# 1 Introduction

The algorithms for association rules extraction were originally developed to find the logical association between variables with the same status. The predictive association rules search the associations between items that characterize a dependent attribute. We are in a supervised learning framework.

Basically, the algorithm is not really modified. Exploration is just limited to itemsets that include the dependent variable. The computation time is then reduced. Two components of Tanagra are dedicated to this task: these are ASSOC SPV and SPV RULE TREE ASSOC. They are available in the ASSOCIATION tab.

Compared to conventional approaches, the components of Tanagra have an additional specificity: we can specify the class value ("dependent variable = value") that we want to predict. The advantage is that we can set precisely the parameters of the algorithm, directly in relation to the characteristics of data. This is crucial for instance when the prior probabilities of the dependent variable values are very different.

We had already described the SPV TREE ASSOC component elsewhere[1]. But it was in the context of multivariate characterization of groups of individuals (from a clustering algorithm for instance). We compare it to the GROUP CHARACTERIZATION component. In this tutorial, we will compare the behavior of TREE ASSOC SPV and SPV RULE ASSOC during a prediction task. We will put forward their shared properties, the problems that they can handle, and their differences. SPV ASSOC RULE, which supplies original rule interestingness measures[2] ("test value" indicator), has the ability to simplify the rule base.

# 2 Dataset

We use a modified version of the GERMAN CREDIT dataset[3]. It describes the characteristics of customers. We discretized the quantitative variables. The file is available on line (http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/credit_assoc.xls).

CLASS is the dependent variable. We want to characterize the “good” customers (CLASS = GOOD). We therefore have two settings to set before the calculations: we indicate that CLASS is the TARGET variable; among the values of CLASS, we want to characterize the GOOD value.

# 3 Creating a diagram

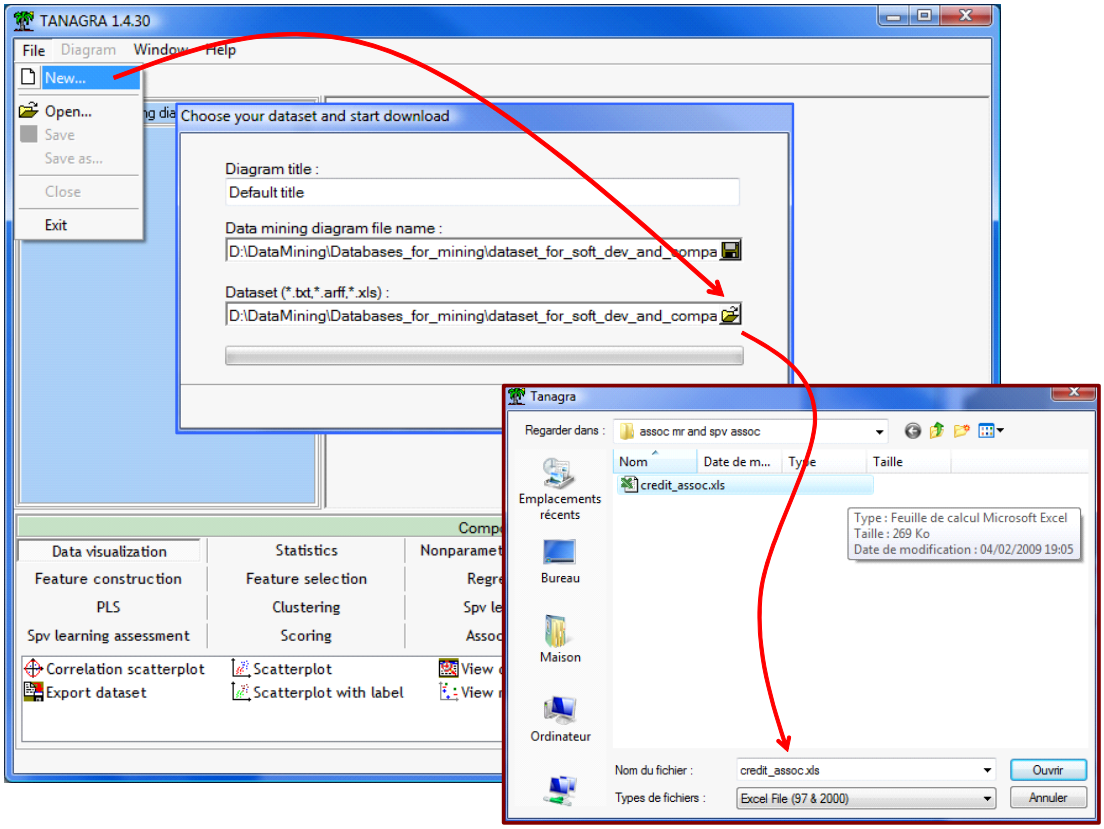
**Importing the dataset**. First, we define a new project (FILE / NEW) and we import the dataset. Tanagra can handle directly the Excel file format (XLS). We select the CREDIT_ASSOC.XLS.

Tanagra shows that 17 variables and 1000 examples were loaded.

