

## 1 Introduction

### Extraction des règles d'association prédictives dans Tanagra.

Les algorithmes d'extraction des règles d'association ont été initialement mis au point pour découvrir des liens logiques entre des variables ayant le même statut. Les règles d'association prédictives en revanche cherchent à produire les combinaisons d'items qui caractérisent au mieux une variable qui joue un rôle à part, on cherche à prédire ses valeurs.

Fondamentalement, l'algorithme est peu modifié. L'exploration est simplement restreinte aux itemsets qui comportent la variable à prédire. Le temps de calcul est d'autant réduit. Deux composants de Tanagra sont dédiés à cette tâche, il s'agit de SPV ASSOC RULE et SPV ASSOC TREE. Ils sont accessibles dans l'onglet ASSOCIATION.

Par rapport aux approches classiques, les composants de Tanagra introduisent une spécificité supplémentaire : nous avons la possibilité de préciser la classe (couple « variable à prédire = valeur ») que l'on souhaite prédire. L'intérêt est de pouvoir ainsi paramétrer finement l'algorithme de recherche, en relation directe avec les caractéristiques des données. Cela s'avère décisif par exemple lorsque les prévalences des modalités de la variable à prédire sont très différentes.

Nous avons déjà présentés le composant SPV ASSOC TREE par ailleurs (<http://tutoriels-data-mining.blogspot.com/2008/04/rgles-dassociation-supervises.html>). Mais c'était dans le contexte de la caractérisation multivariée de groupes d'individus. Nous l'opposons alors au composant GROUP CHARACTERIZATION. Dans ce didacticiel, nous comparerons le comportement des composants SPV ASSOC TREE et SPV ASSOC RULE sur un problème de prédiction. Nous mettrons en avant leurs points communs, les problèmes qu'ils savent traiter ; et leurs différences, SPV ASSOC RULE, en plus de proposer des mesures d'intérêt des règles originales, a la capacité de simplifier la base de règles.

## 2 Données

Nous utilisons une version modifiée du fichier GERMAN CREDIT<sup>1</sup>. Il décrit les caractéristiques de demandeurs de crédit. Nous avons discrétisé les variables quantitatives. Le fichier est accessible en ligne ([http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/credit\\_assoc.xls](http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/credit_assoc.xls)).

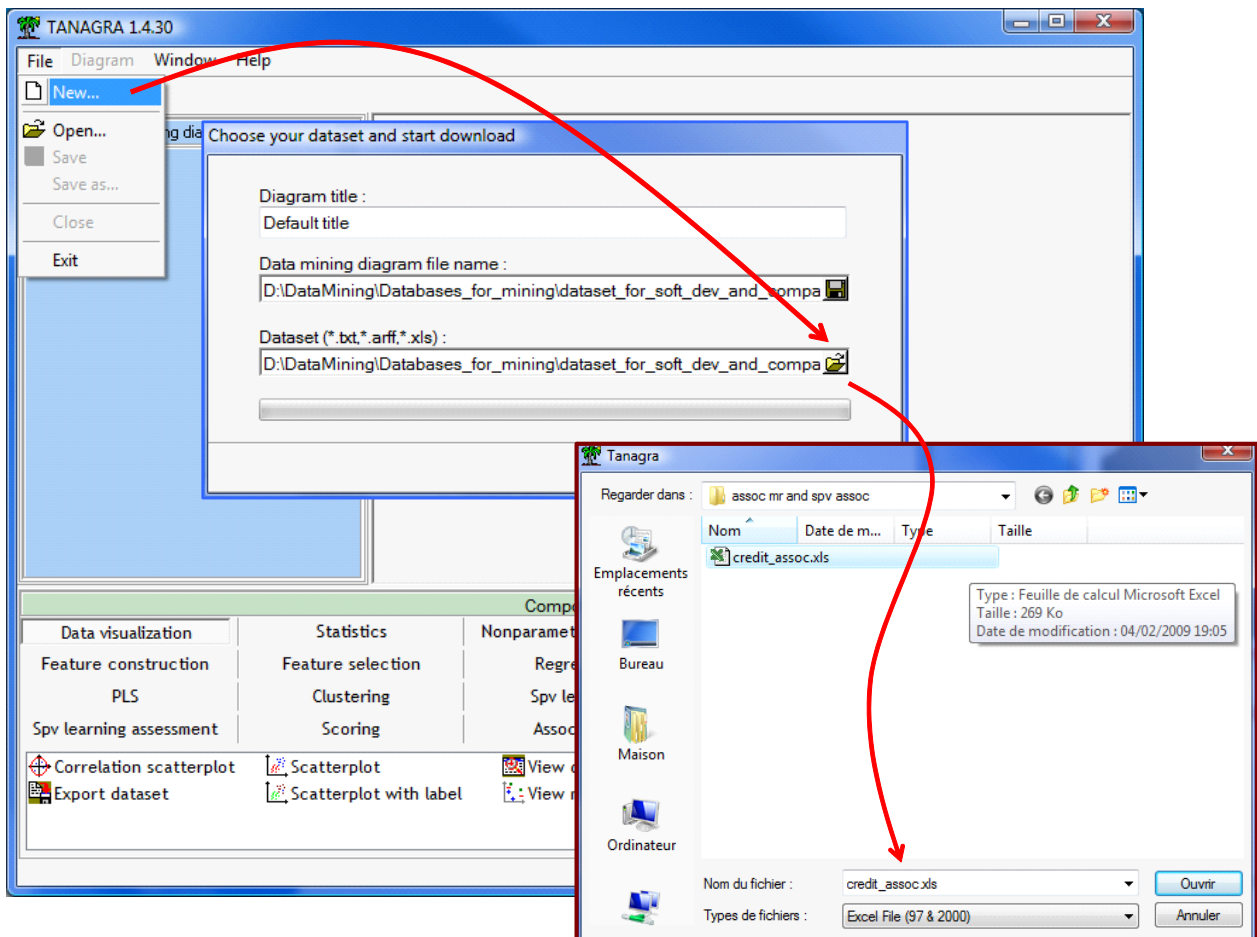
La variable CLASS joue un rôle particulier dans ce didacticiel. Nous cherchons à mettre à jour les caractéristiques des « bons » clients (CLASS = GOOD). Nous avons donc un double paramétrage à faire avant de pouvoir exécuter les calculs : indiquer que CLASS est la variable TARGET ; parmi les valeurs de CLASS, choisir la modalité GOOD.

