JavaKlass – v8.4

SISPD

Alberto Martínez González & Jordi Pascual

2011

Index

Introduction

When some data classification task is fulfilled, most of the methods existing in the literature try to identify the variables that describe the characteristics of the data set in an optimal way, from a statistical and in general mathematical point of view. They try to describe the data with the minimal set of attributes that describe it in better in the great majority of the cases.

Nevertheless, even if we find a method that describes our data with no mistakes at all, this description may be so cryptic that no human operator working with the data may be able to understand what is happening with the system, even if it identifies the class to which a new element interacting with the system does belong to.

For example, when studying a waste water treatment plant as it has been done repeatedly in the literature (1), it may be interesting to show to the technician operating the final system a more global picture of what is happening in the system that optimal result of the traditional systems. Perhaps the notion that the total organic carbon concentration at the effluent of the plant is really low is enough to determine a unique and identifiable case from a mathematical point of view, but when presenting this information to the technician he will not really know what is going on in the system. If instead we present him also with the information that the waste water influent flow rate and the height of the waste water in the tank are very high, he will immediately understand that there is a storm even telemetrically, and he will be able to take the pertinent measures.

With this need in mind, Conceptual Characterization by Embedded Conditioning (2) was developed. Basically, it is a methodology that tries to generate authomatically the conceptual descriptions of a classification process that may support later decision-making. Several strategies have been proposed to select those conceptual descriptions in the most useful way (3).The main focus of this document will be to implement two of this strategies.

The platform selected for the implementation will be Java Klass, a platform for data analysis first developed by Karina Gibert in her doctoral thesis (4), and later improved by several authors. The second focus of this document is to analyze and solve several of the bugs implemented by the previous developers.

Description of the Methodology

Conceptual Characterization by Embedded Conditioning (CCEC) starts from a hierarchical clustering of the dataset. To explain it properly, we have to set up some nomenclature. For every clustering, the input is a standard data matrix

The standard input of a clustering algorithm is a data matrix with the values of K variables $X1 . . . XK$ (numerical or not) observed over a set $I = \{1, . . . n\}$ of individuals. Variables are in columns, while individuals in rows. Cells contain the value (xik), taken by individual $i \in I$ for variable Xk , ($k = 1 : K$). The set of values of Xk is named $Dk = \{ck1 , ck2 , ..., cks \}$ for categorical variables and $Dk = rk$ for numerical ones, being $rk = [minXk ,maxXk]$ the range of XK . A partition in ξ classes of I is denoted by $P\xi = \{C1 , ..., C\xi \}$, and $\tau = \{P1 , P2 , P3 , P4 , ..., Pn \}$ is an indexed hierarchy of I. Finally, $P2 = \{C1 , C2 \}$ is a binary partition of I. Usually, τ is the result of a hierarchical clustering over I, and it can be represented in a graphical way as an horizontal cut of the corresponding dendrogram (or hierarchical tree, see Figure 1, Pérez-Bonilla et al. [2008]).

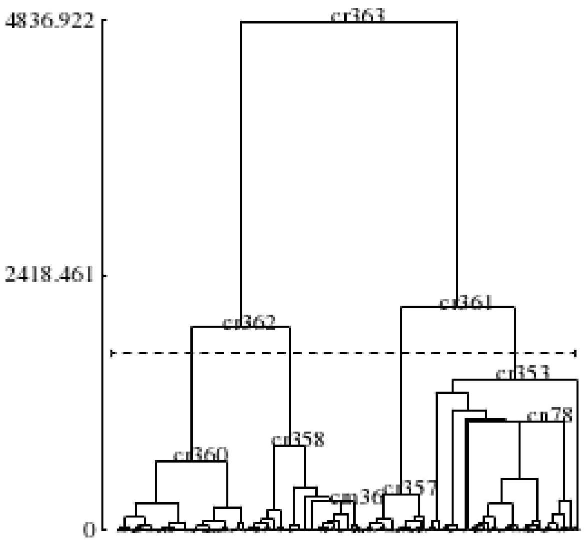


Figure 1: Dendrogram $[\tau^{EnW,G}_{Lj3,R2}]$.

CCEC is a methodology globally described in Pérez-Bonilla et al. [2008] that takes advantage of the existence of τ to generate conceptual interpretations of a of a given partition P ∈ τ in terms of formal descriptions. CCEC uses the property of all binary hierarchical structure that Pξ+1 has the same classes of Pξ except one, which splits in two subclasses in Pξ+1 . The binary hierarchical structure represented in τ is used in CCEC to discover particularities of the final classes step by step by analyzing the hierarchy top-down. It uses Boxplot based discretization (BbD), see Gibert and Pérez-Bonilla [2006]), as an efficient way of transforming all numerical variable into qualitative ones in such a way that every resulting qualitative variable maximizes the association with the reference partition. See Gibert and Pérez-Bonilla [2006] for details. Briefly, main idea is to use as cut-points the extreme values (minimum and maximum) that the numerical variable locally takes in every class of P. BbD is the kernel of Boxplot based induction rules (BbIR) (presented in Pérez-Bonilla and Gibert [2007]). It is a method for inducing probabilistic rules (r : xik ∈ Iks −→ C, being psc ∈ [0, 1] the certainty degree of r). The produced rules have a minimum number of attributes in the antecedent, and those are formalized on the basis of the intervals induced by BbD for every variable. The CCEC methodology was formalized in Pérez-Bonilla et al. [2008]. Here an algorithmic version is presented:

1. Consider the top of the tree: ξ = 1; P1 = I; AP1 = {A1 : true}

2. Go down one level in the tree, by making ξ = ξ + 1 and so considering the new P ξ . Being τ an indexed hierarchy, P ξ is embedded in P ξ−1 in such a way that there is a single class of P ξ−1 , namely Ct ξ−1 , splitting in two new classes of P ξ , namely C ξ i and C ξ j and all other classes C ξq , q ¬= i, j, are common to both partitions and Cξ q = Cξ-1 q ∀q ¬= i, j. Consider the restricted partition P*ξ = {Cξi , Cξj }. It holds that P*ξ ⊂ Pξ and when ξ = 2, P*ξ = Pξ .

   As in previous iteration the class C ξ−1t = {C ξ i ∧ C ξ j } was already distinguished from the rest by proper concept, it is enough to find distinction between Cξi and Cξj .

3. Use BbD (Gibert and Pérez-Bonilla [2006]), to find (total or partial) characteristic values regarding P*ξ Gibert et al. [1998] for all numerical variables.

4. Use BbIR, to induce a knowledge base R(P*ξ) describing both classes {Cξi , Cξj }.

5. Search the best rule for each class of the restricted partition $P^{*}\xi = \{C\xi i , C\xi j \}$. In the next section several criteria are presented to determine them. Name $Ai^{*}\xi$ and $Aj^{*}\xi$ the antecedents of the rules selected for $C\xi i$ and $C\xi j$ respectively.

6. Integrate $Ai^{*}\xi$ and $Aj^{*}\xi$ with the father's concept from previous iteration. Compound concepts are associated to $C\xi i$ and $C\xi j$: $Ai\xi = At\xi{-}1 \wedge Ai^{*}\xi$; $Aj^{*}\xi = At\xi{-}1 \wedge Aj^{*}\xi$.