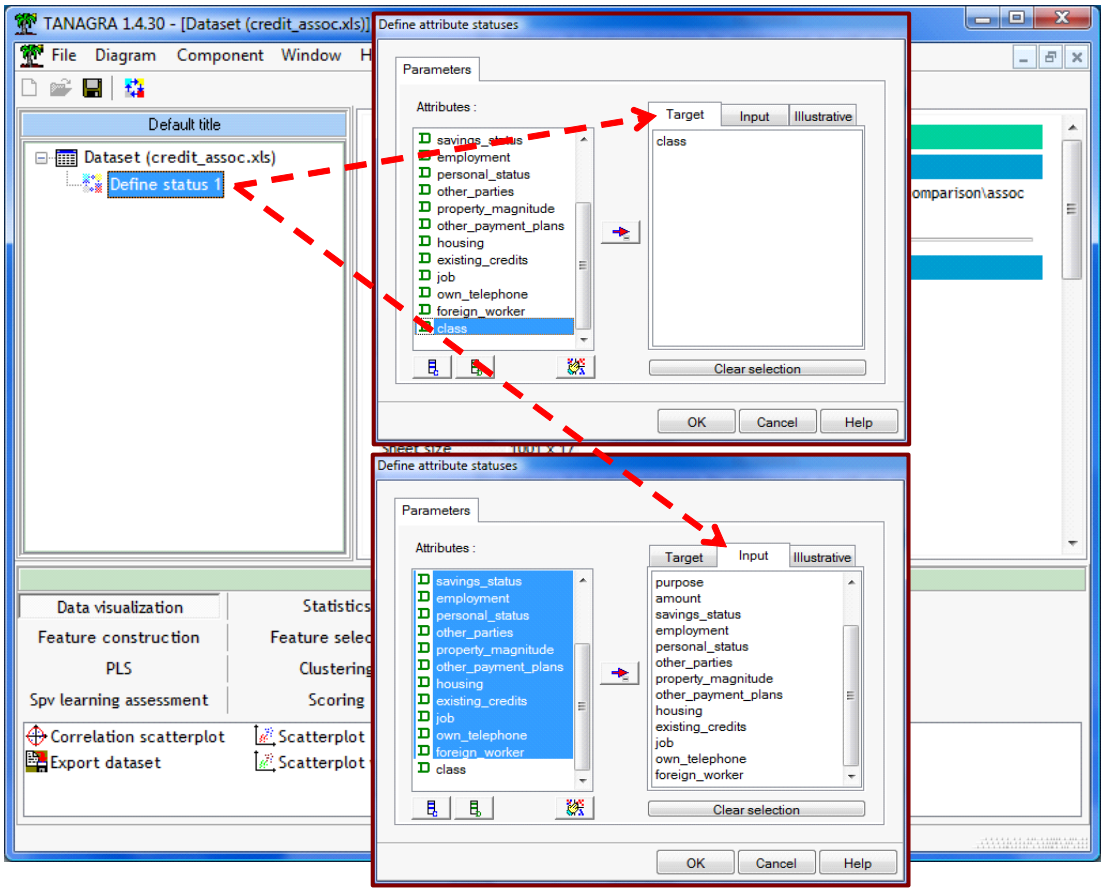
[1] http://data-mining-tutorials.blogspot.com/2008/11/supervised-association-rules.html

[2] http://data-mining-tutorials.blogspot.com/2009/02/interestingness-measures-for.html

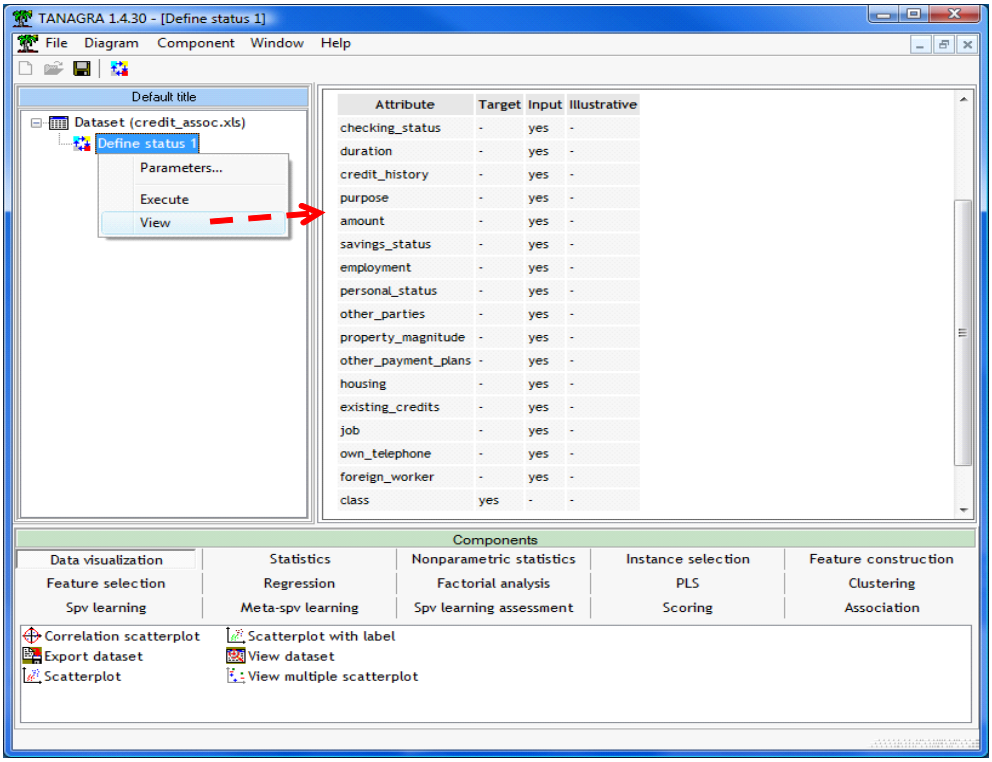
[3] http://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)




**Defining the type of attributes**. We insert the DEFINE STATUS component from the shortcut into the tool bar in order to define the type of attributes: CLASS is the TARGET attribute, the others are the INPUT.



We click on the VIEW menu. Here is the output of Tanagra.