---

<sup>1</sup> [http://archive.ics.uci.edu/ml/datasets/Statlog+\(German+Credit+Data\)](http://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data))

### 3 Création d'un diagramme

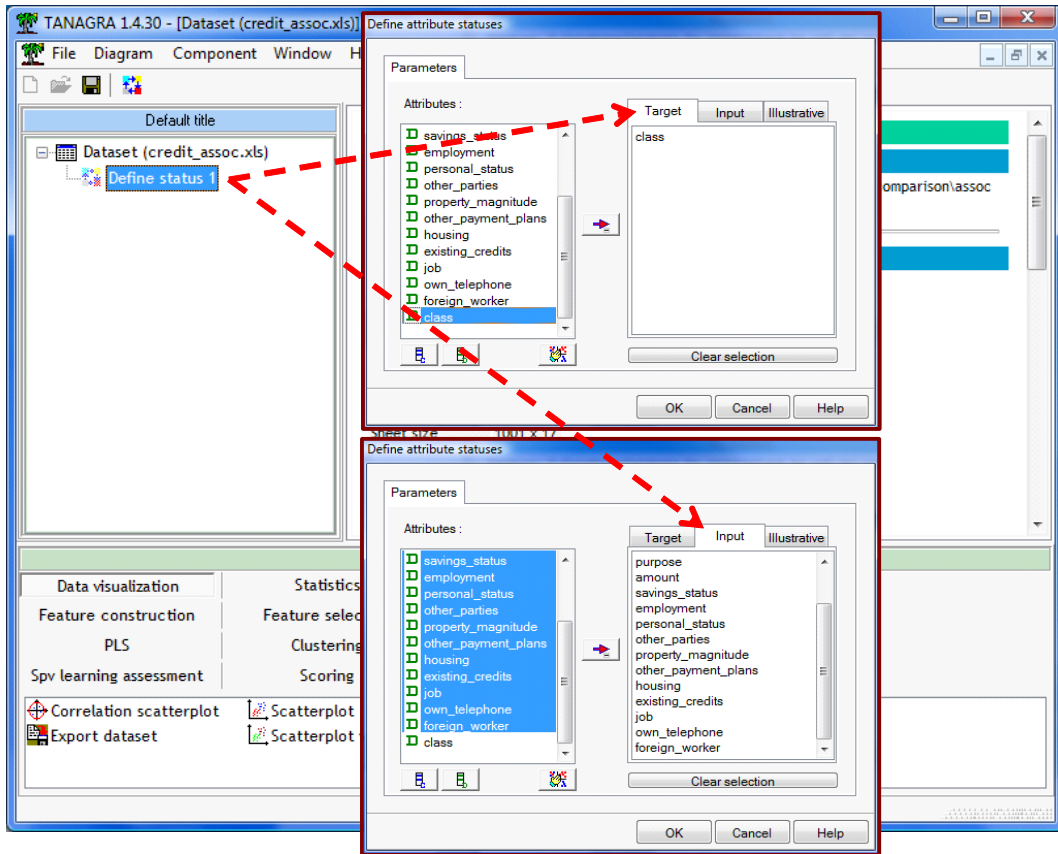
**Importer les données.** Première étape toujours dans tout projet de Data Mining, nous devons définir un nouveau projet et importer les données. Après avoir lancé Tanagra, nous actionnons le menu FILE / NEW. Tanagra sait lire directement les fichiers au format Excel (XLS)<sup>2</sup>. Nous sélectionnons le fichier CREDIT\_ASSOC.XLS.



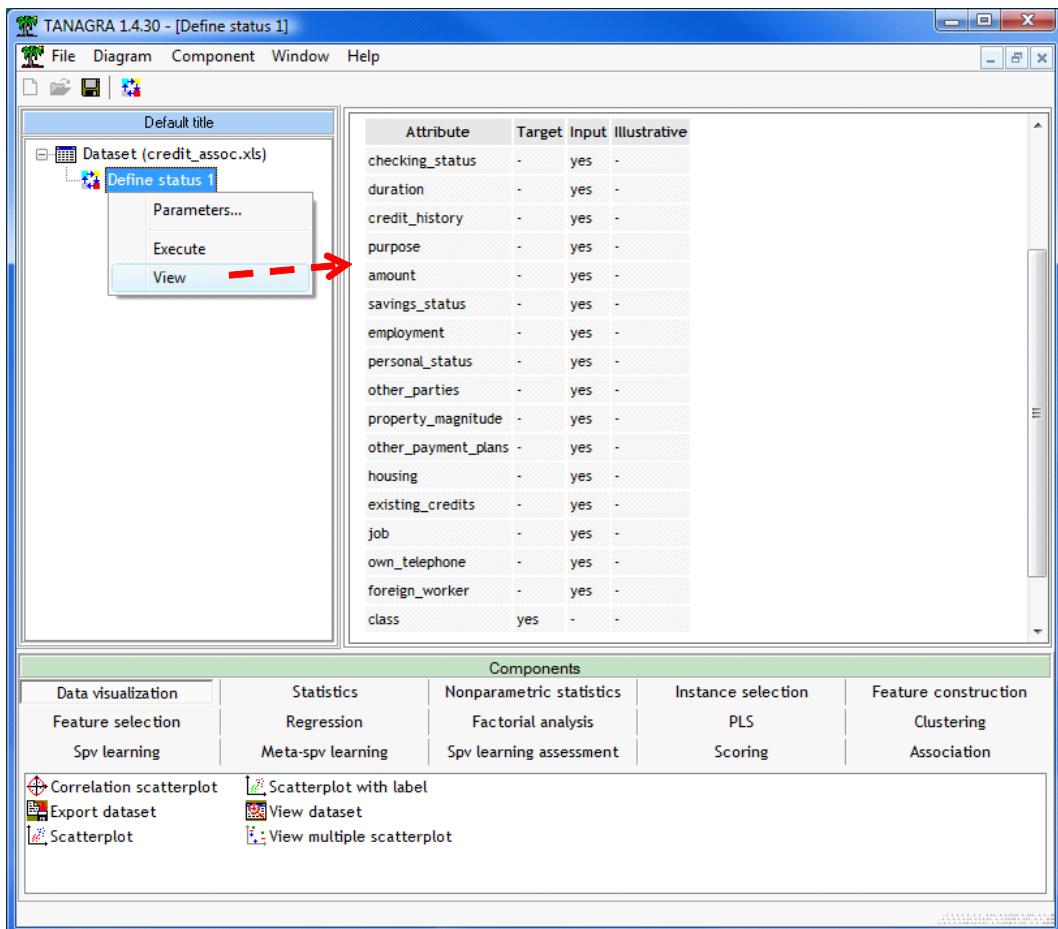
Tanagra nous indique que 17 variables et 1000 observations ont été chargées.