   Description of both $C\xi i$ and $C\xi j$ inherits the properties of the father class $C\xi\text{-}1j$.

7. Build the concepts system:

   $AP\xi = AP\xi{-}1 \setminus \{Ct : At \} \cup \{C\xi i : A\xi i , C\xi j : A\xi j \}$

8. Go down one level in the tree, by making $\xi = \xi + 1$ and so considering

   $P\xi{+}1$. Return to step 2 and repeat until $P\xi = P$, P target partition to be interpreted.

9. Finally, $AP\xi = \{C : AC\ \forall C \in P\xi \}$ and also, the concepts system can be associated to a rules system $R(P\xi) = \{r \text{ tq } r : A \xrightarrow{p(r)} C\ \forall C \in P\xi \}$.

The set of concepts $AP\xi$ can, in fact, be considered as a domain model which can support later decision-making Power [2002] on the application domain. As a standard treatment is previously associated to every class by experts, evaluation of $AP\xi$ on new objects can help for treatment assignment. In this context, the possibility of easily interpreting and understanding the classes is critical. The proposed method provides simple and short rules which use to be easier to handle than those provided by other inductive methods.

BL&partial-CWA

For each of the 2 classes:

Find all the rules that have "Confiança = 1" and maximum "Cobertura relativa"

a) If you find only one rule for each class, being Ai and Aj their antecedents,
   a. if they refer to the same variable, store both as the rules for each class (10.21 and 10.22 in the thesis)
   b. else, the rule for class 1 is "Ai or NOT(Aj)" and the other way round for class2 (10.23 and 10.24)
b) If there are many rules
   a. If you have rules with "Confiança = 1" and "Cobertura relativa = 1" you chain their antecedents as it is said in (10.25) and (10.26)
   b. else you chain the antecedents as it is said in (10.27) and (10.28)

BL+G&CWA

For each of the 2 classes:

Find all the rules that have "Confiança = 1" and maximum "Cobertura relativa"

a) If you find only one rule for each class, being Ai and Aj their antecedents,

a. if they refer to the same variable,
   i. if ("Cobertura relativa" if the rule with Ai) > ("Cobertura relativa" if the rule with Aj), then apply (10.29) and (10.30) in the thesis. I think you can understand it, it is the same naming as in the previous criteria.
   ii. else apply (10.31) and (10.32) in the thesis.
b. else (10.33 and 10.34 in the thesis, which are the same as 10.23 and 10.24)

b) if there are many rules
   a. If you have rules with "Confiança = 1" and "Cobertura relativa = 1" you chain their antecedents as it is said in (10.35) and (10.36) (but for 10.36 use DeMorgan law, please, you know (NOT(A) and NOT(B)) = NOT(A or B), because it is a trick I used to simplify the code)
   b. else you chain the antecedents as it is said in (10.37) and (10.38) (Notice that, out of the four terms in each expression, the first one specifies that you select the rules that do not share the same variable with the rules of the other class, the second term says that you select those rules that do not share the same variable with any of the rules from the first class, the third is selecting those with greater "Cobertura relativa" than any of those of the second class, and the fourth is selecting those with greater "Cobertura relativa" than any of those of the first class)

Description of JavaKlass

JavaKlass is a software platform implemented by several authors. The main purpose of the platform is to provide a framework to solve decision support problems in the intersection of statistical techniques and artificial intelligence ones. This piece of software integrates several methodologies, and it has been used successfully in several different real applications.

Structure of JavaKLASS

Klass is a tool developed entirelly in Java programming language. It has been structured in the three following layers:

- jklass.ui: This layer contains all the files related to the graphical interface of the program.
- jklass.nucli: The Kernel of the program. All the classes associated data processing are embedded here.
- jklasss.util: Here there are all the classes specifying configuration parameters, management system options or calls to the operative system, so all the data access is also here.

KLASS Chronology

- Feb. 1991 KLASS v0. Karina Gibert's dissertation (6). It classifies data matrixes of heterogeneous data with mixed distance.
- Nov. 1994 KLASS v1. Karina Gibert's thesis (4). It is an extension of KLASS v0. It includes classication based in rules.

- Jul. 1996 KLASS v1.1. PFC Xavier Castillejo (7). 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 xcn.KLASS to the Lisp kernel of this new version and xcn.i to the C interface.
- Oct. 1997 jj.KLASS. PFC Juan José Marquez and Juan Carlos Martín (8). It incorporates to the KLASS.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.
- Set. 1999 KLASS v1.2. PFC Xavier Tubau (beta version) (9). It incorporates to the xcn.KLASS version the comparison of classifications module of jj.KLASS, the Ralambondrainy's mixed metric (10)and prepares the formulation of three more for their later implementation. Let's call xt.KLASS to the Lips kernel of this new version and X to the associated C interface.
- 1999-2000 KLASS+ v1. PFC Silvia Bayona (11). Denitive fusion of versions xt.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 sbh.KLASS to the Lisp kernel of this new version and sbh.i to the associated C interface.
- 2000-2002 KLASS+ v2. PFC Josep Oliveras (12). It incorporates to sbh.KLASS the pending mixed metrics (Gower (13) (14) (15),Gowda-Diday (16) (17) and Ichino-Yaguchi (18)). Let's call joc.KLASS to this new version.
- 2000-2003 jr.KLASS+. Jorge Rodas's thesis (19). Integrates KLASS+ v.2 and Columbus.
- 2000-2003 Anna Salvador and Fernando Vázquez research (20). Developement of CIADEC, that is later presented.
- 2002-2003 Java-KLASS v0. PFC Mª del Mar Colillas. Java version of the descriptive analysis module and integration with CIADEC and Columbus.
- 2003-2005 Java-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 Java-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 Java-KLASS v2.0. PFC Jose Ignacio Mateos. Extension of JavaKLASS 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 Java-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 Java-KLASS v4.0. PFC Laia Riera Guerra. Introduction, management and evaluation of Knowledge Bases. Extension of Java-KLASS with a module of

transformation of variables that allows discretizations, recodifications and arithmetic calculations with numerical variables. Finally, thisversion 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 Java-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 Java-KLASS v6.0. Alejandro Garc?a's thesis. Exogenous classification based on rules. Internationalization and location to three languages (Catalan,English, Spanish). Matrix merger.
- 2008 Java-KLASS v6.4. Master thesis of Alfons Bosch Sansa, Patricia García Giménez, 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.
- 2009. Esther Lozano thesis. Inducted Concepts From Embbeded Classes For Automatic Interpretation In Hierarchical Clustering.
- 2010. Assignment for the course Intelligent Decision Support Systems, by Beltran Fiz, Roc Oliver and Narcís Margall (5). Implementation of the first three first strategies for selecting the rules in the CCEC methodology, proposed by Perez-Bonilla (3).
- 2011. Assignment for the course Intelligent Decision Support Systems, by Alberto Martinez and Jordi Pascual. Implementation of the BL&partial-CWA and BL+G&CWA methods, and some latex generation improvements.