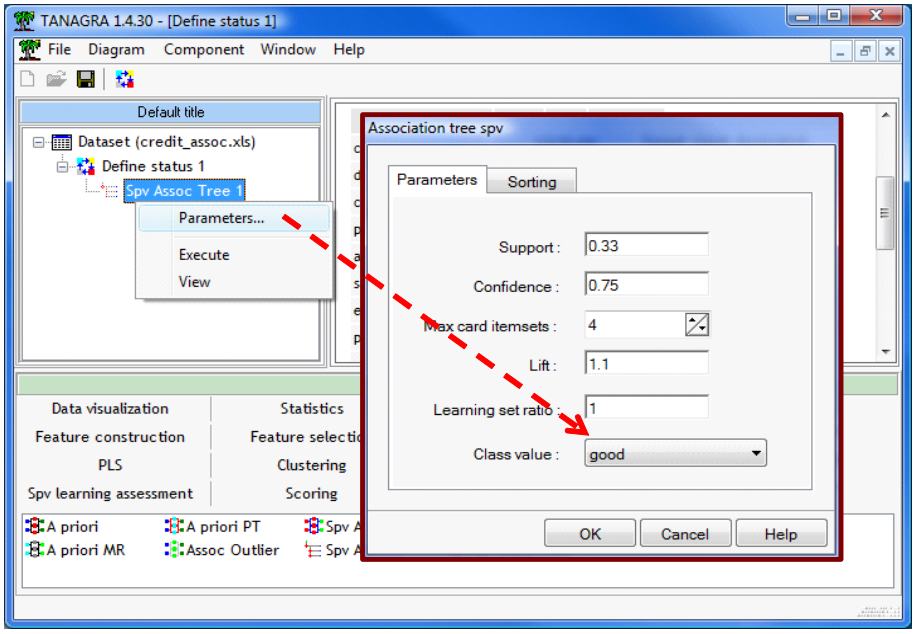



# 4 The SPV ASSOC TREE component

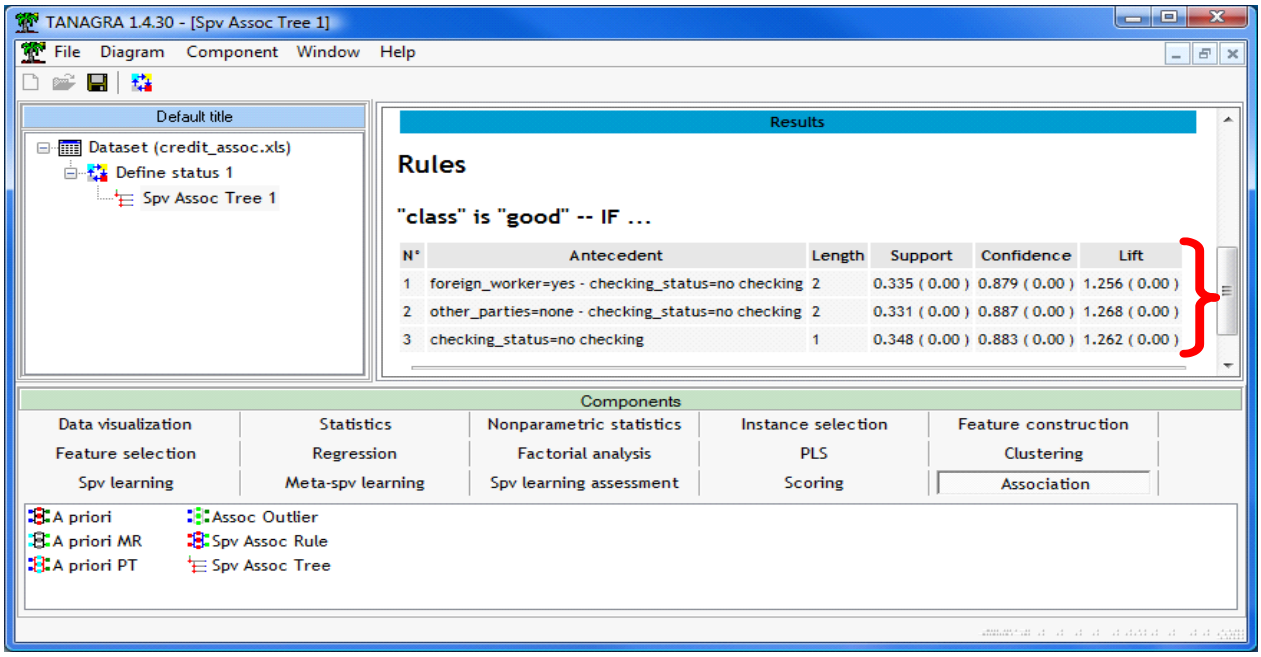
The SPV ASSOC TREE component extracts association rules from dataset. The procedure uses internally a search tree but the outputs are rules.

## 4.1 Choosing the class value

We insert SPV ASSOC TREE into the diagram. We click on the PARAMETERS contextual menu. We must set the class value that we want to characterize. Into the dialog box, we set CLASS VALUE = GOOD.