**Définir le rôle des attributs.** Nous insérons le composant DEFINE STATUS via le raccourci dans la barre d'outils pour indiquer le rôle des variables : nous plaçons CLASS en TARGET, toutes les autres en INPUT.

<sup>2</sup> Sur les différentes manières d'importer des données au format XLS (Excel), voir <http://tutoriels-data-mining.blogspot.com/2008/03/importation-fichier-xls-excel-mode.html> et <http://tutoriels-data-mining.blogspot.com/2008/03/importation-fichier-xls-excel-macro.html>



Nous validons et nous cliquons sur VIEW. Tanagra affiche un récapitulatif.

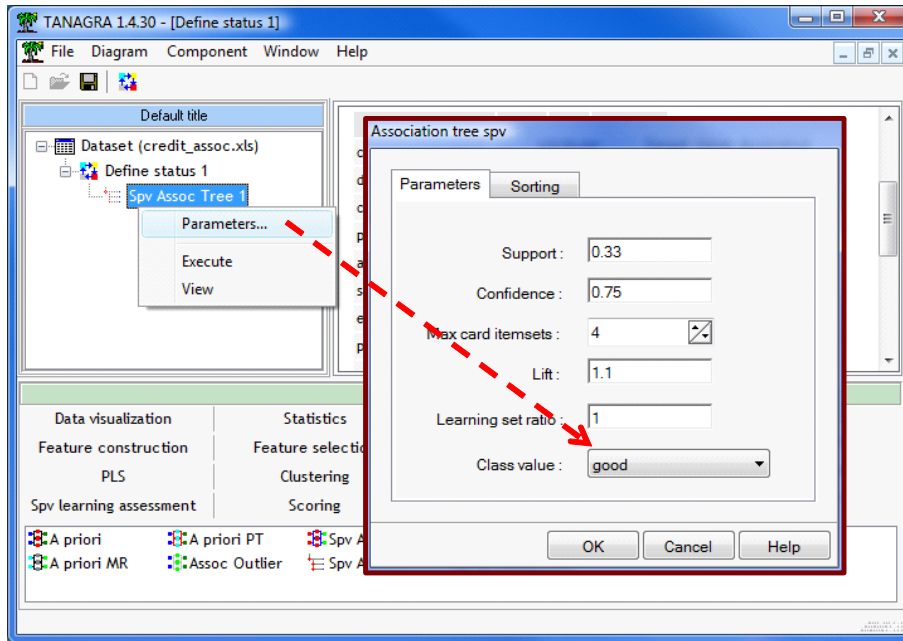


## 4 Le composant SPV ASSOC TREE

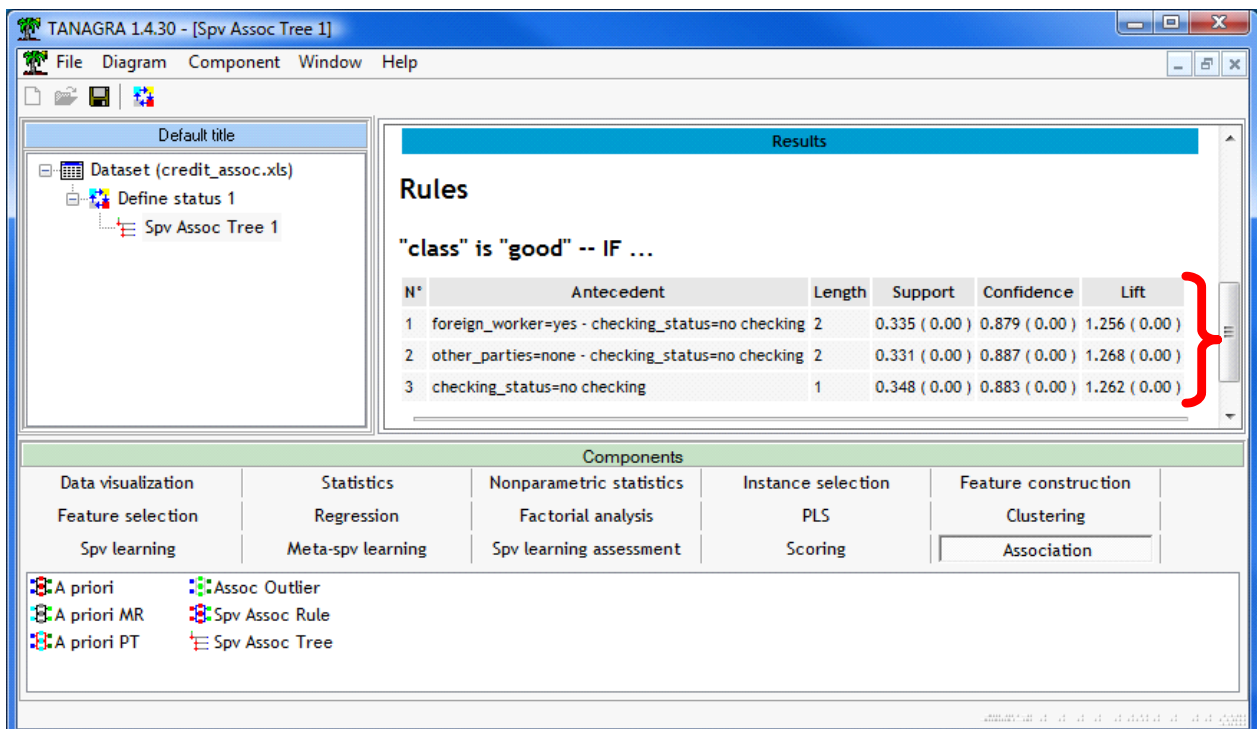
Le composant SPV ASSOC TREE extrait des règles d'association. L'implémentation utilise en interne un arbre de recherche, d'où son nom, mais au final nous obtenons bien des règles.

