{\forall\ k_j \in \mathcal{K}_j} \neg A_j^{\xi+1,k_j}) \end{aligned} \tag{6.9}$$

and

$$\begin{aligned} A_j^{*\xi+1} = & \bigwedge_{\forall\ k_i \in \mathcal{K}_i,\ k_j \in \mathcal{K}_j} (A_j^{\xi+1,k_j} \vee \neg A_i^{\xi+1,k_i}) = \\ & (\bigwedge_{\forall\ k_j \in \mathcal{K}_j} A_j^{\xi+1,k_j}) \vee (\bigwedge_{\forall\ k_i \in \mathcal{K}_i} \neg A_i^{\xi+1,k_i}) \end{aligned} \tag{6.10}$$

- If X_k is partially characterizing (generates rules $p_{sc} = 1$ and $CovR < 100\%$). It is built in the following way:

$$\begin{aligned} A_i^{*\xi+1} = & \bigvee_{\forall\ k_i \in \mathcal{K}_i,\ k_j \in \mathcal{K}_j} (A_i^{\xi+1,k_i} \vee \neg A_j^{\xi+1,k_j}) = \\ & (\bigvee_{\forall\ k_i \in \mathcal{K}_i} A_i^{\xi+1,k_i}) \vee (\bigvee_{\forall\ k_j \in \mathcal{K}_j} \neg A_j^{\xi+1,k_j}) \end{aligned} \tag{6.11}$$

and

$$\begin{aligned} A_j^{*\xi+1} = & \bigvee_{\forall\ k_i \in \mathcal{K}_i,\ k_j \in \mathcal{K}_j} (A_j^{\xi+1,k_j} \vee \neg A_i^{\xi+1,k_i}) = \\ & (\bigvee_{\forall\ k_j \in \mathcal{K}_j} A_j^{\xi+1,k_j}) \vee (\bigvee_{\forall\ k_i \in \mathcal{K}_i} \neg A_i^{\xi+1,k_i}) \end{aligned} \tag{6.12}$$

Chapter 7

Enlarging KLASS with automatic interpretation of classes

7.1 Introduction

As has been defined in §2, in this project have been implemented new functionalities within an existing framework application, in order to extend it with new modules. In chapter §5 there is a slight description of this framework, the KLASS application, and here there is more detailed information about implementation of the system, and more specifically, implementation of the new functionalities.

It is important to remark that maintaining alive an application like KLASS during more than ten years is only possible with a strict methodology for the expansion of new functionalities. This allows the easy incorporation of new modules without affecting the running of the rest application.

7.2 KLASS' structure

In order to understand the implementation details of the next section, is necessary to introduce how KLASS is structured from the imlementation point of view.

KLASS is a Java application that has been implemented following a structure of layers to separate the graphic interface part from the main methods and data objects. KLASS consists of the following packages:

- **jklass.ui:** This package contains the classes related to the graphical user interface. KLASS has a windows interface, and each of these windows is implemented in a specific class. If a new window wants to be added, a new class must be created for the description of the window, and it must be declared in the corresponding menu. These classes can only call to methods

from the kernel part to execute the actions, so the data is protected of this *external* layer.

- **jklass.nucli:** This package is the kernel part of KLASS. It contains all the methods that actually execute the actions asked for the user. The two main classes of this package are:

  - *GestorMatriu.java:* It represents a data matrix and contains all the necessary methods to manage it. It also contains the list of knowledge bases associated to the matrix, as well as the dendograms generated over it.
  - *GestorKlass.java:* As KLASS is an application thought for a multisession use, this means that more than one data matrix can exist at the same time. This class allows to manage the different matrixes, calling to the methods of the corresponding instance, and it allows to access to all the functionality provided by the KLASS' kernel.

- **jklass.util:** This package contains classes for the management of the system options, configuration parameters and calls to the operative system.

7.3 CCEC

For the implementation of CCEC methodology, a new functionality has been added to the system, that means, a new graphical user interface (a new Java class) has been added and some new methods has been included in some existing classes of the kernel. This new functionality generates an automatic interpretation of a classification previously done by the system (this classification is represented as a dendogram). It allows to select the dendogram to be interpreted, the number of classes that want to be obtained, and select different options for the interpretation, as for example the use of reviewed boxplot-based induction, the knowledge integration criteria and visualizations options as the generation of all the intermediate knowledge bases.

7.3.1 New classes added to the system

PanelConceptJerarq.java: It defines the new graphical interface for the new functionality, defining also the default parameters and generating the final file with the execution results. The new functionality has been placed in the next menu route:

Interpretation → Hierarchical conceptualization CCEC

7.3.2 Methods included into existing classes

In the GestorKlass.java class, have been implemented the following methods:

obtenirConceptJerarq: Method that realizes the hierarchical conceptualization. This consists of doing cuts in the dendogram, integrating the resulting classification in the data matrix and selecting the best rules from the whole set of inducted rules. Every new iteration makes the cut with a higher number of classes until reach the number given by the user.

bestRules: Method that, according to the knowledge integration criteria chosen by the user, returns the two best rules of the given knowledge base (one for each new generated classes at the current step).

compareRules: Method that, given two rules belonging to the same class, applies the selected knowledge integration criteria and returns the best one.

generarDescrp: Method that generates the descriptive file for the given knowledge base. It returns the name of the generated file.