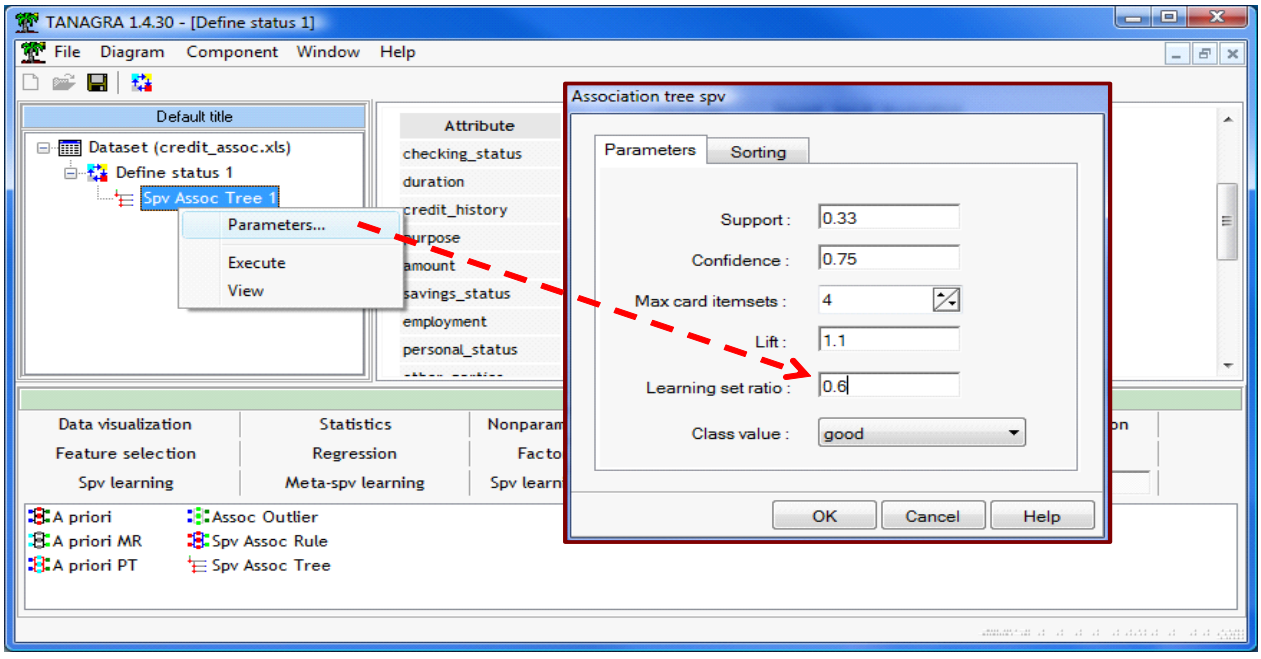

We validate and we click on the VIEW menu in order to execute the calculations.



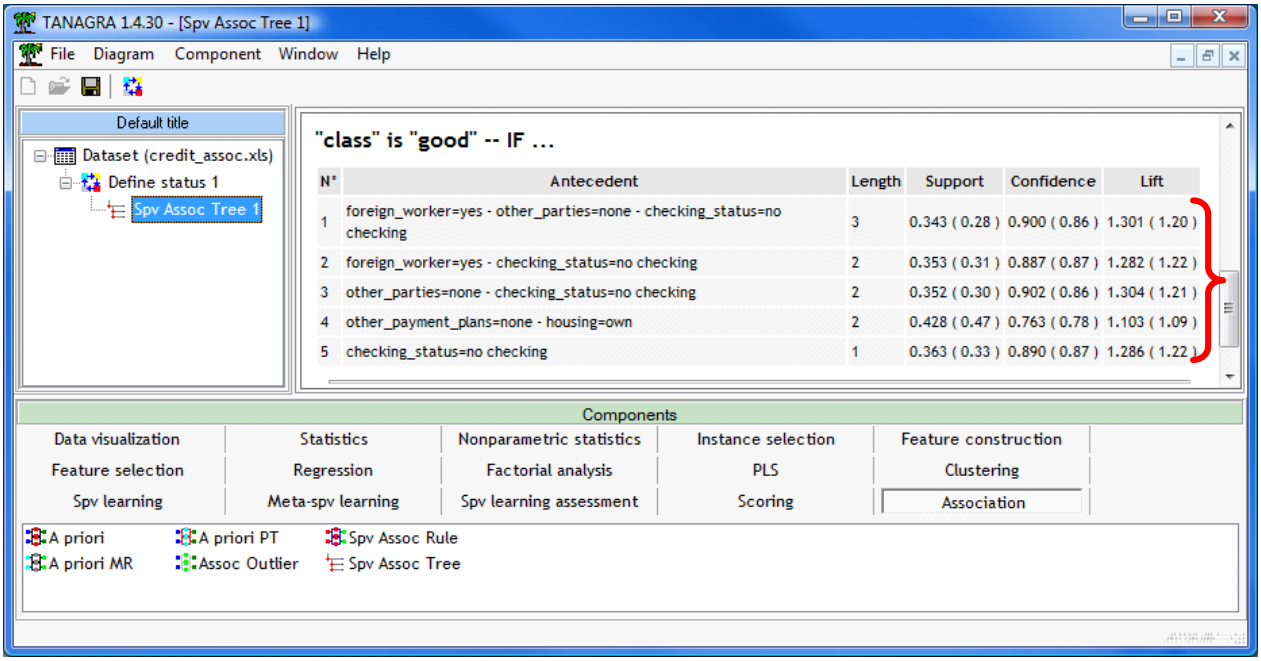

The component generates 3 rules. They are displayed in the lower part of the window. The support, confidence and lift are provided. When the numerical indicators are in parentheses, it means that the indicator was calculated on a test sample, which was not used during the learning phase. Here, we have systematically zero. All observations belong to the learning set.

## 4.2 Partitioning the dataset in a "train set" and "test set"

According to the approach usually implemented in a supervised learning framework, we can subdivide the dataset in a train set and a test set. The first is used during the extraction of the rules from data; the second is used for the assessment of the rules. We know that the measures computed on the test set give an honest estimate of their interestingness. In order to subdivide the dataset, we click again on the PARAMETERS menu. We set the LEARNING SET RATIO to 0.6 i.e. 60 percent of the dataset are used as a train set, 40 percent as a test set.



We validate this setting and we click on the VIEW menu.




We note that we obtain more rules without modifying the other settings. It is an artifact. It means that some rules are very near to the support (rule n°1), confidence (rule n°4) and lift threshold values.

We note also that we have more indications about the reliability of the rules now. The measures computed on the test set are less optimistic.