### 4.1 Choix de la modalité à prédire

Nous l'insérons dans le diagramme. Nous actionnons le menu PARAMETERS pour introduire un premier paramétrage obligatoire : nous devons indiquer au logiciel la modalité de la variable à prédire que l'on souhaite expliquer. Dans l'option CLASS VALUE, nous désignons la modalité GOOD.



Nous validons et nous lançons les calculs.

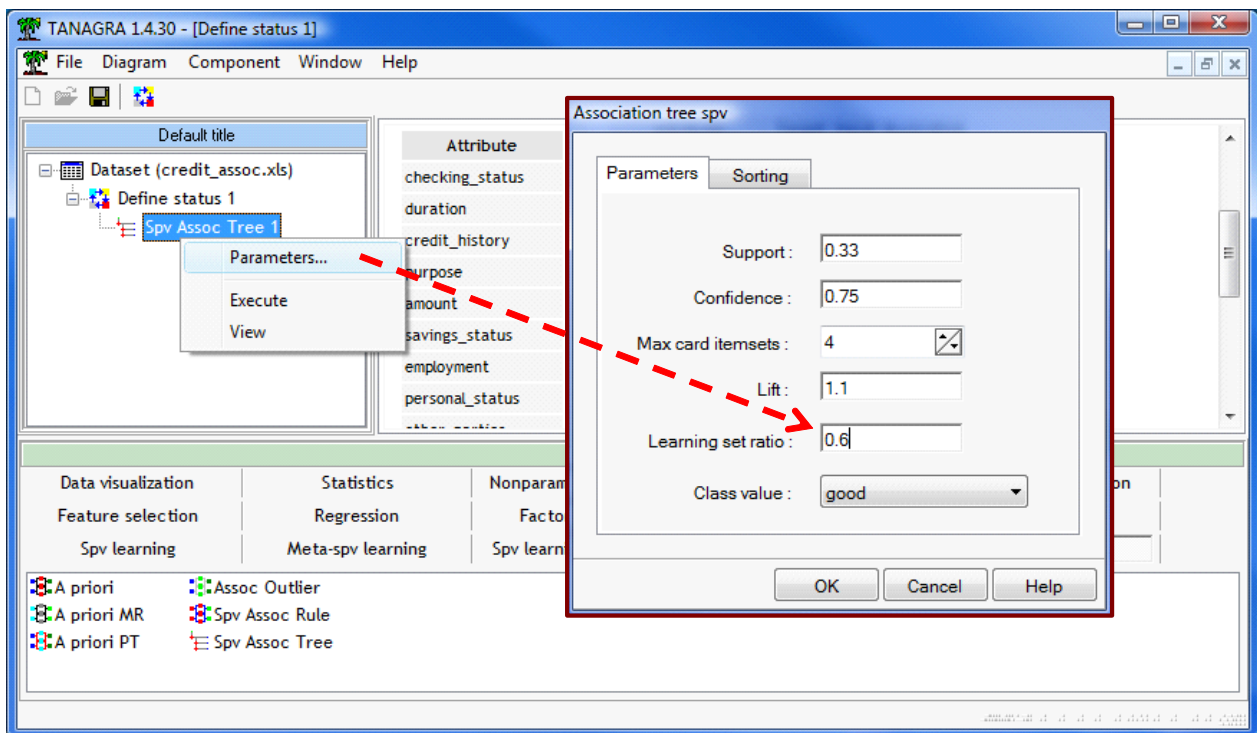


Le composant produit 3 règles. Elles sont énumérées dans la partie basse de la fenêtre d'affichage. Le support, la confiance et le lift sont fournis. Lorsque les chiffres sont entre parenthèses, cela veut dire que l'indicateur a été calculé sur un échantillon test, n'ayant pas participé à l'élaboration des règles. Ici, nous avons systématiquement la valeur zéro. Toutes les observations sont utilisées pour l'élaboration des règles.

#### 4.2 Partition « apprentissage – test » des données

Le danger lors de l'induction est de produire des solutions trop spécifiques aux données traitées, qui ne correspondent à aucune réalité dans la population. Pour éviter cet écueil, il est généralement conseillé de partitionner les observations en deux parties : la première, dite échantillon d'apprentissage, sert à élaborer les règles de prédiction ; la seconde, dite échantillon de test, sert uniquement à leur évaluation. Les indicateurs calculés dans ce second contexte sont nettement plus crédibles, traduisant les performances réelles de la règle. A mon sens, il faut surtout s'inquiéter lorsque le même indicateur calculé sur les deux échantillons prend des valeurs anormalement différentes.

Pour subdiviser les données, nous revenons sur la boîte de paramétrage via le menu PARAMETERS. Nous modifions le paramètre LEARNING SET RATIO, nous le fixons à 0.6 (*attention, le point décimal dépend de la configuration de votre système*) c.-à-d. nous utilisons 60% des données pour la construction des règles, 40% pour leur évaluation.



Nous validons puis nous cliquons sur VIEW.

N°	Antecedent	Length	Support	Confidence	Lift
1	foreign_worker=yes - other_parties=none - checking_status=no checking	3	0.343 ( 0.28 )	0.900 ( 0.86 )	1.301 ( 1.20 )
2	foreign_worker=yes - checking_status=no checking	2	0.353 ( 0.31 )	0.887 ( 0.87 )	1.282 ( 1.22 )
3	other_parties=none - checking_status=no checking	2	0.352 ( 0.30 )	0.902 ( 0.86 )	1.304 ( 1.21 )
4	other_payment_plans=none - housing=own	2	0.428 ( 0.47 )	0.763 ( 0.78 )	1.103 ( 1.09 )
5	checking_status=no checking	1	0.363 ( 0.33 )	0.890 ( 0.87 )	1.286 ( 1.22 )

Première information importante, nous avons davantage de règles, sans que nous ayons modifié les autres paramètres. Cela veut dire simplement que certaines règles (la n°1 pour le critère de support ; la n°4 pour le critère de confiance) étaient toutes proches des seuils de coupures, elles avaient été arbitrairement éliminées précédemment.