Implementation

This section explains how the JavaKLASS platform was updated in other to accomplish the assignment requirements, and the developer solving that concrete problem:

Problems solved:

- Implement BL&partial-CWA method (PanelConceptJerarq.java, GestorKlass.java and Regla.java).
- Implement BL+G&CWA method (PanelConceptJerarq.java, GestorKlass.java and Regla.java).
- Fix the problem with the intervals when revised method is applied (generarReglaProbNumerica(...) in BaseConeixementProb.java)
- Change the option Interpretation>Induccio de regles "boxplot-based" so that it works with any qualitative variable (PanelGenerarRegles.java).

- Change the conceptualitzacio-jerarquica method, in such a way that it can properly work with the "corrected" version of “induccio de regles boxplot based” (PanelGenerarRegles.java) .
- Scroll Panel didn't work properly in the PanelFiltre.java class (Dades -> Selecciona submatriu, panel "Llista vars.") .
- When selecting "Afegir a matriu de dades", force that "Genera fitxer .cls" is also selected (and that will also force that "Genera fitxer .par" is also selected) and cannot be deselected in PanelTallArbre.java class (Classificacio -> Dendrograma -> Talla Dendrograma).
- Solve some inconsistencies related to the use of the “ñ” charácter in the names of several variables and methods [1.- GestorMatriu.java -> sustituido "añadirArbol(...)" por "agregarArbol(...)"] AND [2.-DlgMerge.java, GestorClasificacion.java, GestorKlass.java, GestorMatriu.java -> Corregidas llamadas a "añadirArbol(...)" por "agregarArbol(...)"] AND [3.- PanelAgregarClase.java -> sustituida variable "tamañoLista" por "sizeList"].
- Reorganize the window of CCEC in such a way that the criterion BG&CWA appears as a first option in the list and BL&noCWA the second one.(PanelConceptJerarq.java).
- Reorganize the window of CCEC in such a way that the options that appears for the generation of the latex inform are “generar informació BC intermèdies” followed by the option “Amb descriptiva”.(PanelConceptJerarq.java).
- Some graphical problem fixed, like centered some frames at the screen. Or some labels that appeared cut in the graphic interface.
- Fixed the problem that overwrites the tex file generated by each action that should generate an output tex file. Now each action generates a tex file with a different name depending of the task.
- Re-implemented process of the generation of the latex inform. Now, the structure of the file are: Kb 1 Qt 1, Kb 2 Qt 2, Kb 3 Qt 3..., Kb final Qt final, D1, D2, D3...D final.

  Where: Kb -> Knowledge Base.

  Qt -> Quality Table.

  D -> Description of the Knowledge Base.

- Change the "simplified version" so that it appears as a flag to mark or not in the right hand side of this second option.(PanelConceptJerarq.java) .
- Changed version name for the project to “Java-KLASS v8.4 (2011)” and added the name of the two last developers (FinestraInicial.java) and (FrPrincipal_AboutBox.java).
- Change the "simplified version" so that it appears as a flag to mark or not in the right hand side of this second option.(PanelConceptJerarq.java) .
- Changed version name for the project to “Java-KLASS v8.4 (2011)” (FinestraInicial.java)
- Solve the compilation problem with Latex when clicking "D'acord" in the "Conceptualizació Jerarquica" panel. (Jordi)
- Solve the problem that provokes that, if you select several qualitative variables in the boxplot-based induction rules, it does not work, because probably the external loop

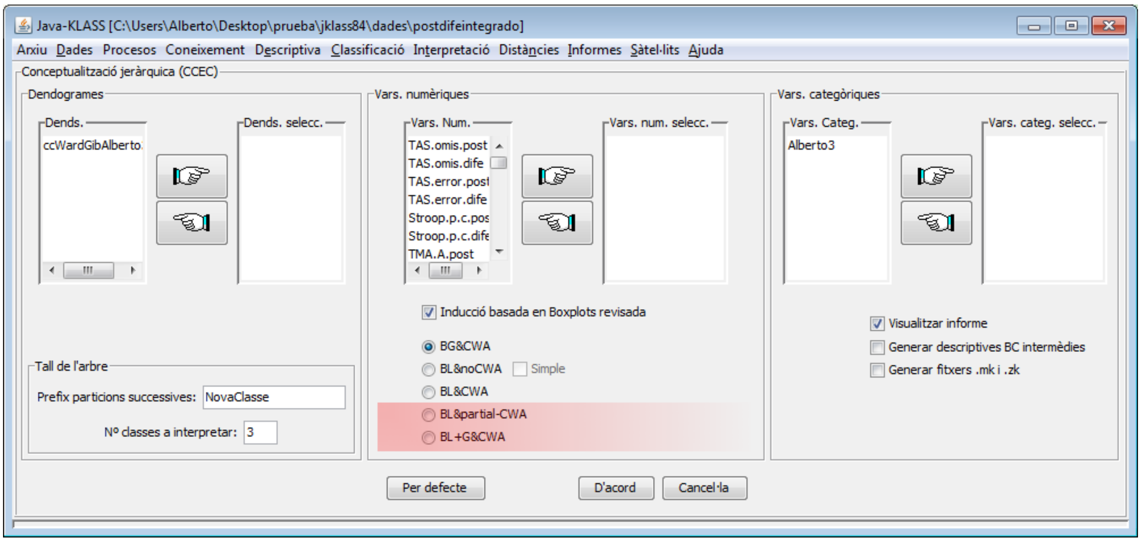
must be introduced into the method in such a way that it iterates the process over the whole set of class-variables selected by the user.

- Test BL&partial-CWA method.
- Test BL+G&CWA method.