## 4.3 Ranking the rules

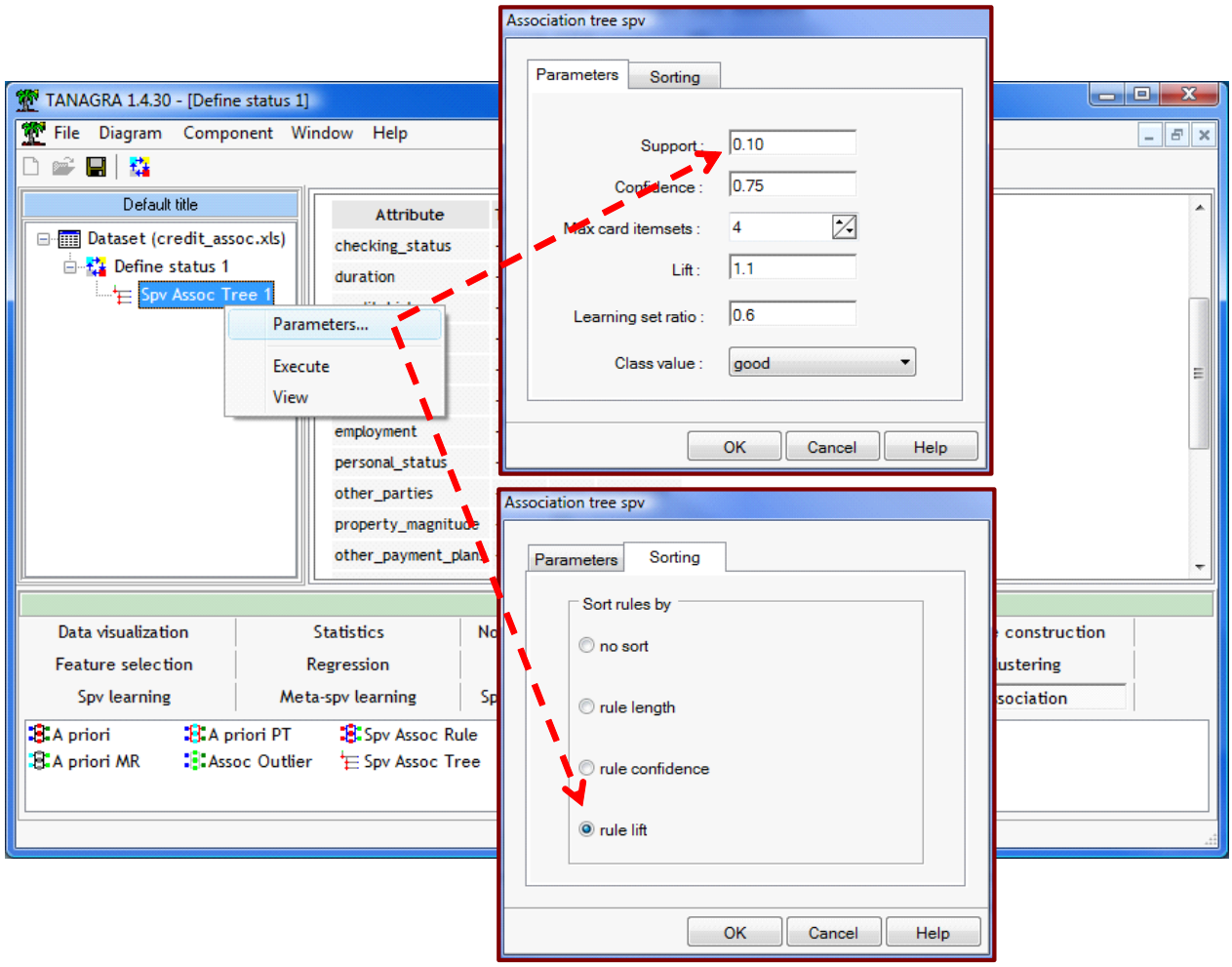
The settings can heavily influence the results. Some important rules can be hidden by inappropriate threshold values. An alternative approach is to set less restrictive settings to obtain more rules and organize them in order to highlight the most informative rules.

The available parameters are:

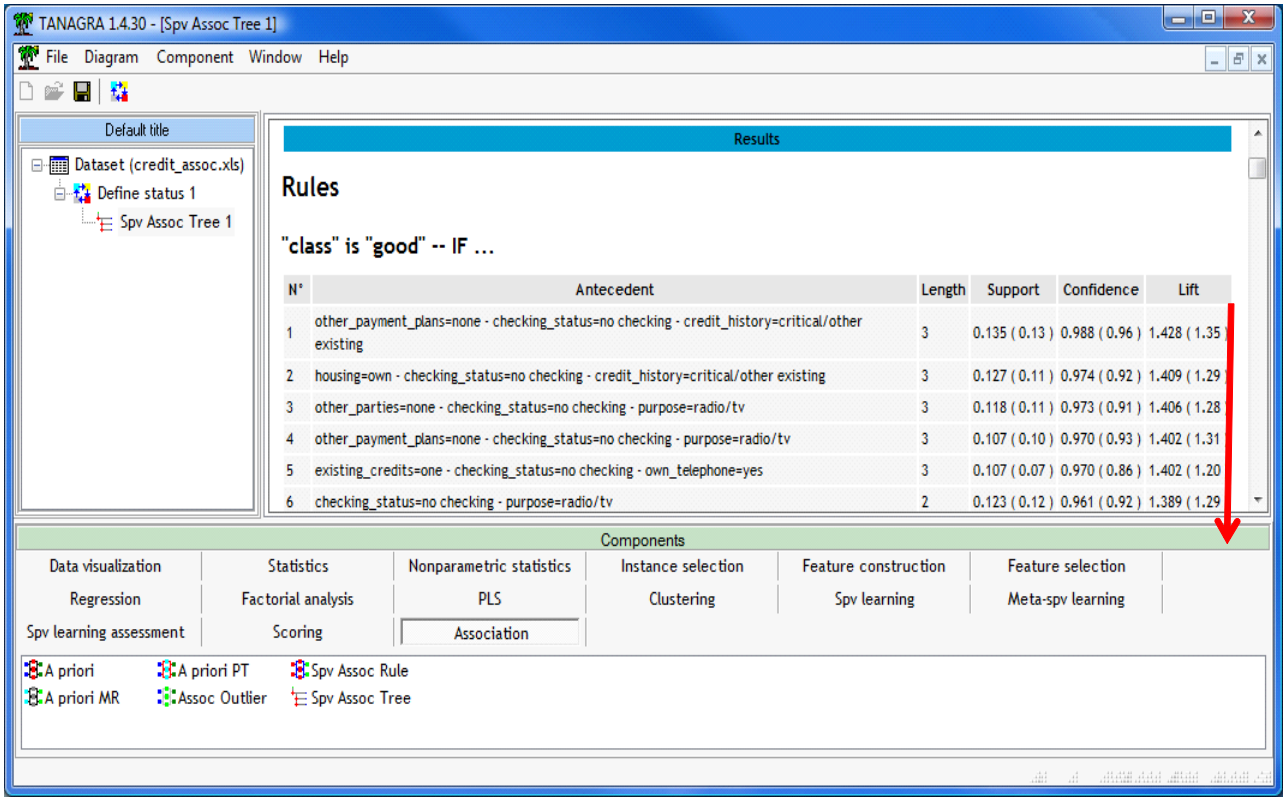
- SUPPORT defines the minimum support of the extracted rules ;
- CONFIANCE defines the minimum confidence ;
- MAX CARD ITEMSETS defines the maximum length ;
- LIFT defines the minimum lift.