Deuxième information importante, nous avons une meilleure visibilité sur la stabilité de la règle en comparant le même indicateur calculé sur l'échantillon d'apprentissage et de test. Nous noterons entre autres que les chiffres en test sont moins optimistes.

### 4.3 Produire plus de règles et les organiser

Comme nous avons pu le constater, les paramètres de filtrage jouent un rôle important lors de l'extraction. SPV ASSOC TREE propose les éléments suivants :

- SUPPORT définit le support minimum des règles extraites ;
- CONFIANCE définit la confiance minimum des règles extraites ;
- MAX CARD ITEMSETS restreint la longueur maximale de la règle ;
- LIFT indique le LIFT minimum des règles.

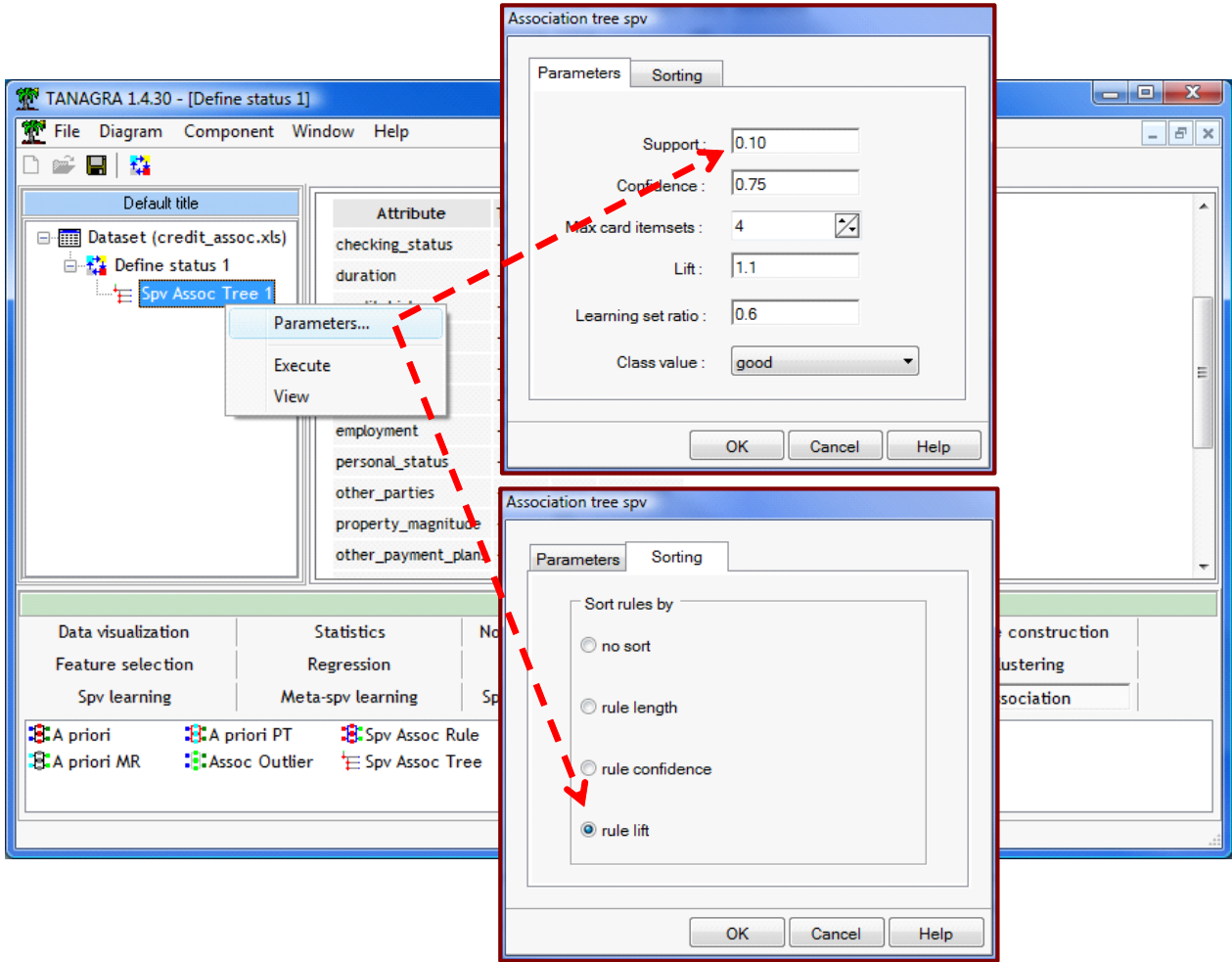
Les trois premiers paramètres restreignent l'espace de recherche, ils pèsent sur le temps de calcul. Le dernier agit a posteriori, uniquement pour limiter le nombre de règles à afficher.

Un fois les règles produites, il faut pouvoir les organiser de manière à ce que les plus intéressantes apparaissent en premier. Tanagra sait classer les règles selon un des critères numériques ci-dessus.