In the Regla.java class, have been implemented the following methods:

support: Method that calculates the support value of the rule.

confidence: Method that calculates the confidence value of the rule.

coveringR: Method that calculates the relative covering value of the rule.

In the BaseConeixement.java class, have been implemented the following methods:

totalSupport: Method to calculate the total support of the knowledge base. It is calculated by adding the support values of the rules.

meanConfidence: Method to calculate the mean confidence of the knowledge base.

coveringG: Method that calculates the global covering value of the knowledge base.

7.4 Knowledge Base Quality

This functionality analyze the quality of the selected knowledge base, according to the evaluation criteria chosen by the user, as for example the confidence, the support, the covering, etc. applying this functions to all the rules of the knowledge base. The results of the analysis is a quality table that contains the name of the rule, the consequent of the rule, and the values for the selected criteria. It also allows to generate the descriptive analysis of the knowledge base, all in the same file.

7.4.1 New classes added to the system

PanelQualitatBC.java: It defines the new graphical interface for the new functionality, defining also the default parameters and generating the final file with the execution results. The new functionality has been placed in the next menu route:

Knowledge → Quality KB

TaulaQualitat.java: This class writes a table in a latex file with values for quality of the knowledge base.

7.4.2 Methods included into existing classes

In the GestorKlass.java class, have been implemented the following methods:

ferQualitatBC: Method that generates a TEX file with values of quality of the knowledge base.

In the GeneradorTex.java class, have been implemented the following methods:

generarLtxQualitatBC: Method that generates a LaTeX file with the quality values of the knowledge base.

Chapter 8

Case Study

8.1 Introduction

In this chapter is presented the application of the automatic interpretation process developed in this work to a case study within a medical domain, specifically the study of response to traumatic brain injury neurorehabilitation. The international scientific community sustains the existence of intrinsic characteristics of this population of patients that make difficult the use of the standard methodology used in other therapies or clinical trials. The strongest factors detected are the heterogeneity of the studied populations with a lack of knowledge about the natural evolution of the process for different patients, and the lack of knowledge about the active components of the treatments to be controlled [42].

8.2 Application domain

All patients meet criteria to initiate neuropsychological rehabilitation. Neuropsychological assessment covered the major cognitive domains:

- Language tests assess repetition of words, confrontation naming, and verbal comprehension.
- Measures of attention included Digit Span Forward, Trail Making Test-A, Sustained Attention Test, and Stroop Test (word-colour condition).
- Memory and Learning was assessed with Digit Span Backward, Immediate and delayed stories from the PIEN and Learning Curve Test.
- Executive functions was assessed with the Wisconsin Card Sorting Test, Trail Making Test-B, and Stroop Test (interference condition) [9]. After

the initial evaluation all the patients initiated a two months rehabilitation program with a personalized intervention, where patients worked in each one of the specific cognitive domains, considering the degree of the deficit and the residual functional capacity.

The target sample includes 47 patients with TBI between 17 and 68 years, receiving neurorehabilitation treatment at the Institut Guttmann-Hospital de Neurorehabilitació from November 2005 to December 2006. All patients were administered the neuropsychological assessment at admission. Same evaluation was also performed at the end of the rehabilitation. Differences between pre- and post-treatment test scores were used to measure particular patients improvements in the domains of language, attention, memory and executive functions.

8.2.1 Classification process

The classification process used to mine the data was a clustering based on rules, from the knowledge base provided by the experts. This knowledge base consists of the six following rules:

$r1$: If $(TAS.omis.pre = 10)$ and $(TAS.omis.dife = 0)$ and $(Stroop.int.pos = -25) \longrightarrow$ serious

$r2$: If $(TMT.A.pre = 300)$ and $(TMT.A.dife = 0)$ and $(TMT.B.pre = 500)$ and $(TMT.B.dife = 0)$ and $(Stroop.int.post = -25) \longrightarrow$ serious

$r3$: If $(TMT.B.pre = 500)$ and $(TMT.B.dife = 0)$ and $(WSCT.e.pre = 50)$ and $(WSCT.e.dife = 0)$ and $(Stroop.int.post = -25) \longrightarrow$ serious

$r4$: If $(TAS.omis.pre = 10)$ and $(TAS.error.pre = 10)$ and $(B.d.d.pre = 0)$ and $(B.d.ri.dife = 0)$ and $(Stroop.int.post > -25) \longrightarrow$ inhibited

$r5$: If $(TAS.omis.pre < 10)$ and $(TAS.error.pre < 10) \longrightarrow$ assessable

$r6$: If $(B.d.d.pre > 0)$ and $(B.d.i.pre > 0) \longrightarrow$ assessable

In this knowledge base there are basically three groups of patients:

- **Serious:** Those patients that at the beginning of the rehabilitation treatment are in a seriously state, so they are not even able to do the easiest tests and that do not modify their situation after the rehabilitation treatment.