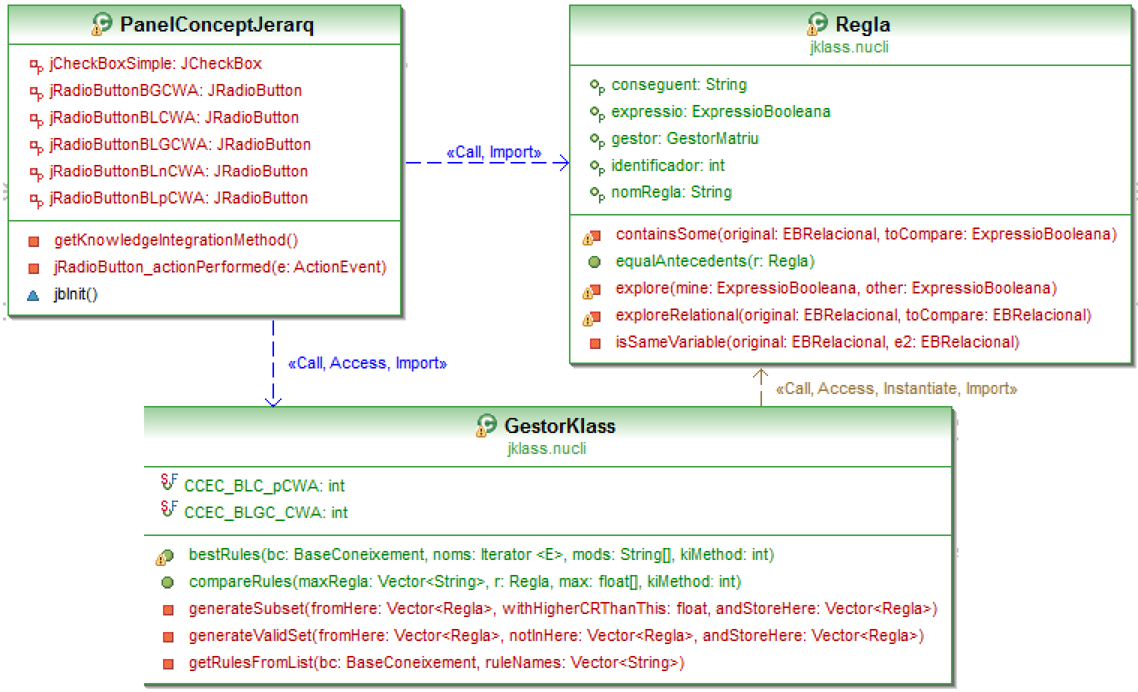
Implementation of the new strategies

- Implement BL&partial-CWA method (PanelConceptJerarq.java, GestorKlass.java and Regla.java).
- Implement BL+G&CWA method (PanelConceptJerarq.java, GestorKlass.java and Regla.java).

We see in the following image the final result in the user interface of this new strategies:



The way they were implemented was the following, implementing or modifying all the methods and variables that appear in the following UML diagram:



BestRules() is a method that receives a knowledge base of rules, and extract the best two rules in it, or makes a combination of the best ones, having each a different consequent.

The way BestRules manages this is by calling compareRules(), that gives the best one of two rules, and so comparing all of the rules to extract two sets, one for each target class, having all the rules the previously explained properties.

Testing

For testing purposes, we will use a subset of the variables in the “postdifeintegrado” dataset. This variables are:

- TAS:omis:post
- TAS:omis:dife
- Stroop:p:c:post
- B:d:d:dife
- WCST:e:pe:dife

We will do three iterations with each method, generating up to a total of 4 rules for each strategy. That will give us a sense of the good or bad results of our implementation.

All the Knowledge bases will be generated by cutting the dentrogram and making sequential binary classes, and for each pair of classes, performing a boxplot based induction of rules.

First iteration

BC0 :

$r0 : (TAS.omis.post \geq 0) \wedge (TAS.omis.post < 4) \xrightarrow{1} (ccWardGibAlberto30K2)C1$

$r1 : (TAS.omis.post \geq 4) \wedge (TAS.omis.post \leq 10) \xrightarrow{0,62} (ccWardGibAlberto30K2)C0$

$r2 : (TAS.omis.post \geq 4) \wedge (TAS.omis.post \leq 10) \xrightarrow{0,38} (ccWardGibAlberto30K2)C1$

$r3 : (TAS.omis.dife \geq -10) \wedge (TAS.omis.dife < -6) \xrightarrow{1} (ccWardGibAlberto30K2)C1$

$r4 : (TAS.omis.dife \geq -6) \wedge (TAS.omis.dife \leq 0) \xrightarrow{0,48} (ccWardGibAlberto30K2)C0$

$r5 : (TAS.omis.dife \geq -6) \wedge (TAS.omis.dife \leq 0) \xrightarrow{0,52} (ccWardGibAlberto30K2)C1$

$r6 : (TAS.omis.dife > 0) \wedge (TAS.omis.dife \leq 1) \xrightarrow{1} (ccWardGibAlberto30K2)C1$

$r7 : (Stroop.p.c.post \geq 0) \wedge (Stroop.p.c.post \leq 0) \xrightarrow{1} (ccWardGibAlberto30K2)C0$

$r8 : (Stroop.p.c.post \geq 14) \wedge (Stroop.p.c.post \leq 60) \xrightarrow{1} (ccWardGibAlberto30K2)C1$

$r9 : (B.d.d.dife \geq -1) \wedge (B.d.d.dife < 0) \xrightarrow{1} (ccWardGibAlberto30K2)C1$

$r10 : (B.d.d.dife \geq 0) \wedge (B.d.d.dife \leq 7) \xrightarrow{0,33} (ccWardGibAlberto30K2)C0$

$r11 : (B.d.d.dife \geq 0) \wedge (B.d.d.dife \leq 7) \xrightarrow{0,67} (ccWardGibAlberto30K2)C1$

$r12 : (B.d.d.dife > 7) \wedge (B.d.d.dife \leq 8) \xrightarrow{1} (ccWardGibAlberto30K2)C0$

$r13 : (WCST.e.pe.dife \geq -50) \wedge (WCST.e.pe.dife \leq 0) \xrightarrow{0,36} (ccWardGibAlberto30K2)C0$

$r14 : (WCST.e.pe.dife \geq -50) \wedge (WCST.e.pe.dife \leq 0) \xrightarrow{0,64} (ccWardGibAlberto30K2)C1$

$r15 : (WCST.e.pe.dife > 0) \wedge (WCST.e.pe.dife \leq 6) \xrightarrow{1} (ccWardGibAlberto30K2)C1$

First we select those rules with p(r)=1, and from that subset, we select those rules that have the maximum CovR.

BC0	Consecuent	p(r)	Sup	CovR
r7	(ccWardGibAlberto30K2)C0	1	0,3404	1
r12	(ccWardGibAlberto30K2)C0	1	0,0213	0,0625
r1	(ccWardGibAlberto30K2)C0	0,6154	0,5532	1
r4	(ccWardGibAlberto30K2)C0	0,4848	0,7021	1
r13	(ccWardGibAlberto30K2)C0	0,3556	0,9574	1
r10	(ccWardGibAlberto30K2)C0	0,3333	0,9574	0,9375

r8	(ccWardGibAlberto30K2)C1	1	0,6596	1
r0	(ccWardGibAlberto30K2)C1	1	0,4468	0,6774
r3	(ccWardGibAlberto30K2)C1	1	0,2553	0,3871
r6	(ccWardGibAlberto30K2)C1	1	0,0426	0,0645
r15	(ccWardGibAlberto30K2)C1	1	0,0426	0,0645
r9	(ccWardGibAlberto30K2)C1	1	0,0213	0,0323
r11	(ccWardGibAlberto30K2)C1	0,6667	0,9574	0,9677
r14	(ccWardGibAlberto30K2)C1	0,6444	0,9574	0,9355
r5	(ccWardGibAlberto30K2)C1	0,5152	0,7021	0,5484
r2	(ccWardGibAlberto30K2)C1	0,3846	0,5532	0,3226

In the table we see that only one rule from each class is good enough. This process is common to both strategies, so now we will define which should be the output for each:

BL&partial-CWA

As we have only identified one rule for each class, and both have as antecedent an interval of the same variable, we apply

$$A_i^{*\varepsilon+1} = A_i^{\varepsilon+1,k_i} \text{ (10.21)}$$

$$A_j^{*\varepsilon+1} = A_j^{\varepsilon+1,k_j} \text{ (10.22)}$$

And get the following rules

$$BC0: r7 \rightarrow C0$$

$$BC0: r8 \rightarrow C1$$

Our results

When we run our developed methods in Klass we obtain the following rules

CCECPrimeraIteracionBLpCWA :

$r1.BC0.BC0.r7 : (Stroop.p.c.post \geq 0) \wedge (Stroop.p.c.post \leq 0) \longrightarrow (PrimeraIteracionBLpCWA2)C0$

$r2.BC1.BC0.r8 : (Stroop.p.c.post \geq 14) \wedge (Stroop.p.c.post \leq 60) \longrightarrow (PrimeraIteracionBLpCWA2)C1$

We can see that our results are consistent with the test set, so until this point we can confirm that our methods work properly.