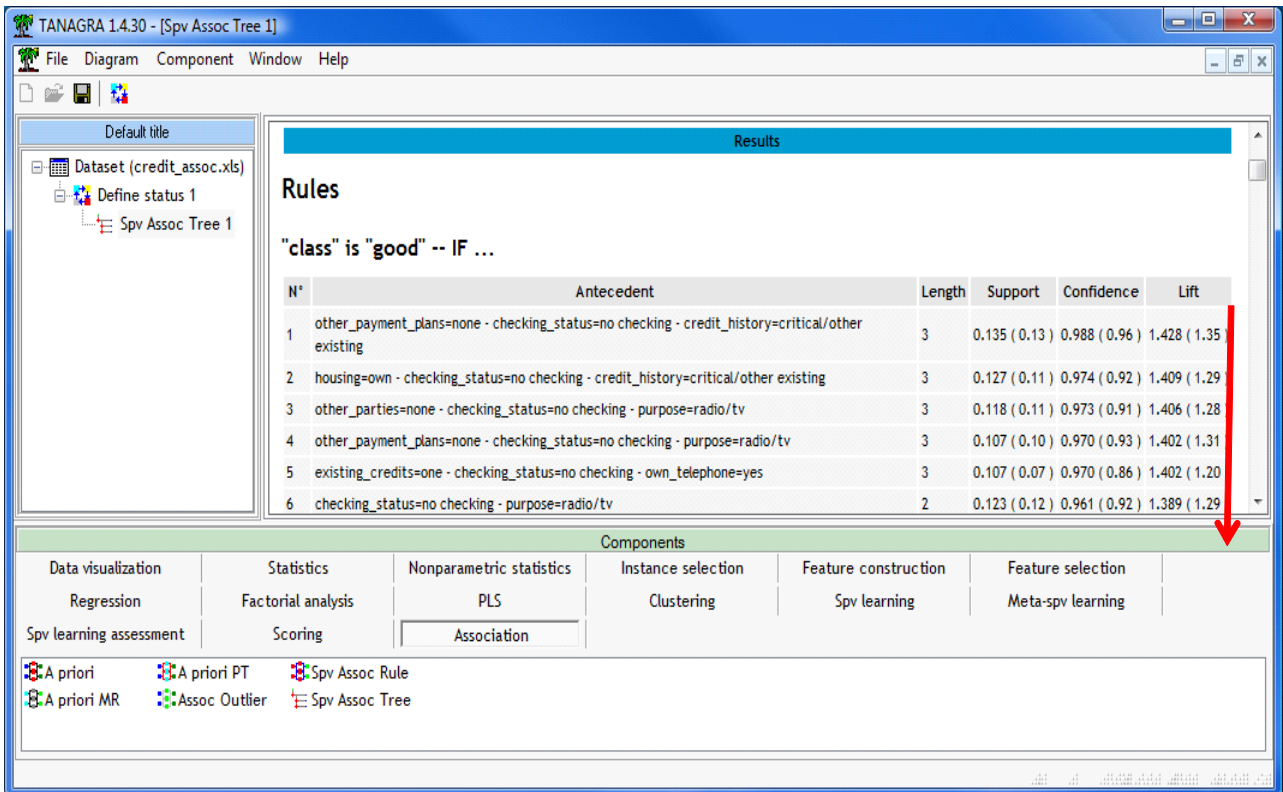
Dans ce qui suit, nous allons diminuer le support minimum pour obtenir plus de règles (SUPPORT = 0.10). Puis nous les classerons selon un lift décroissant.

Nous cliquons sur le menu PARAMETERS.





Nous cliquons sur VIEW pour obtenir les résultats.



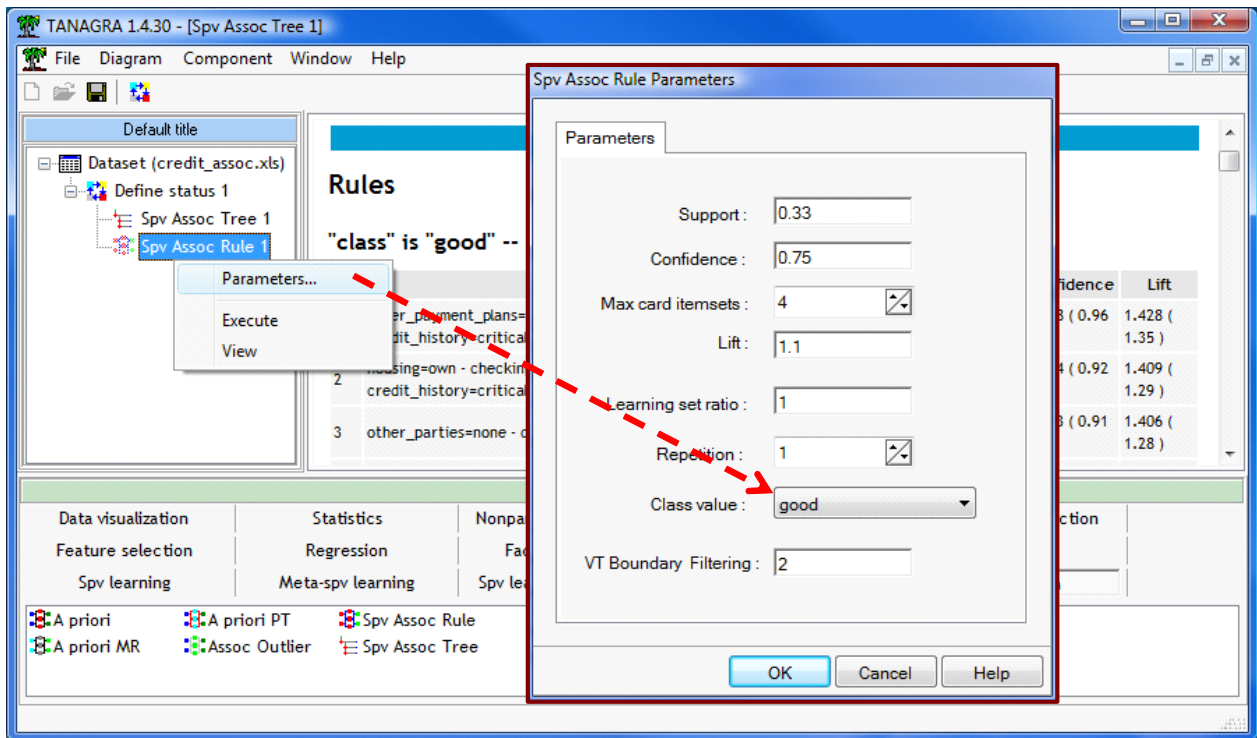
327 règles sont maintenant produites. Les plus intéressantes au sens du LIFT apparaissent en premier. On remarquera encore la différence entre les valeurs des indicateurs selon que nous les calculons sur l'échantillon d'apprentissage ou de test. C'est le chiffre entre parenthèses (échantillon test) qui est réellement crédible.