Once the rules extracted, we need to organize so that the most interesting appear first. Tanagra can rank the rules according to one of the numerical criteria above. In this tutorial, we rank them according to the lift criterion.

We click on the PARAMETERS menu.




We obtain the following results.



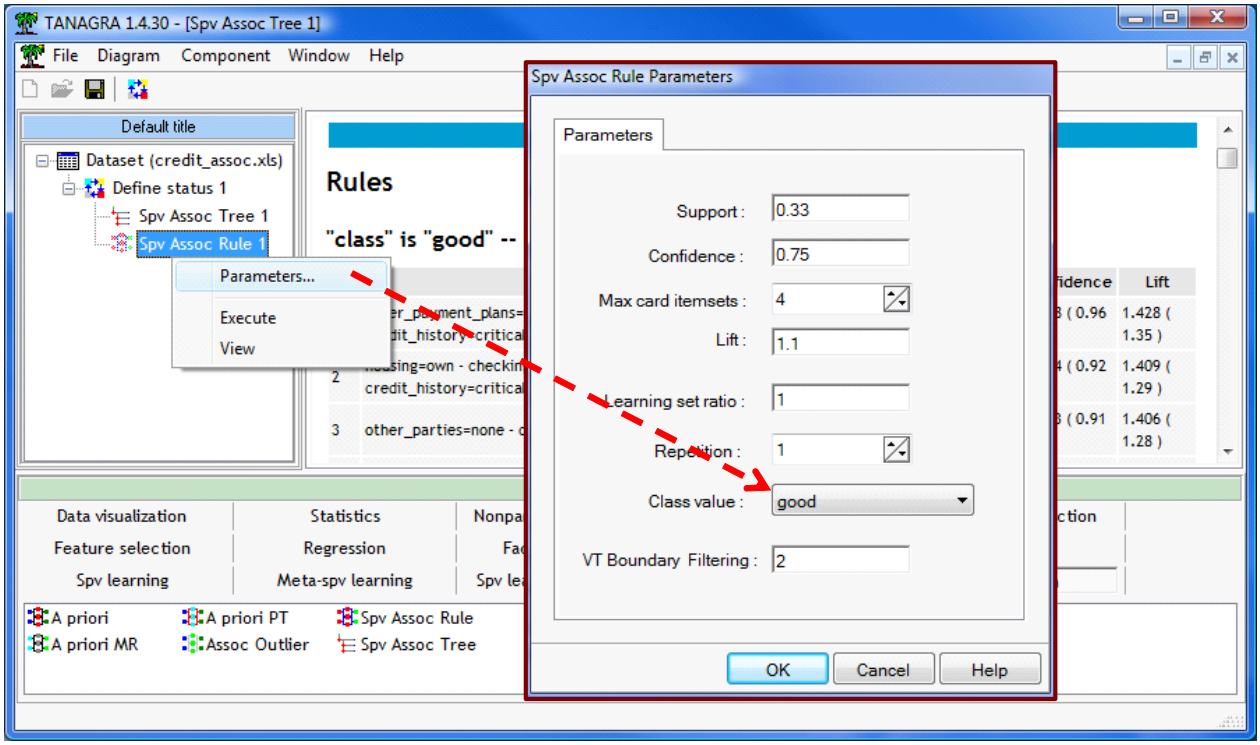
We obtain 327 rules. The most interesting according to the lift are in the upper part of the table. The values into the brackets, computed on the test sample, give an honest estimate of the rule performance.

# 5 SPV ASSOC RULE component

SPV ASSOC RULE extracts also predictive association rules. It is very similar to the previous component, but: (a) the computations are organized differently; (b) it provides more interestingness measure; (c) it can simplify the rule base.