- **Inhibited:** They are patients that are able to inhibit specific signals although they still present problems.

- **Assessable:** Those patients that at the beginning of the rehabilitation process can be evaluated even in the more complex tasks.

With these rules, a clustering based on rules has been made. The classification method used is the *Ward* method, with distance *Mixed Gibert* (alpha and beta values automatically generated and missing values substituted by mean values). The result of this classification is the dendogram of Figure 8.2.1. This classification is the start point of this case study, as the generated dendogram is used to do the automatic interpretation for 6 classes, selecting the boxplot based induction in its reviewed form.

Two interpretations have been made, one with each knowledge integration method (see §6.3).

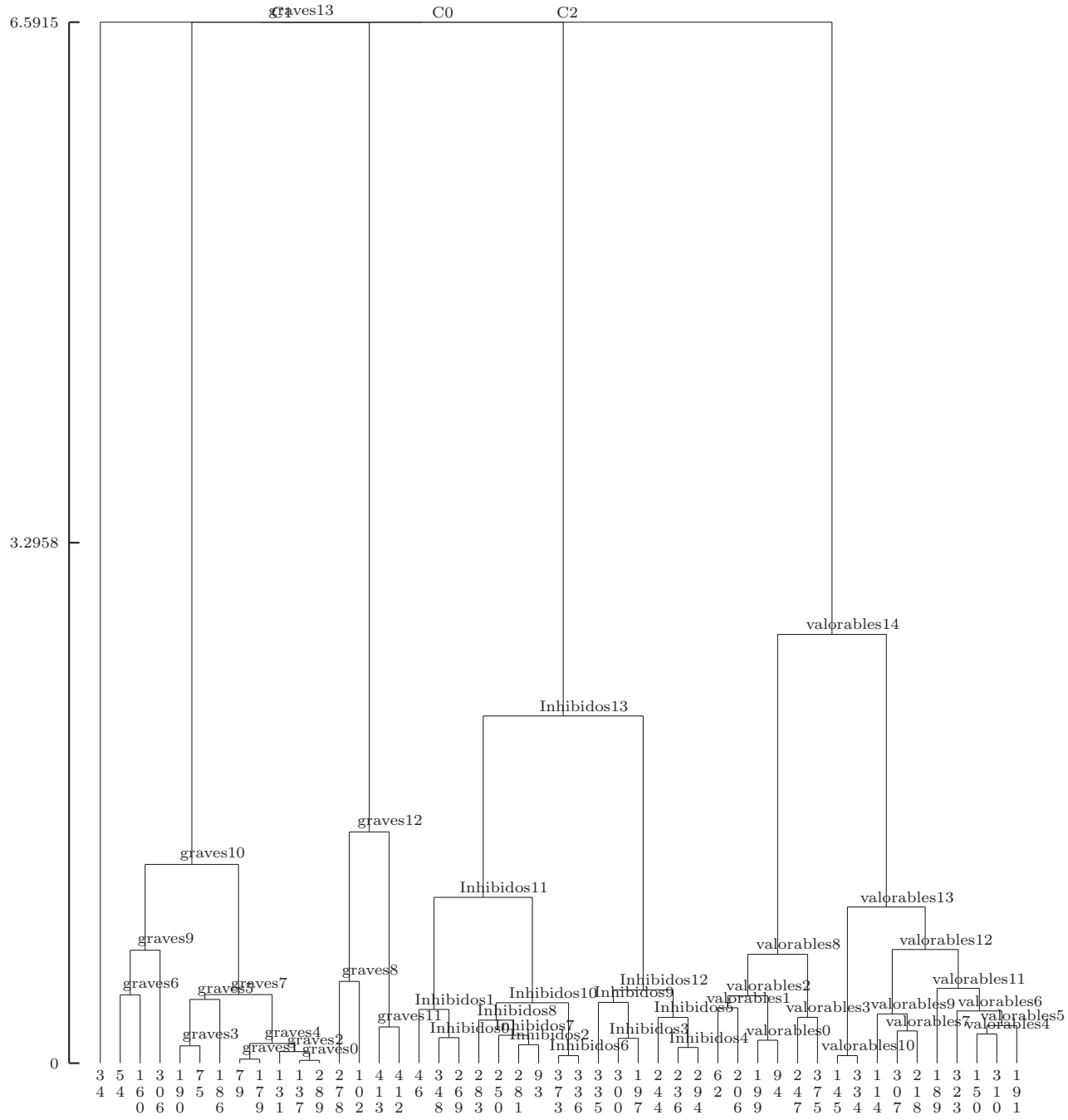

Figure 8.1: CAJ. Dendogram

8.3 Best Local and no Close-World Assumption Simple

8.3.1 Final rules

$r1.BC0.r30 : (B.aprend.pre >= 0)$ & $((B.aprend.pre <= 42)$ & $((B.d.d.pre > 0)$ & $(B.d.d.pre <= 5))) \longrightarrow (Simple3)34$