BL+G&CWA

We have only identified one rule for each class, both have as antecedent an interval of the same variable, and $CovR\left(r_{s_i,c_i}^{k_i}\right) \leq CovR\left(r_{s_j,c_j}^{k_j}\right)$

$$A_i^{*\varepsilon+1} = \neg A_j^{\varepsilon+1,k_j} \text{ (10.31)}$$

$$A_j^{*\varepsilon+1} = A_j^{\varepsilon+1,k_j} \text{ (10.32)}$$

And get the rules

$$BC0: not(r8) \rightarrow C0$$

$$BC0: r8 \rightarrow C1$$

Our results

CCECPrimeraIteracionBLGCWA :

$$r1.BC0.BC6.not(r8) : \neg((Stroop.p.c.post \geq 14) \wedge (Stroop.p.c.post \leq 60)) \longrightarrow (PrimeraIteracionBLGCW$$

$$r2.BC1.BC6.r8 : (Stroop.p.c.post \geq 14) \wedge (Stroop.p.c.post \leq 60) \longrightarrow (PrimeraIteracionBLGCWA2)C1$$

As we see, also this method behaves as expected in this first iteration.

Second Iteration

BC1 :

$$r0 : (TAS.omis.post \geq 4) \wedge (TAS.omis.post < 10) \xrightarrow{1} (VAR1)graves13$$

$$r1 : (TAS.omis.post \geq 10) \wedge (TAS.omis.post \leq 10) \xrightarrow{0{,}08} (VAR1)34$$

$$r2 : (TAS.omis.post \geq 10) \wedge (TAS.omis.post \leq 10) \xrightarrow{0{,}92} (VAR1)graves13$$

$$r3 : (TAS.omis.dife \geq -6) \wedge (TAS.omis.dife < 0) \xrightarrow{1} (VAR1)graves13$$

$$r4 : (TAS.omis.dife \geq 0) \wedge (TAS.omis.dife \leq 0) \xrightarrow{0{,}08} (VAR1)34$$

$$r5 : (TAS.omis.dife \geq 0) \wedge (TAS.omis.dife \leq 0) \xrightarrow{0{,}92} (VAR1)graves13$$

$$r6 : (Stroop.p.c.post \geq 0) \wedge (Stroop.p.c.post \leq 0) \xrightarrow{0{,}06} (VAR1)34$$

$$r7 : (Stroop.p.c.post \geq 0) \wedge (Stroop.p.c.post \leq 0) \xrightarrow{0{,}94} (VAR1)graves13$$

$$r8 : (B.d.d.dife \geq 0) \wedge (B.d.d.dife \leq 0) \xrightarrow{0{,}17} (VAR1)34$$

$$r9 : (B.d.d.dife \geq 0) \wedge (B.d.d.dife \leq 0) \xrightarrow{0{,}83} (VAR1)graves13$$

$$r10 : (B.d.d.dife > 0) \wedge (B.d.d.dife \leq 8) \xrightarrow{1} (VAR1)graves13$$

$$r11 : (WCST.e.pe.dife \geq -50) \wedge (WCST.e.pe.dife < 0) \xrightarrow{1} (VAR1)graves13$$

$$r12 : (WCST.e.pe.dife \geq 0) \wedge (WCST.e.pe.dife \leq 0) \xrightarrow{0{,}08} (VAR1)34$$

$$r13 : (WCST.e.pe.dife \geq 0) \wedge (WCST.e.pe.dife \leq 0) \xrightarrow{0{,}92} (VAR1)graves13$$

Once again, we select those rules with p(r)=1, and from that subset, we select those rules that have the maximum CovR.

BC1	Consecuent	p(r)	Sup	CovR
r8	(VAR1)34	0,1667	0,375	1
r1	(VAR1)34	0,0769	0,8125	1
r4	(VAR1)34	0,0769	0,8125	1
r12	(VAR1)34	0,0769	0,8125	1

r6	(VAR1)34	0,0625	1	1
r10	(VAR1)graves13	1	0,625	0,6667
r0	(VAR1)graves13	1	0,1875	0,2
r3	(VAR1)graves13	1	0,1875	0,2
r11	(VAR1)graves13	1	0,1875	0,2
r7	(VAR1)graves13	0,9375	1	1
r2	(VAR1)graves13	0,9231	0,8125	0,8
r5	(VAR1)graves13	0,9231	0,8125	0,8
r13	(VAR1)graves13	0,9231	0,8125	0,8
r9	(VAR1)graves13	0,8333	0,375	0,3333

In the table we see that there is no rule good enough from the first class, and only one rule from the second class is good enough. This process is common to both strategies, so now we will define which should be the output for each:

BL&partial-CWA

As we have only identified one rule for the second class, and none for the second one, they don't have as antecedent an interval of the same variable, so we apply

$$A_i^{*\varepsilon+1} = A_i^{\varepsilon+1,k_i} \vee \neg A_j^{\varepsilon+1,k_j} \quad (10.23)$$

$$A_j^{*\varepsilon+1} = A_j^{\varepsilon+1,k_j} \vee \neg A_i^{\varepsilon+1,k_i} \quad (10.24)$$

And get

$$BC1{:}\,NULL \bigvee BC1{:}\,not(r10) \Leftrightarrow BC1{:}\,not(r10) \rightarrow 34$$

$$BC1{:}\,r10 \bigvee BC1{:}\,not(NULL) \Leftrightarrow BC1{:}\,r10 \rightarrow graves13$$

Which leads finally to the following rules:

$$BC0{:}\,r7 \bigwedge BC1{:}\,not(r10) \rightarrow 34$$

$$BC0{:}\,r7 \bigwedge BC1{:}\,r10 \rightarrow graves13$$

$$BC0{:}\,r8 \rightarrow C1$$

Our results

$$\mathbf{CCECSegundaIteracionBLpCWA}\ :$$

$$r1.BC0.BC2.r8 : (Stroop.p.c.post \geq 14) \wedge (Stroop.p.c.post \leq 60) \longrightarrow (SegundaIteracionBLpCWA2)C1$$

$$r2.BC1.BC2.r7BC3.r0{-}not(r8) : ((Stroop.p.c.post \geq 0) \wedge (Stroop.p.c.post \leq 0)) \wedge (((TAS.omis.post \geq 4) \wedge (TAS.omis.post < 10)) \vee (\neg((B.d.d.dife > 0) \wedge (B.d.d.dife \leq 8)))) \longrightarrow (SegundaIteracionBLpCWA3)34$$

$$r3.BC2.BC2.r7BC3.r8{-}not(r0) : ((Stroop.p.c.post \geq 0) \wedge (Stroop.p.c.post \leq 0)) \wedge (((B.d.d.dife > 0) \wedge (B.d.d.dife \leq 8)) \vee (\neg((TAS.omis.post \geq 4) \wedge (TAS.omis.post < 10)))) \longrightarrow (SegundaIteracionBLpCWA3)$$

We see that the results do not coincide.