## 5 Le composant SPV ASSOC RULE

Le composant SPV ASSOC RULE produit aussi des règles prédictives. Il répond au même cahier de charges que le SPV ASSOC TREE, à la différence que : (a) l'algorithme interne est un peu différent ; (b) il propose une vaste panoplie de mesures ; (c) il intègre la possibilité de réduire la base de règle par simplification logique. Voyons comment tout cela fonctionne.

### 5.1 Paramétrage et lecture des résultats

Nous insérons le composant dans le diagramme. Nous actionnons le menu PARAMETERS. Nous indiquons la modalité de la variable à prédire que nous souhaitons caractériser c.-à-d. CLASS VALUE = GOOD.



D'autres paramètres sont proposés. Ils sont en relation avec les nouvelles mesures d'évaluation de règles intégrées dans l'outil. Il y a en particulier la valeur test que nous décrivons longuement par ailleurs (<http://tutoriels-data-mining.blogspot.com/2009/02/mesures-dinteret-des-regles-dans-priori.html>).

Citons rapidement ces paramètres :

- REPETITION définit le nombre de réplifications lors du calcul à l'aide de la procédure de Monte Carlo ;
- VT Boundary Filtering définit la valeur test seuil qui permet d'accepter une règle. Il se réfère à l'indicateur **Z (HYP)**. Il agit a posteriori pour limiter le nombre de règles à afficher.

Nous cliquons sur VIEW.



**Filtered = 1 rules**

Rules evaluation

N°	Antécédent	Conséquent	n	n[A]	n[C]	n[A^C]	Support	Confiance	Lift	Leverage	Importan
1	"checking_status=no checking"	"class=good"	1000	394	700	348	0.3480	0.8832	1.2618	0.0722	0.4191

**All rules**

Rules evaluation

N°	Antécédent	Conséquent	n	n[A]	n[C]	n[A^C]	Support	Confiance	Lift	Leverage	Importan
1	"checking_status=no checking"	"class=good"	1000	394	700	348	0.3480	0.8832	1.2618	0.0722	0.4191
2	"checking_status=no checking" - "other_parties=none"	"class=good"	1000	373	700	331	0.3310	0.8874	1.2677	0.0699	0.4107
3	"checking_status=no checking" - "foreign_worker=yes"	"class=good"	1000	381	700	335	0.3350	0.8793	1.2561	0.0683	0.3995

**Components**

Data visualization	Statistics	Nonparametric statistics	Instance selection	Feature construction
Feature selection	Regression	Factorial analysis	PLS	Clustering
Spv learning	Meta-spv learning	Spv learning assessment	Scoring	Association

Correlation scatterplot | Export dataset | Scatterplot | Scatterplot with label | View dataset

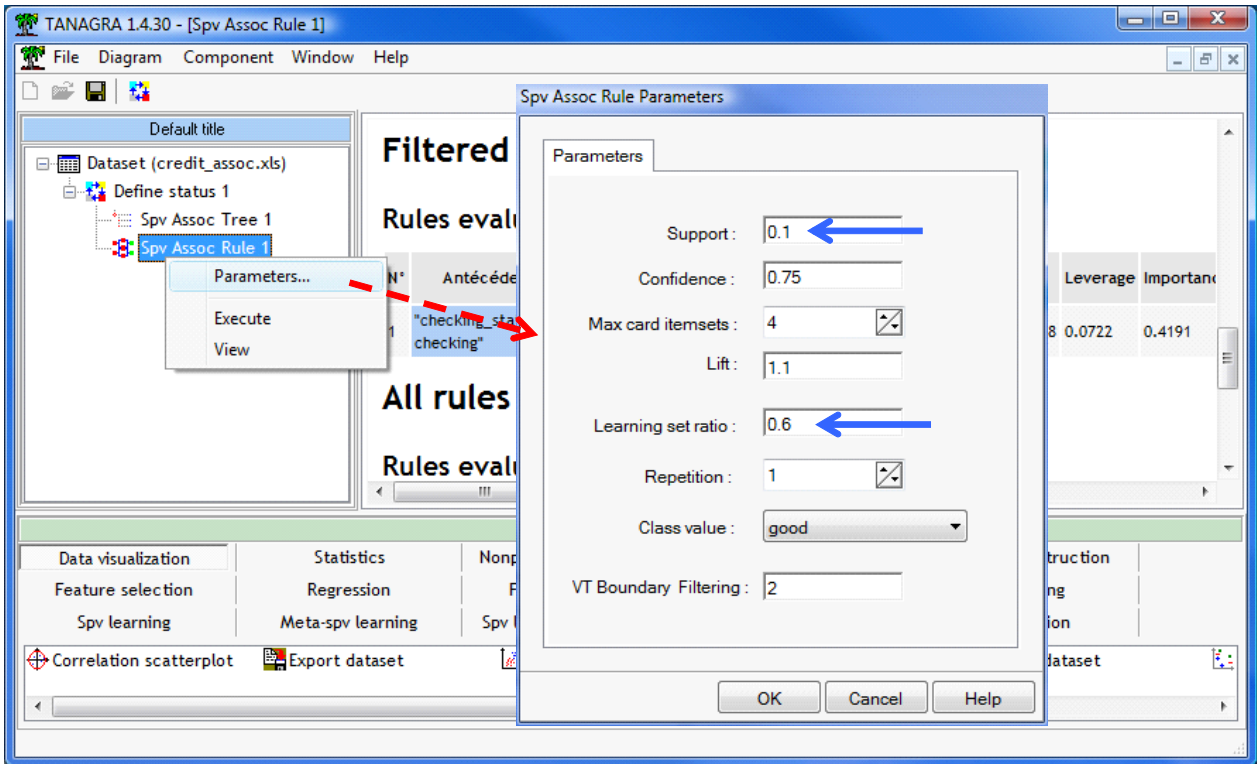
Les règles sont affichés dans 2 blocs séparés : **ALL RULES** indique toutes les règles extraites, nous en obtenons 3, tout comme avec le composant SPV ASSOC TREE ; **FILTERED RULES** indique la base de règles après simplification c.-à-d. **élimination des règles redondantes**, dans notre cas il ne reste plus qu'une règle.