$r2.BC1.r28 : (B.aprend.pre >= 0)$ & $(((B.aprend.pre <= 42)$ & $((B.d.d.pre >= 0)$ & $(B.d.d.pre <= 0)))$ & $((Stroop.p.c.post > 14)$ & $(Stroop.p.c.post <= 42))) \longrightarrow (Simple4)Inhibidos13$

$r3.BC2.r119 : (B.aprend.pre >= 0)$ & $((((B.aprend.pre <= 42)$ & $((B.d.d.pre >= 0)$ & $(B.d.d.pre <= 0)))$ & $((Stroop.p.c.post >= 0)$ & $(Stroop.p.c.post <= 0)))$ & $((rep.p.pre > 8)$ & $(rep.p.pre <= 10))) \longrightarrow (Simple5)graves10$

$r4.BC3.r118 : (B.aprend.pre >= 0)$ & $((((B.aprend.pre <= 42)$ & $((B.d.d.pre >= 0)$ & $(B.d.d.pre <= 0)))$ & $((Stroop.p.c.post >= 0)$ & $(Stroop.p.c.post <= 0)))$ & $((rep.p.pre >= 0)$ & $(rep.p.pre <= 8))) \longrightarrow (Simple5)graves12$

$r5.BC4.r33 : (B.m.c.p.e.pre > 5)$ & $((B.m.c.p.e.pre <= 18)$ & $((Stroop.p.c.pre >= 0)$ & $(Stroop.p.c.pre <= 0))) \longrightarrow (Simple6)valorables8$

$r6.BC5.r41 : (B.m.c.p.e.pre > 5)$ & $((B.m.c.p.e.pre <= 18)$ & $((Stroop.p.c.dife >= -9)$ & $(Stroop.p.c.dife <= 15))) \longrightarrow (Simple6)valorables13$

Quality analysis

Values	**C**onsequent	**C**onfidence	**S**upport	**R**. covering	**G**. covering
r1	(Simple3)34	1	0.0213	1	0.0213
r2	(Simple4)Inhibidos13	1	0.2979	0.9333	0.2979
r3	(Simple5)graves10	1	0.234	1	0.234
r4	(Simple5)graves12	1	0.0851	1	0.0851
r5	(Simple6)valorables8	1	0.1277	1	0.1277
r6	(Simple6)valorables13	1	0.1915	0.9	0.1915

Total support of the knowledge base: 0.9574 **M**ean confidence of the knowledge base: 1

Table 8.1: Quality table with evaluation criteria values for BLnoCWA Simple case

8.4 Best Local and no Close-World Assumption

8.4.1 Final rules

$r1.BC0.r30-r38-r42-r46-r54-r70-r78-r86-r149 : (B.aprend.pre >= 0)$ & $((B.aprend.pre <= 42)$ & $((B.d.d.pre > 0)$ & $((B.d.d.pre <= 5)$ & $((B.d.i.pre > 0)$ & $((B.d.i.pre <= 5)$ & $((B.d.i.dife >= -1)$ & $((B.d.i.dife <= -1)$ & $((B.m.c.p.e.pre > 0)$ & $((B.m.c.p.e.pre <= 5)$ & $((B.m.c.p.p.pre > 0)$ & $((B.m.c.p.p.pre <= 11)$ & $((B.m.l.p.p.pre > 0)$ & $((B.m.l.p.p.pre <= 5)$ & $((B.aprend.pre > 0)$ & $((B.aprend.pre <= 42)$ & $((B.v.rec.pre > 0)$ & $((B.v.rec.pre <= 8)$ & $((EDAT >= 16)$ & $(EDAT <= 16)))))))))))))))))) \longrightarrow (BLnoCWA3)34$

$r2.BC1.r28-r31-r148-r151 : (B.aprend.pre >= 0)$ & $(((B.aprend.pre <= 42)$ & $((B.d.d.pre >= 0)$ & $((B.d.d.pre <= 0)$ & $((B.d.i.pre >= 0)$ & $((B.d.i.pre <= 0)$ & $((B.m.c.p.e.pre >= 0)$ & $((B.m.c.p.e.pre <= 0)$ & $((B.m.c.p.p.pre >= 0)$ & $((B.m.c.p.p.pre <= 0)$ & $((B.m.l.p.p.pre >= 0)$ & $((B.m.l.p.p.pre <= 0)$ & $((B.aprend.pre >= 0)$ & $((B.aprend.pre <= 0)$ & $((B.v.rec.pre >= 0)$ & $(B.v.rec.pre <= 0))))))))))))))$ & $((Stroop.p.c.post > 14)$ & $((Stroop.p.c.post <= 42) \| ((Stroop.p.c.dife > 14)$ & $((Stroop.p.c.dife <= 42) \| ((Stroop.int.post > -10)$ & $((Stroop.int.post <= 13.6) \| ((Stroop.int.dife > 15)$ & $(Stroop.int.dife <= 38.6)))))))))) \longrightarrow (BLnoCWA4)Inhibidos13$