The problem here is that during the processing, it uses a different Knowledge base than the one we used for the testing specification, that is generated outside our methods. It is the following one (we copypaste directly from the results obtained from Klass, as it does not compile properly):

```
$r0: (TAS.omis.post \geq 4) \wedge (TAS.omis.post <
10)\stackrel{0,06}{\longrightarrow} (NovaClasse3)34$\\
$r1: (TAS.omis.post \geq 4) \wedge (TAS.omis.post <
10)\stackrel{0,94}{\longrightarrow} (NovaClasse3)graves13$\\
$r2: (TAS.omis.dife \geq -6) \wedge (TAS.omis.dife <
0)\stackrel{0,06}{\longrightarrow} (NovaClasse3)34$\\
$r3: (TAS.omis.dife \geq -6) \wedge (TAS.omis.dife <
0)\stackrel{0,94}{\longrightarrow} (NovaClasse3)graves13$\\
$r4: (Stroop.p.c.post \geq 0) \wedge (Stroop.p.c.post <
0)\stackrel{0,06}{\longrightarrow} (NovaClasse3)34$\\
$r5: (Stroop.p.c.post \geq 0) \wedge (Stroop.p.c.post <
0)\stackrel{0,94}{\longrightarrow} (NovaClasse3)graves13$\\
$r6: (B.d.d.dife \geq 0) \wedge (B.d.d.dife <
0)\stackrel{0,17}{\longrightarrow} (NovaClasse3)34$\\
$r7: (B.d.d.dife \geq 0) \wedge (B.d.d.dife <
0)\stackrel{0,83}{\longrightarrow} (NovaClasse3)graves13$\\
$r8: (B.d.d.dife>0) \wedge (B.d.d.dife \leq
8)\stackrel{1}{\longrightarrow} (NovaClasse3)graves13$\\
$r9: (WCST.e.pe.dife \geq -50) \wedge (WCST.e.pe.dife <
0)\stackrel{0,06}{\longrightarrow} (NovaClasse3)34$\\
$r10: (WCST.e.pe.dife \geq -50) \wedge (WCST.e.pe.dife <
0)\stackrel{0,94}{\longrightarrow} (NovaClasse3)graves13$\\
```

Given this Knowledge base, our results make total sense. The only question now is why does Klass generate two different Knowledge bases from the same data when you generate them in different parts of the system.

BL+G&CWA

We have only identified one rule for for the second class, and none for the second one

$$A_i^{*\varepsilon+1} = A_i^{\varepsilon+1,k_i} \vee \neg A_j^{\varepsilon+1,k_j} \quad (10.23)$$

$$A_j^{*\varepsilon+1} = A_j^{\varepsilon+1,k_j} \vee \neg A_i^{\varepsilon+1,k_i} \quad (10.24)$$

And get

$$BC1: NULL \bigvee BC1: not(r10) \Leftrightarrow BC1: not(r10) \rightarrow 34$$

$$BC1: r10 \bigvee BC1: not(NULL) \Leftrightarrow BC1: r10 \rightarrow graves13$$

Which finally leads to

$$BC0: not(r8) \bigwedge BC1: not(r10) \rightarrow 34$$

$$BC0: not(r8) \bigwedge BC1: r10 \rightarrow graves13$$

$$BC0: r8 \rightarrow C1$$

Our results

CCECTerceraIteracionBLGCWA :

$r1.BC0.BC0.not(r8) : \neg((Stroop.p.c.post \geq 14) \wedge (Stroop.p.c.post \leq 60)) \longrightarrow (TerceraIteracionBLGCW$

$r2.BC1.BC0.r8 : (Stroop.p.c.post \geq 14) \wedge (Stroop.p.c.post \leq 60) \longrightarrow (TerceraIteracionBLGCWA2)C1$

$r3.BC2.BC1.r0 - not(r8) : ((TAS.omis.post \geq 4) \wedge (TAS.omis.post < 10)) \vee (\neg((B.d.d.dife > 0) \wedge (B.d.d.dife \leq 8))) \longrightarrow (TerceraIteracionBLGCWA3)34$

$r4.BC3.BC1.r8 - not(r0) : ((B.d.d.dife > 0) \wedge (B.d.d.dife \leq 8)) \vee (\neg((TAS.omis.post \geq 4) \wedge (TAS.omis.pos$ $10))) \longrightarrow (TerceraIteracionBLGCWA3)graves13$

For some reason for this method the rule for class C0 was not erased (the first one). Anyway, the last two rules are obviously generated by the good logic but with the same Knowledge base than for the previous strategy.

Third Iteration

BC2 :

$$r0 : (TAS.omis.post \geq 4) \wedge (TAS.omis.post < 10) \xrightarrow{1} (VAR2)graves9$$

$$r1 : (TAS.omis.post \geq 10) \wedge (TAS.omis.post \leq 10) \xrightarrow{0{,}58} (VAR2)graves9$$

$$r2 : (TAS.omis.post \geq 10) \wedge (TAS.omis.post \leq 10) \xrightarrow{0{,}42} (VAR2)graves12$$

$$r3 : (TAS.omis.dife \geq -6) \wedge (TAS.omis.dife < 0) \xrightarrow{1} (VAR2)graves9$$

$$r4 : (TAS.omis.dife \geq 0) \wedge (TAS.omis.dife \leq 0) \xrightarrow{0{,}58} (VAR2)graves9$$

$$r5 : (TAS.omis.dife \geq 0) \wedge (TAS.omis.dife \leq 0) \xrightarrow{0{,}42} (VAR2)graves12$$

$$r6 : (Stroop.p.c.post \geq 0) \wedge (Stroop.p.c.post \leq 0) \xrightarrow{0{,}67} (VAR2)graves9$$

$$r7 : (Stroop.p.c.post \geq 0) \wedge (Stroop.p.c.post \leq 0) \xrightarrow{0{,}33} (VAR2)graves12$$

$$r8 : (B.d.d.dife \geq 0) \wedge (B.d.d.dife \leq 0) \xrightarrow{1} (VAR2)graves12$$

$$r9 : (B.d.d.dife \geq 4) \wedge (B.d.d.dife \leq 8) \xrightarrow{1} (VAR2)graves9$$

$$r10 : (WCST.e.pe.dife \geq -50) \wedge (WCST.e.pe.dife < 0) \xrightarrow{1} (VAR2)graves9$$

$$r11 : (WCST.e.pe.dife \geq 0) \wedge (WCST.e.pe.dife \leq 0) \xrightarrow{0{,}58} (VAR2)graves9$$

$$r12 : (WCST.e.pe.dife \geq 0) \wedge (WCST.e.pe.dife \leq 0) \xrightarrow{0{,}42} (VAR2)graves12$$

One more time, we select those rules with p(r)=1, and from that subset, we select those rules that have the maximum CovR.