En effet, nous nous rendons compte que les règles n°2 et n°3 n'apportent pas d'informations supplémentaires, en termes logiques, par rapport à la première règle n°1. Il suffit que « CHECKING STATUS = NO » pour que la classe soit GOOD. L'adjonction de conditions (items) supplémentaires ne modifie pas la conclusion. Notons un élément important, **le module de simplification est uniquement basé sur des critères logiques. On considère que les règles ont le même poids dès lors qu'elles ont passé le filtrage numérique.**

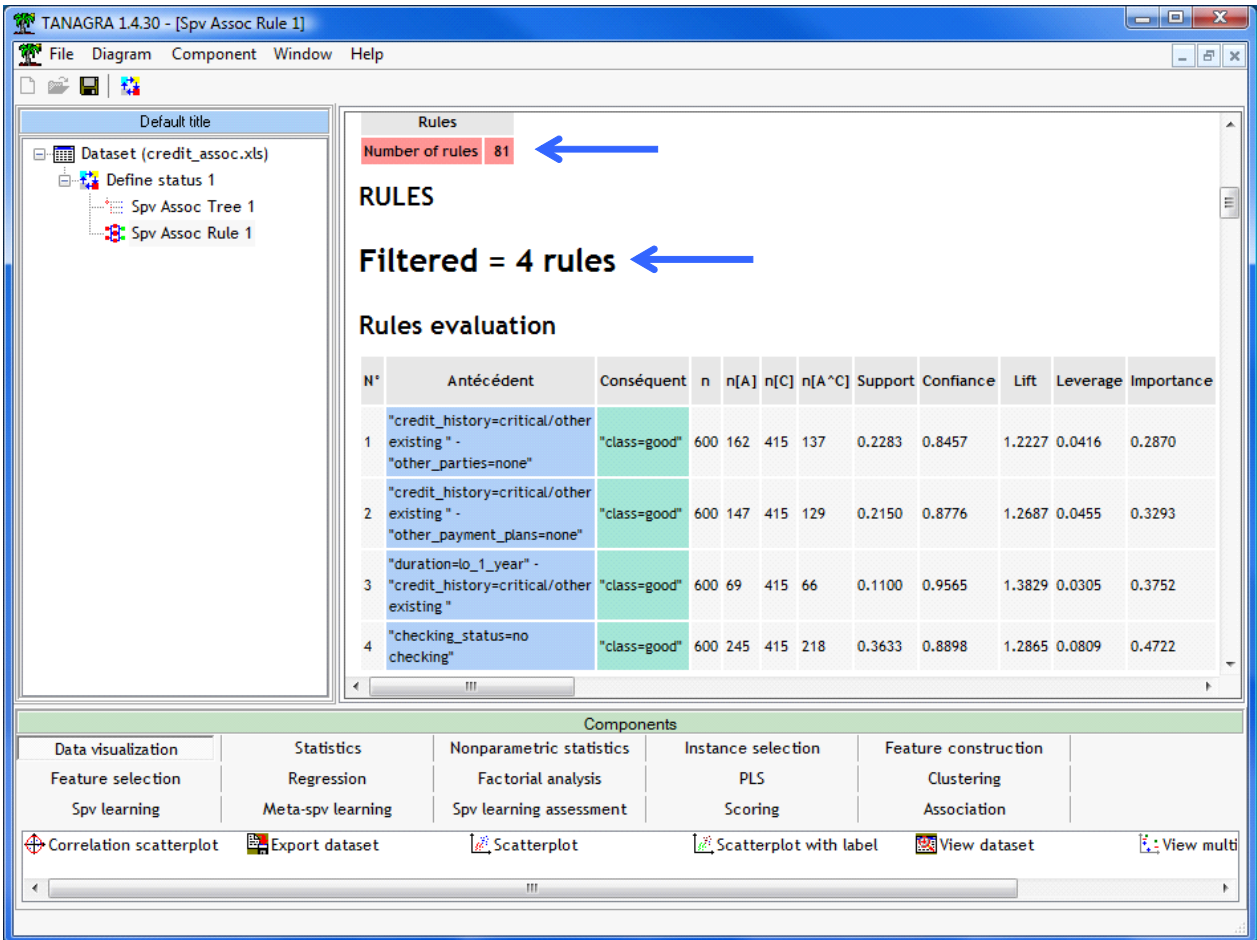
## 5.2 Produire plus de règles

Comme précédemment, nous pouvons réduire nos exigences numériques pour obtenir plus de règles. L'intérêt de SPV ASSOC RULE est la possibilité de réduire après coup la base de règles en éliminant les règles redondantes à l'aide du module de simplification.

Nous actionnons le menu PARAMETERS. Nous partitionnons les données en 60% pour l'apprentissage et 40% pour le test (LEARNING SET RATIO = 0.6 – *attention toujours au point décimal*), nous réduisons le support minimal à 10% (SUPPORT = 0.1).



Nous validons et cliquons sur VIEW. Les indicateurs calculés sur l'échantillon test sont affichés dans la deuxième partie du tableau, tout à droite.



La base complète comporte 81 règles. Nous en avons moins qu'avec SPV ASSOC TREE car VT BOUNDARY FILTERING a également agit pour en limiter le nombre : une règle est acceptée si et seulement si  $[Z (HYP) > VT BOUNDARY FILTERING]$ . Si nous avons fixé VT BOUNDARY FILTERING à 0, nous aurions obtenu 327 règles, exactement comme avec SPV ASSOC TREE.

Après simplification des règles redondantes, nous n'avons plus que 4 règles. L'interprétation des résultats est grandement simplifiée.

## 6 Conclusion

Dans ce didacticiel, nous avons présenté deux composant originaux de Tanagra pour l'élaboration de règles d'association prédictives. Elles se distinguent par la stratégie utilisée pour dépasser l'écueil de la profusion des règles inhérente à l'algorithme d'extraction : SPV ASSOC TREE offre la possibilité d'organiser les règles selon un critère numérique choisi par l'utilisateur ; SPV ASSOC RULE introduit un module de simplification logique de la base de règles.