$r3.BC2.r119-r121-r122-r138-r140-r145-r147-r148 : (B.aprend.pre >= 0)$ & $((((B.aprend.pre <= 42)$ & $((B.d.d.pre >= 0)$ & $((B.d.d.pre <= 0)$ & $((B.d.i.pre >= 0)$ & $((B.d.i.pre <= 0)$ & $((B.m.c.p.e.pre >= 0)$ & $((B.m.c.p.e.pre <= 0)$ & $((B.m.c.p.p.pre >= 0)$ & $((B.m.c.p.p.pre <= 0)$ & $((B.m.l.p.p.pre >= 0)$ & $((B.m.l.p.p.pre <= 0)$ & $((B.aprend.pre >= 0)$ & $((B.aprend.pre <= 0)$ & $((B.v.rec.pre >= 0)$ & $(B.v.rec.pre <= 0))))))))))))))$ & $((Stroop.p.c.post >= 0)$ & $((Stroop.p.c.post <= 0) \| ((Stroop.p.c.dife >= 0)$ & $((Stroop.p.c.dife <= 0) \| ((Stroop.int.post >= -25)$ & $((Stroop.int.post <= -25) \| ((Stroop.int.dife >= 0)$ & $(Stroop.int.dife <= 0)))))))))$ & $((rep.p.pre > 8)$ & $((rep.p.pre <= 10)$ & $((rep.p.post > 9)$ & $((rep.p.post <= 10)$ & $((rep.p.dife >= 0)$ & $((rep.p.dife <= 0)$ & $((comp.p.pre > 0)$ & $((comp.p.pre <= 12)$ & $((comp.p.post > 10)$ & $((comp.p.post <= 12)$ & $((comp.ord.pre > 1)$ & $((comp.ord.pre <= 16)$ & $((comp.ord.post > 10)$ & $((comp.ord.post <= 16)$ & $((comp.ord.dife >= 0)$ & $(comp.ord.dife <= 0)))))))))))))))) \longrightarrow (BLnoCWA5)graves10$

$r4.BC3.r118 - r120 - r137 - r139 - r144 - r146 : (B.aprend.pre >= 0)$ & $((((B.aprend.pre <= 42)$ & $((B.d.d.pre >= 0)$ & $((B.d.d.pre <= 0)$ & $((B.d.i.pre >= 0)$ & $((B.d.i.pre <= 0)$ & $((B.m.c.p.e.pre >= 0)$ & $((B.m.c.p.e.pre <= 0)$ & $((B.m.c.p.p.pre >= 0)$ & $((B.m.c.p.p.pre <= 0)$ & $((B.m.l.p.p.pre >= 0)$

& ((*B.m.l.p.p.pre* <= 0) & ((*B.aprend.pre* >= 0) & ((*B.aprend.pre* <= 0) & ((*B.v.rec.pre* >= 0) & (*B.v.rec.pre* <= 0)))))))))))))) & ((*Stroop.p.c.post* >= 0) & ((*Stroop.p.c.post* <= 0)‖((*Stroop.p.c.dife* >= 0) & ((*Stroop.p.c.dife* <= 0)‖((*Stroop.int.post* >= −25) & ((*Stroop.int.post* <= −25)‖((*Stroop.int.dife* >= 0) & (*Stroop.int.dife* <= 0)))))))) & ((*rep.p.pre* >= 0) & ((*rep.p.pre* <= 8) & ((*rep.p.post* >= 2) & ((*rep.p.post* <= 9) & ((*comp.p.pre* >= 0) & ((*comp.p.pre* <= 0) & ((*comp.p.post* >= 0) & ((*comp.p.post* <= 10) & ((*comp.ord.pre* >= 0) & ((*comp.ord.pre* <= 1) & ((*comp.ord.post* >= 1) & (*comp.ord.post* <= 10)))))))))))) ⟶ (*BLnoCWA*5)*graves*12

*r*5.*BC*4.*r*33 − *r*185 : (*B.m.c.p.e.pre* > 5) & (((*B.m.c.p.e.pre* <= 18) & ((*B.m.l.p.p.pre* > 5) & ((*B.m.l.p.p.pre* <= 20) & ((*B.aprend.pre* > 43) & (*B.aprend.pre* <= 88))))) & ((*Stroop.p.c.pre* >= 0) & ((*Stroop.p.c.pre* <= 0) & ((*Stroop.int.pre* >= −25) & (*Stroop.int.pre* <= −25))))) ⟶ (*BLnoCWA*6)*valorables*8