BC2	Consecuent	p(r)	Sup	CovR
r8	(VAR2)graves12	1	0,3333	1
r2	(VAR2)graves12	0,4167	0,8	1
r5	(VAR2)graves12	0,4167	0,8	1
r12	(VAR2)graves12	0,4167	0,8	1

r7	(VAR2)graves12	0,3333	1	1
r9	(VAR2)graves9	1	0,6667	1
r0	(VAR2)graves9	1	0,2	0,3
r3	(VAR2)graves9	1	0,2	0,3
r10	(VAR2)graves9	1	0,2	0,3
r6	(VAR2)graves9	0,6667	1	1
r1	(VAR2)graves9	0,5833	0,8	0,7
r4	(VAR2)graves9	0,5833	0,8	0,7
r11	(VAR2)graves9	0,5833	0,8	0,7

In the table we see that only one rule from each class is good enough. This process is common to both strategies, so now we will define which should be the output for each:

BL&partial-CWA

As we have only identified one rule for each class, and both have as antecedent an interval of the same variable, we apply again

$$A_i^{*\varepsilon+1} = A_i^{\varepsilon+1,k_i} \text{ (10.21)}$$

$$A_j^{*\varepsilon+1} = A_j^{\varepsilon+1,k_j} \text{ (10.22)}$$

And get

$$BC2: r8 \rightarrow graves12$$

$$BC2: r9 \rightarrow graves9$$

To finally get

$$BC0: r7 \bigwedge BC1: not(r10) \rightarrow 34$$

$$BC0: r7 \bigwedge BC1: r10 \bigwedge BC2: r8 \rightarrow graves12$$

$$BC0: r7 \bigwedge BC1: r10 \bigwedge BC2: r9 \rightarrow graves9$$

$$BC0: r8 \rightarrow C1$$

Our results

$$\mathbf{CCECTerceraIteracionBLpCWA}\ :$$

$$r1.BC0.BC4.r8 : (Stroop.p.c.post \geq 14) \wedge (Stroop.p.c.post \leq 60) \longrightarrow (TerceraIteracionBLpCWA2)C1$$

$$r2.BC1.BC4.r7BC5.r0{-}not(r8) : ((Stroop.p.c.post \geq 0) \wedge (Stroop.p.c.post \leq 0)) \wedge (((TAS.omis.post \geq 4) \wedge (TAS.omis.post < 10)) \vee (\neg((B.d.d.dife > 0) \wedge (B.d.d.dife \leq 8)))) \longrightarrow (TerceraIteracionBLpCWA3)34$$

$$r3.BC2.BC4.r7BC5.r8{-}not(r0)BC6.r8{-}not(r1) : (((Stroop.p.c.post \geq 0) \wedge (Stroop.p.c.post \leq 0)) \wedge (((B.d.d.dife > 0) \wedge (B.d.d.dife \leq 8)) \vee (\neg((TAS.omis.post \geq 4) \wedge (TAS.omis.post < 10))))) \wedge (((B.d.d.dife > 4) \wedge (B.d.d.dife \leq 8)) \vee (\neg((TAS.omis.post \geq 4) \wedge (TAS.omis.post < 10)))) \longrightarrow (TerceraIteracionBLpCWA4)g$$

$$r4.BC3.BC4.r7BC5.r8{-}not(r0)BC6.r1{-}not(r8) : (((Stroop.p.c.post \geq 0) \wedge (Stroop.p.c.post \leq 0)) \wedge (((B.d.d.dife > 0) \wedge (B.d.d.dife \leq 8)) \vee (\neg((TAS.omis.post \geq 4) \wedge (TAS.omis.post < 10))))) \wedge (((TAS.omis.post\ 4) \wedge (TAS.omis.post < 10)) \vee (\neg((B.d.d.dife > 4) \wedge (B.d.d.dife \leq 8)))) \longrightarrow (TerceraIteracionBLpCWA4)grave$$

The problem the same as before, it uses again a different Knowledge base:

`$r0: (TAS.omis.post \geq 4) \wedge (TAS.omis.post < 10)\stackrel{0,67}{\longrightarrow} (TerceraIteracionBLpCWA4)graves9$\\`
`$r1: (TAS.omis.post \geq 4) \wedge (TAS.omis.post < 10)\stackrel{0,33}{\longrightarrow} (TerceraIteracionBLpCWA4)graves12$\\`
`$r2: (TAS.omis.dife \geq -6) \wedge (TAS.omis.dife < 0)\stackrel{0,67}{\longrightarrow} (TerceraIteracionBLpCWA4)graves9$\\`
`$r3: (TAS.omis.dife \geq -6) \wedge (TAS.omis.dife < 0)\stackrel{0,33}{\longrightarrow} (TerceraIteracionBLpCWA4)graves12$\\`
`$r4: (Stroop.p.c.post \geq 0) \wedge (Stroop.p.c.post < 0)\stackrel{0,67}{\longrightarrow} (TerceraIteracionBLpCWA4)graves9$\\`
`$r5: (Stroop.p.c.post \geq 0) \wedge (Stroop.p.c.post < 0)\stackrel{0,33}{\longrightarrow} (TerceraIteracionBLpCWA4)graves12$\\`
`$r6: (B.d.d.dife \geq 0) \wedge (B.d.d.dife < 0)\stackrel{1}{\longrightarrow} (TerceraIteracionBLpCWA4)graves12$\\`
`$r7: (B.d.d.dife \geq 0) \wedge (B.d.d.dife \leq 4)\stackrel{1}{\longrightarrow} (TerceraIteracionBLpCWA4)graves9$\\`
`$r8: (B.d.d.dife>4) \wedge (B.d.d.dife \leq 8)\stackrel{1}{\longrightarrow} (TerceraIteracionBLpCWA4)graves9$\\`
`$r9: (WCST.e.pe.dife \geq -50) \wedge (WCST.e.pe.dife < 0)\stackrel{0,67}{\longrightarrow} (TerceraIteracionBLpCWA4)graves9$\\`
`$r10: (WCST.e.pe.dife \geq -50) \wedge (WCST.e.pe.dife < 0)\stackrel{0,33}{\longrightarrow}`

```
(TerceraIteracionBLpCWA4)graves12$\\
```

BL+G&CWA

We have only identified one rule for each class, both have as antecedent an interval of the same variable, and $ovR\left(r_{s_i,c_i}^{k_i}\right) \leq CovR\left(r_{s_j,c_j}^{k_j}\right)$, once again

$$A_i^{*\varepsilon+1} = \neg A_j^{\varepsilon+1,k_j} \text{ (10.31)}$$

$$A_j^{*\varepsilon+1} = A_j^{\varepsilon+1,k_j} \text{ (10.32)}$$

And get

$$BC2{:}\, not(r9) \rightarrow graves12$$

$$BC2{:}\, r9 \rightarrow graves9$$

To finally get

$$BC0{:}\, not(r8) \bigwedge BC1{:}\, not(r10) \rightarrow 34$$

$$BC0{:}\, not(r8) \bigwedge BC1{:}\, r10 \bigwedge BC2{:}\, not(r9) \rightarrow graves12$$

$$BC0{:}\, not(r8) \bigwedge BC1{:}\, r10 \bigwedge BC2{:}\, r9 \rightarrow graves9$$

$$BC0{:}\, r8 \rightarrow C1$$

Our results

CCECNovaClasse :