## 5.1 Setting the parameters

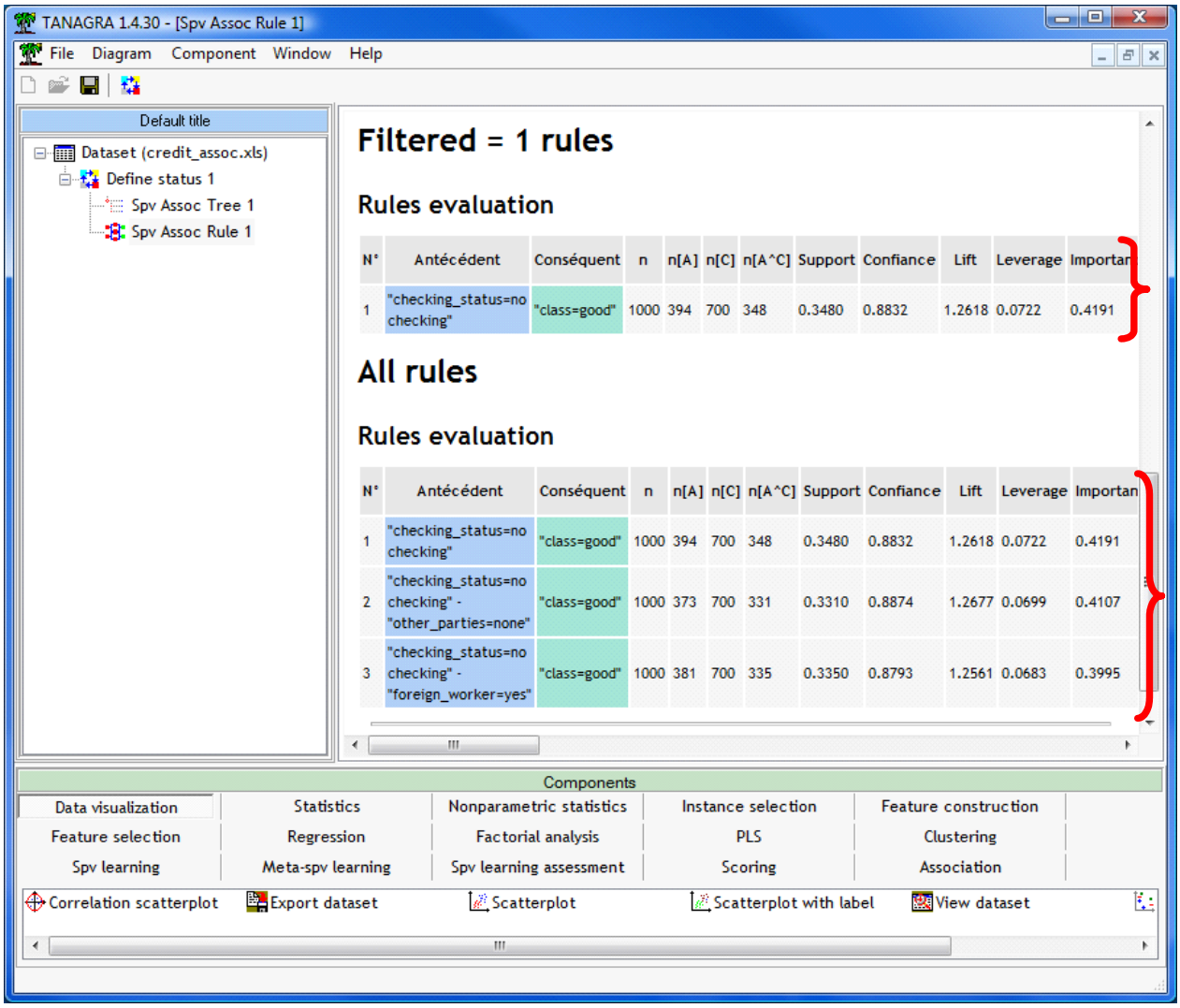
We insert SPV ASSOC RULE into the diagram. We click on the PARAMETERS menu. We specify the class value (CLASS = GOOD).




Other parameters are provided. They are in relation to the new measures described on line (http://data-mining-tutorials.blogspot.com/2009/02/interestingness-measures-for.html). There is especially the test value. These parameters are:

- REPETITION defines the number of replication during the Monte Carlo procedure;
- VT Boundary Filtering defines the minimum test value for the extracted rules. We use the **Z (HYP)** measure for the comparison.

We click on VIEW.



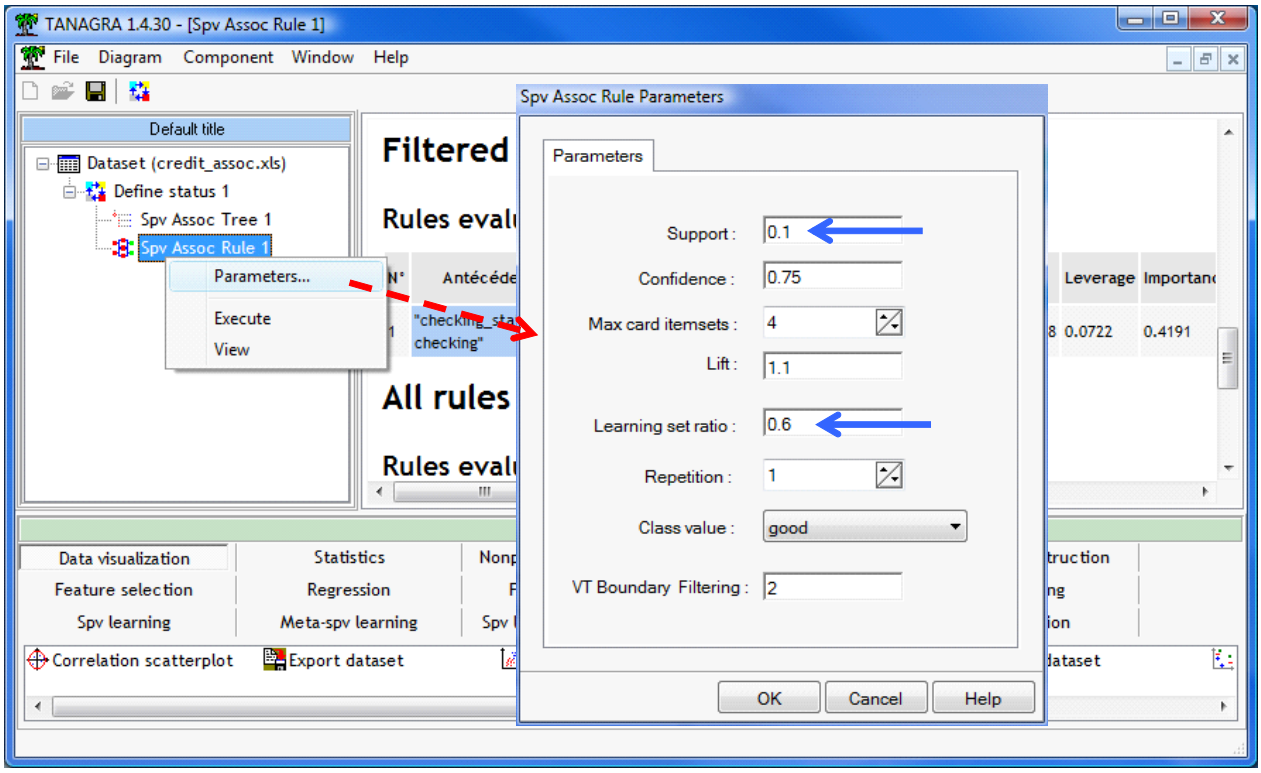

The rules are displayed in 2 separate parts: **ALL RULES** shows all the extracted rules, there are 3 here; **FILTERED RULES** shows the simplified rule base i.e. *after the elimination of redundant rules*.

Indeed, we note that the rules n°2 and n°3 do not give more information compared to the first rule (n°1). The computation is based only on a logical criterion. We assume that all the rules have the same weight.