*r*6.*BC*5.*r*41 : (*B.m.c.p.e.pre* > 5) & (((*B.m.c.p.e.pre* <= 18) & ((*B.m.l.p.p.pre* > 5) & ((*B.m.l.p.p.pre* <= 20) & ((*B.aprend.pre* > 43) & (*B.aprend.pre* <= 88))))) & ((*Stroop.p.c.dife* >= −9) & (*Stroop.p.c.dife* <= 15))) ⟶ (*BLnoCWA*6)*valorables*13

Quality analysis

Values	**C**onsequent	**C**onfidence	**S**upport	**R**. covering	**G**. covering
r1	(BLnoCWA3)34	1	0.0213	1	0.0213
r2	(BLnoCWA4)Inhibidos13	1	0.2979	0.9333	0.2979
r3	(BLnoCWA5)graves10	1	0.234	1	0.234
r4	(BLnoCWA5)graves12	1	0.0851	1	0.0851
r5	(BLnoCWA6)valorables8	1	0.1064	0.8333	0.1064
r6	(BLnoCWA6)valorables13	1	0.1915	0.9	0.1915

Total support of the knowledge base: 0.9362
Mean confidence of the knowledge base: 1

Table 8.2: Quality table with evaluation criteria values for BLnoCWA case

We see that this second criteria obtained a more detailed description of the generated classes being the chosen option to present the interpretation to the expert.

Chapter 9

Conclusions and Future Work

9.1 Conclussions

The aim of this project was to enlarge KLASS with a first version of the automatic interpretation module. Extend an on-going application is always a delicate task, since new functionalities must be correctly integrated without affecting the right running of what was already done. However, this task becomes especially difficult in cases like KLASS, a ten years old project that has been developed through this time by very different people. The only way to maintain an application like this is maintaining a strict methodology in expansion of new functionalities tasks, as well as keeping detailed documentation through comments to understand the code and user's guides to know how it must be used. In fact, this intervention has been possible due to the powerful substrate found. Even so, a considerable effort must be done to understand how it works before any change could be introduced. In fact, creation of a new project has been needed to decide where and how to introduce the new modules and functionalities here exposed.

9.2 Future work

As it has been said in §6.3.1, the system has been prepared to allow introduction of new evaluation criteria and knowledge integration methods. Although two of these methods (BL&noCWA Simple and BL&noCWA) have been implemented, is necessary to implement all the criteria defined in [3] and do the comparison between all of them to elaborate a final proposal.

Finally, related to the case study presented in §8, the automatic interpretation of classes obtained under the Best Local and no Close-World assumption must be contrasted with the manually interpretation of experts to validate the results.

Bibliography

[1] L Adelman. *Evaluating Decision Support and Expert Systems*. John Wiley and Sons, New York, NY, 1992. [cited at p. 7]

[2] S. Bayona. Descriptiva de dades y de classes. PFC Facultat d'Informàtica, UPC, 2000. [cited at p. 32]

[3] Alejandra Pérez Bonilla. Metodología de caracterización conceptual por condicionamientos sucesivos. una aplicación a sistemas medioambientales. Master's thesis, Statistics and Operative Research Department, UPC, April 2009. [cited at p. 6, 22, 29, 37, 55]

[4] R. Brachman and T. Anand. The Process of Knowledge Discovery in Databases: A Human-Centered Approach. *In Advances in Knowledge Discovery and Data Mining*, pages 65–78, 1996. [cited at p. 11]

[5] X. Castillejo. Un entorn de treball per a Klass. PFC Facultat d'Informàtica, UPC, 1996. [cited at p. 32]

[6] U. Cortés, M. Sànchez-Marrè, L. Ceccaroni, I. R-Roda, and M. Poch. Artificial intelligence and environmental decision support systems. *Applied Intelligence*, 13(1):77–91, 2000. [cited at p. 8]

[7] E. Diday and K.C. Gowda. Symbolic clustering using a new dissimilarity measure. 24(6):567–578, 1991. [cited at p. 32]

[8] E. Diday and K.C. Gowda. Symbolic clustering using a new similaritiy measure. In *IEEE Trans. on systems, man.;and cib.*, volume 22, pages 368–378, 1992. [cited at p. 32]

[9] Lezak MD. et al. *Neuropsychological Assesment*. Oxford University Press, New York, 2004. [cited at p. 47]

[10] Rafel Farr, Robert Nieuwenhuis, Pilar Nivela, Albert Oliveras, and Enric Rodrguez. *Introduccin a la lgica*. Notas de clase. FIB. Barcelona, 2008. [cited at p. 28]

[11] U. Fayyad. From data mining to knowledge discovery: An overview. *Advances in KD and DM, Fayyad, AAAI/MIT.*, 1996. [cited at p. 9, 11]

[12] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From Data Mining to Knowledge Discovery in Databases (a survey). *AI Magazine.*, 3(17):37–54., 1996. [cited at p. 9]

[13] K. Gibert. Klass. Estudi d'un sistema d'ajuda al tractament estadístic de grans bases de dades. Master's thesis, UPC, 1991. [cited at p. 31]