$r1.BC0.BC0.not(r8) : \neg((Stroop.p.c.post \geq 14) \wedge (Stroop.p.c.post \leq 60)) \longrightarrow (NovaClasse2)C1$

$r2.BC1.BC0.r8 : (Stroop.p.c.post \geq 14) \wedge (Stroop.p.c.post \leq 60) \longrightarrow (NovaClasse2)C1$

$r3.BC2.BC1.r0 - not(r8) : ((TAS.omis.post \geq 4) \wedge (TAS.omis.post < 10)) \vee (\neg((B.d.d.dife > 0) \wedge (B.d.d.dife \leq 8))) \longrightarrow (NovaClasse3)34$

$r4.BC3.BC1.r8{-}not(r0)BC2.r8{-}not(r1) : (((B.d.d.dife > 0) \wedge (B.d.d.dife \leq 8)) \vee (\neg((TAS.omis.post \geq 4) \wedge (TAS.omis.post < 10)))) \wedge (((B.d.d.dife > 4) \wedge (B.d.d.dife \leq 8)) \vee (\neg((TAS.omis.post \geq 4) \wedge (TAS.omis.post < 10)))) \longrightarrow (NovaClasse4)graves9$

$r5.BC4.BC1.r8{-}not(r0)BC2.r1{-}not(r8) : (((B.d.d.dife > 0) \wedge (B.d.d.dife \leq 8)) \vee (\neg((TAS.omis.post \geq 4) \wedge (TAS.omis.post < 10)))) \wedge (((TAS.omis.post \geq 4) \wedge (TAS.omis.post < 10)) \vee (\neg((B.d.d.dife > 4) \wedge (B.d.d.dife \leq 8)))) \longrightarrow (NovaClasse4)graves12$

Here, once again, the same problems.

Conclusions

Klass is a very unstable software platform in which too many non-expert hands seem to have been messing up. Methods developed in previous versions didn't behave as they where supposed to, contradictory results can be obtained supposedly using the same methods from different parts of the program, programming errors that not even a first year student is supposed to commit, different non cohesive programming paradigms along the code, the same things are calculated several times in different parts of the code just because of the misunderstandings of the previous programmers…

Under this circumstances, programming Klass has been a really intricate maze. With a little more time and not such an outstanding number of setbacks, our methods would have ended working better than any other part of the code, but that has not been the case.

Anyway, solving them is now not much of a problem. The first method seems to work perfectly fine, the only problem it has is produced in the method that calls it. On the other hand, the second method may need a bit more fine-tuning, but anyway the solution should be quite straightforward.

Nevertheless, a huge lot of improvements have been made to the code, lots of errors solved, many of the original inconsistencies are now history, and so, we leave a much better piece of software to the people coming after us.

References

1. *Automatic generation of conceptual descriptions of classifications in environmental domains.* **Pérez-Bonilla, A., Gibert, K. and Vrecko, D.** 2008. Procs.iEMSs. pp. volume 3, pages 1791–1798.

2. *Towards automatic generation of conceptual interpretation of clustering.* **Pérez-Bonilla, A and Gibert, K.** s.l. : Springer, 2007., 2008. LNCS. pp. 653–663, volume 4756.

3. **Pérez Bonilla, Alejandra A. and Gibert, K.** *Metodología de Caracterización Conceptual por Condicionamientos Sucesivos. Una aplicación a sistemas medioambientales.* Barcelona, Spain : Departament d'Estadística i Investigació Operativa. Universitat Politécnica de Catalunya, 2009.

4. **Gibert, K.** *L'us de la Informació Simbòlica en l'Automatització del Tractament Estadístic de Dominis Poc Estructurats.* Barcelona, Spain : Statistics and operations research phd. Thesis. Universitat Politecnica de Catalunya, 1994.

5. **Fiz, Beltran, Oliver, Roc and Maragall, Narcís.** *Assignment 02 - jKlass.* Barcelona, Spain : Intelligence Decision Support Systems. Universitat Politécnica de Catalunya., 2010.

6. **Gibert, K.** *Klass. estudi d'un sistema d'ajuda al tractament estadístic de grans bases de dades.* Barcelona : UPC, 1991.

7. **Castillejo, X.** *Un entorn de treball per a Klass.* Barcelona : PFC Facultat d'Informàtica, UPC, 1996.

8. **Márquez, J. and Martín, J.C.** *La clasificación automática en las ciencias de la salud.* Barcelona, Spain : PFC. Facultad de Matemàtiques i Estadística. UPC., 1997.

9. **Tubau, X.** s.l. : PFC, 1999.

10. **Ralambondrainy, Henri.** *A clustering method for nominal data and mixture of numerical and nominal data. Clasifiation and Related Methods of Data Analysis.* North-Holland : H.H.Bock, Elsevier Science Publishers, B.V., 1988.

11. **Bayona, S.** *Descriptiva de dades y de classes. .* Barcelona : PFC Facultat d'Informàtica. UPC., 2000.

12. **Oliveras Castellà, Josep.** *Dades heterogènies amb classificació automàtica. Implementació i comparativa de mètriques mixtes.* s.l. : PFC.

13. *A General coeficient of similarity and some of its properties.* **Gower, J.C.** s.l. : Biometrics, 1971, Vols. 27; 857-874.

14. *Some distance properties of latent root and vector methods used in multivariate analysis.* **Gower, J.C.** s.l. : Biomedika, 1966, Vols. 53:315-328.

15. *A comparison of some methods of cluster analysis.* **Gower, J.C.** s.l. : Biometrix, 1967, Vols. 23(4):623-37.

16. *Symbolic clustering using a new similaritiy measure.* **E. Diday, E. and Gowda, K.C.** s.l. : IEEE Trans. on systems, man.;and cib., 1992, Vols. 22, pages 368-378.

17. *Symbolic clustering using a new dissimilarity measure.* **Diday, E. and Gowda, K.C.** 1991, Vols. 24(6):567-578.

18. *Generalized Minkowski metrics for mixed features.* **Ichino, M. and Yaguchi, H.** Japan : IEICE Japó, 1989, Vols. J72-A(2):398{405.

19. **Rodas Osollo, Jorge Enrique.** *Knowledge Discovery in repeated and very short serial measures with a blocking factor.* Barcelona, Spain : Programa de doctorado: Inteligencia artificial. Universitat Politecnica de Catalunya., 2003.

20. *Robustness of class prediction depending on reference partition in ill-structured domains.* **Vázquez, F. and Gibert, K.** Sevilla : XVIII Iberoamerican Conference on Artificial Intelligence, Workshop de Minería de Datos y Aprendizaje (IBERAMIA2002), 2002, Vols. pages 13-22.