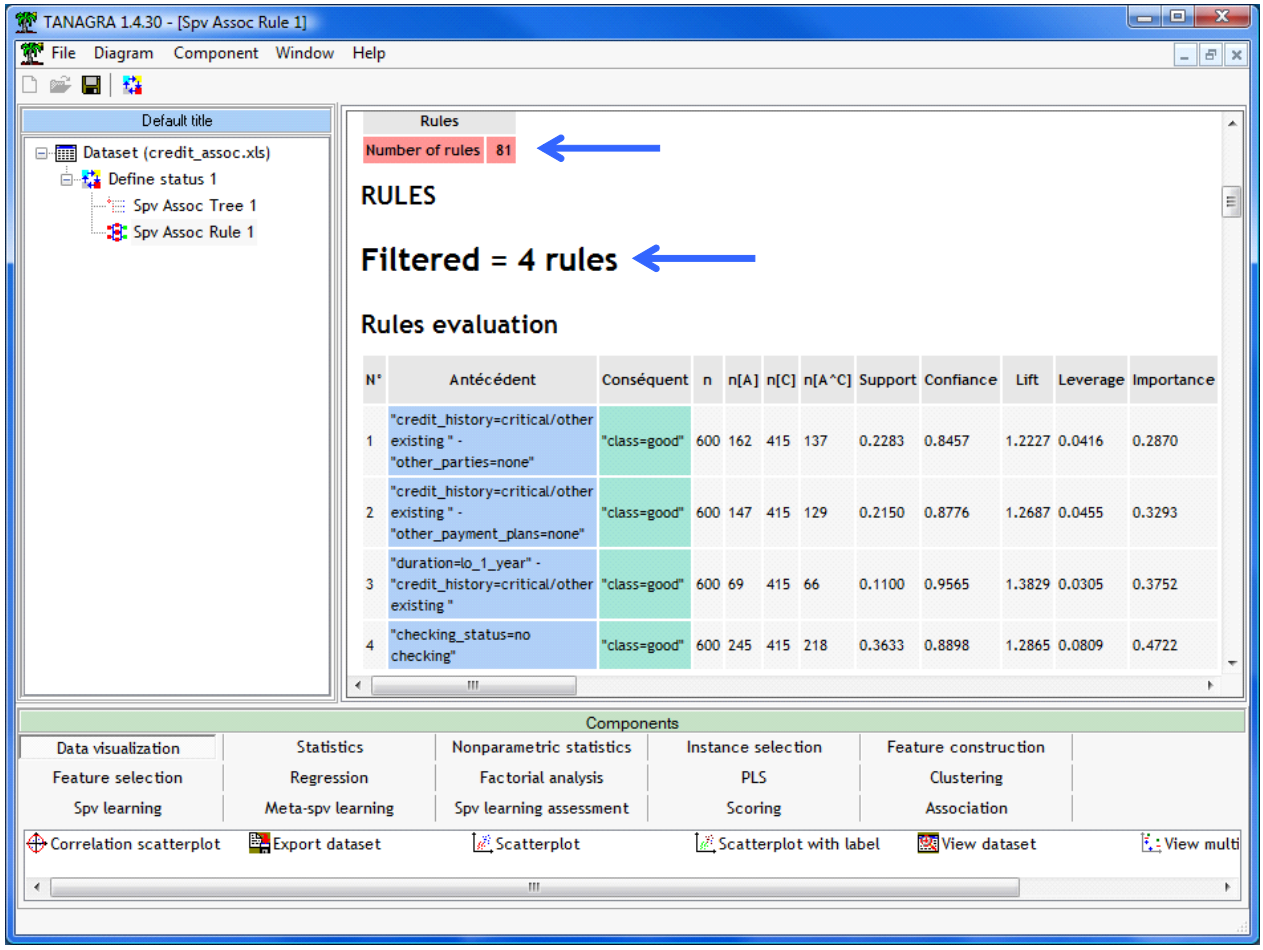
## 5.2 Extracting more rules

As above, we can generate more rules by modifying some parameters. The simplification module allows highlighting the most important information afterwards.

We click on the PARAMETERS menu. We set LEARNING SET RATIO = 0.6 and SUPPORT = 0.1




We validate and we click on the VIEW menu. The indicators computed on the test set are displayed in the second part of the table.




| N° | Antécédent | Conséquent | n | n[A] | n[C] | n[A^C] | Support | Confiance | Lift | Leverage | Importance |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | "credit_history=critical/other existing " - "other_parties=none" | "class=good" | 600 | 162 | 415 | 137 | 0.2283 | 0.8457 | 1.2227 | 0.0416 | 0.2870 |
| 2 | "credit_history=critical/other existing " - "other_payment_plans=none" | "class=good" | 600 | 147 | 415 | 129 | 0.2150 | 0.8776 | 1.2687 | 0.0455 | 0.3293 |
| 3 | "duration=lo_1_year" - "credit_history=critical/other existing " | "class=good" | 600 | 69 | 415 | 66 | 0.1100 | 0.9565 | 1.3829 | 0.0305 | 0.3752 |
| 4 | "checking_status=no checking" | "class=good" | 600 | 245 | 415 | 218 | 0.3633 | 0.8898 | 1.2865 | 0.0809 | 0.4722 |

The complete rule base contains 81 rules. We have less than SPV TREE ASSOC because VT BOUNDARY FILTERING also limits the number of rule: a rule is accepted if and only if [Z (HYP)> VT BOUNDARY FILTERING]. If we set VT BOUNDARY FILTERING 0, we obtained 327 rules, like to SPV TREE ASSOC.

After removing the redundant rules, we have only 4 rules. The interpretation of the results is simplified.

# 6 Conclusion

In this tutorial, we presented two components of Tanagra for the extraction of predictive association rules. They differ in the strategy used to overcome the problem of the abundance of rules inherent to the extraction algorithm: SPV TREE ASSOC offers the possibility to organize the rules according to a numerical criterion chosen by the user; SPV RULE ASSOC uses a simplification procedure.