[14] K. Gibert. Klass. estudi d'un sistema d'ajuda al tractament estadístic de grans bases de dades. Master's thesis, UPC, 1991. [cited at p. 32]

[15] K. Gibert. *L'us de la Informació Simbòlica en l'Automatització del Tractament Estadístic de Dominis Poc Estructurats.* In the statistics and operations research phd. thesis., Universitat Politecnica de Catalunya, Barcelona, Spain, 1994. [cited at p. 17, 21, 31]

[16] K. Gibert. *Ph. D. Thesis.* Eio dep., UPC, Barcelona, Spain, 1994. [cited at p. 32]

[17] K. Gibert. On the uses and costs of rules-based classification. In A. Prat. Physica-Verlag, editor, *Proceedings of Computational Statistics*, pages 265–270, 1996. [cited at p. 23]

[18] K. Gibert. The use of symbolic information in automation of statistical treatment for ill-structured domains. *AI Communications*, 9(1):36–37, 1996. [cited at p. 19]

[19] K. Gibert. *Tendencia de la Minería de Datos en España*, chapter :Técnicas híbridas de Inteligencia Artificial y Estadística para el descubrimiento de conocimiento y la minería de datos, pages 119–130. Thompson Ed., 2004. [cited at p. 19, 20, 23, 31, 37]

[20] K. Gibert and T. Aluja. A computational technique for comparing classifications and its relationship with knowledge discovery. In *International Seminar on New Techniques and Technologies for Statistics*, pages 193–198, Italy, 1998. [cited at p. 31]

[21] K. Gibert, T. Aluja, and U. Cortés. Knowledge Discovery with Clustering Based on Rules. Interpreting results. In *Principles of Data Mining and Knowledge Discovery*, volume 1510 of *LNAI*, pages 83–92. Springer, 1998. [cited at p. 12, 14, 23, 37]

[22] K. Gibert and U Cortés. Combining a knowledge based system with a clustering method for an inductive construction of models. In *Proc. 4th Int Work. on AI and Stats.*, 1993. Florida, USA. [cited at p. 31]

[23] K. Gibert and U Cortés. On the uses of the expert knowledge for automatic biasing of a clustering method. In *ITI 93. Proceedings of the International Conference on Information Technology Interfaces*, pages 219–224, Croatia, 1993. issn 1330-1012. [cited at p. 31]

[24] K. Gibert and U. Cortés. *Combining a knowledge-based system and a clustering method for a construction of models in ill-structured domains*, volume 89 of *LNS*, pages 351–360. Springer-Verlag., 1994. [cited at p. 31]

[25] K. Gibert and U. Cortés. Clustering based on rules and knowledge discovery in ill-structured domains. *Computación y Sistemas.*, 1(4):213–227, 1998. [cited at p. 31]

[26] K. Gibert and U. Cortés. Generación automática de reglas a partir de la caracterización de clases. *Butlleti de l'ACIA*, pages 14–15, 1998. [cited at p. 20]

[27] K. Gibert, M. Hernández, and U. Cortés. Classification based on rules: an application to Astronomy. In U. Tokio. Japón, editor, *5th. IFCS*, pages 69–72, 1996. [cited at p. 31]

[28] K. Gibert and R. Nonell. Pre and postprocessing in klass. In M. Snchez-Marr, J. Bjar, J. Comas, A.E. Rizzoli, and G. Guariso, editors, *Proceedings of the iEMSs Fourth Biennial Meeting*, volume 3, pages 1965–1966, Barcelona-Catalunya, 2008. [cited at p. 31]

[29] K. Gibert and A Pérez-Bonilla. Análisis y propiedades de la metodología Caracterización Conceptual por Condicionamientos Sucesivos (CCCS). Research DR 2005/14, Dep. Estadística e Investigación Operativa. Universidad Politécnica de Catalua, Barcelona, España, 2005. [cited at p. 22]

[30] K. Gibert and A. Pérez-Bonilla. Ventajas de la estructura jerárquica del clustering en la interpretación automática de clasificaciones. In *III Spanish Workshop on Data Mining and Learning*, pages 67–76, Granada, 2005. [cited at p. 37]

[31] K. Gibert and A. Pérez-Bonilla. Revised boxplot based discretization as a tool for automatic interpretation of classes from hierarchical cluster. In *Data Science and Classification*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 229–237, Ljubljana, Slovenia, 2006. Springer-Verlag. [cited at p. 37]

[32] K. Gibert and I. Roda. Identifying characteristic situations in wastewater treatment plants. In *Workshop in Binding Environmental Sciences and Artificial Intelligence*, volume 1, pages 1–9, 2000. [cited at p. 23, 31]

[33] K. Gibert and A. Salvador. Aproximación difusa a la identificación de situaciones características en el tratamiento de aguas residuales. In *X Congreso Español sobre tecnologías y lógica fuzzy*, pages 497–502, España, 2000. [cited at p. 23, 26]

[34] K. Gibert and Z. Sonicki. Classification based on rules and thyroids dysfunctions. *Applied Stochastic Models in Business and Industry*, 15(4):319–324, 1999. [cited at p. 31]

[35] J.C. Gower. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53:315–328, 1966. [cited at p. 32]

[36] J.C. Gower. A comparison of some methods of cluster analysis. *Biometrics*, 23(4):623–37, 1967. [cited at p. 32]

[37] J.C. Gower. A General coefficient of similarity and some of its properties. *Biometrics*, 27:857–874, 1971. [cited at p. 32]

[38] I.G. Haagsma and R.D. Johanns. *Environmental Systems*, chapter Decision support systems: an integrated approach, pages 205–212. Chicago Linguistic Society, 1994. [cited at p. 7]

[39] Moon Yul Huh and Kwangryeol Song. Davis: A java-based data visualization. *Computational Statistics*, 17(3):411–423, 2002. [cited at p. 11]

[40] Eibe Frank. Ian H. Witten. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann., 1999. Reviews: Review by J. Geller (SIGMOD Record, Vol. 32:2, March 2002),Review by E. Davis (AI Journal, Vol. 131:1-2, September 2001), Review by P.A. Flach (AI Journal, Vol. 131:1-2, September 2001). [cited at p. 11]

[41] M. Ichino and H. Yaguchi. Generalized Minkowski metrics for mixed features. *Trans. IEICE Japó*, J72-A(2):398–405, 1989. (en japons). [cited at p. 32]

[42] Alejandro García et al. Karina Gibert. Response to traumatic brain injury neurorehabilitation through an artificial intelligence and statistics hybrid knowledge discovery from databases methodologies. *Medical Archives*, 63(3), 2008. [cited at p. 47]

[43] A. Lebart, A. Morineau, and J.P. Fenelon. *Tratamiento estadstico de datos.* Marcombo, 1985. [cited at p. 17]

[44] L. Lebart, A. Morineau, and T. Lambert. *SPAD.N : manual de referencia, versión 2.5. Sistema compatible para el análisis de datos.* Saint-Mandé: CISIA., 1994. traducido por: T. Aluja , E.Ibáez. [cited at p. 11]

[45] W. Chen S. Ma Y. Liu, B. Hsu. Analyzing the subjective interestigness of association rules. *IEEE Intelligent Systems*, pages 47–55, 2000. [cited at p. 29, 30]

[46] J. Márquez and J.C. Martín. La clasificación automática en las ciencias de la salud. PFC, 1997. Facultat de Matemàtiques i Estadística, UPC. [cited at p. 32]

[47] Oliveras Castellà, Josep. Dades heterogènies amb classificació automàtica. Implementació i comparativa de mètriques mixtes. PFC. [cited at p. 32]

[48] H.A. Ralambondrainy. A conceptual version of K-means algorithm. *Pattern Recognition Letters.*, 16:1147–1157, 1995. [cited at p. 32]

[49] Henri Ralambondrainy. *A clustering method for nominal data and mixture of numerical and nominal data. Clasification and Related Methods of Data Analysis.* H.H.Bock, Elsevier Science Publishers, B.V. (North-Holland), 1988. [cited at p. 32]

[50] Jorge Enrique Rodas Osollo. *Knowledge Discovery in repeated and very short serial measures with a blocking factor.* Programa de doctorado: Inteligencia artificial, Universitat Politecnica de Catalunya, 2003. [cited at p. 32]

[51] R.S. Sojda. *Ph. D. Thesis: Artificial intelligence based decision support for trumpeter swan management.* Fort collins, colorado., Colorado State University, USA, 2002. [cited at p. 7]

[52] X. Tubau. Sobre el comportament de les mètriques mixtes en algorismes de Clustering. PFC. [cited at p. 32]

[53] J.W. Tukey. *Exploratory Data Analysis.* Addison-Wesley, 1977. [cited at p. 18]

[54] F. Vázquez and K Gibert. Generación automática de reglas difusas en dominios poco estructurados con variables numéricas. In *IXth CAEPIA v. I*, pages 143–152, 2001. [cited at p. 22, 24, 25]

[55] F. Vázquez and K Gibert. Robustness of class prediction depending on reference partition in ill-structured domains. In *XVIII Iberoamerican Conference on Artificial Intelligence, Workshop de Minera de Datos y Aprendizaje (IBERAMIA2002)*, pages 13–22, Sevilla, 2002. [cited at p. 32]

[56] Fernando Vázquez Torres. Caracterización e Interpretación Automática de Descripciones Conceptuales en Dominios Poco Estructurados usando Variables Numéricas. Master's thesis, Facultad de Informtica de Barcelona. Universidad Politécnica de Cataluã, 2002. [cited at p. 20, 22]

[57] B. Visauta. *Análisis Estadístico con SPSS para WINDOWS.* Mc.Graw Hill., 1998. (Vol II. Análisis Multivariante). [cited at p. 11]

[58] M. Volle. Analyse des données, 1985. Ed. Economica, Paris, France. [cited at p. 17]

List of Figures

List of Tables