

# 1 Objectif

## Extraction des itemsets à l'aide du composant FREQUENT ITEMSETS.

La recherche des régularités dans les bases de données est l'idée principale du data mining. Ces régularités s'expriment sous différentes formes. Dans l'analyse du panier d'achats de consommateurs, l'extraction des itemsets consiste à mettre en exergue les cooccurrences entre les produits achetés c.-à-d. déterminer les produits (les items) qui sont « souvent » achetés simultanément. On parle alors d'itemsets fréquents. Par exemple, en analysant les tickets de caisse d'un supermarché, on pourrait produire des itemsets (un ensemble d'items) du type « le pain et le lait sont présents dans 10% des caddies ».

La recherche des itemsets fréquents est souvent présentée comme un préalable à l'extraction des règles d'association où l'on essaie, en sus, de mettre en évidence des relations de causalité. En reprenant notre exemple ci-dessus, une règle possible serait « ceux qui ont acheté du pain et du lait ont aussi acheté du beurre ». L'objectif est d'exploiter ce type de connaissance pour mieux agencer les rayons (mettre le beurre pas trop loin du pain et du lait) ou pour faire une offre promotionnelle ciblée (faire une promotion sur le pain et le lait dans le but d'augmenter les ventes de beurre).

En réalité, les itemsets fréquents sont en elles-mêmes porteuses d'informations. Savoir quels sont les produits achetés ensembles permet d'identifier les liens existants entre eux et, par là, de réaliser une typologie des achats ou de dégager des comportements types chez les consommateurs. Dans le cas du pain et du lait, il s'agit certainement d'achats relatifs au petit déjeuner. Si les consommateurs se mettent à acheter conjointement de la viande et du charbon, nous sommes en été, c'est la saison des barbecues...

Dans ce tutoriel, nous décrivons la mise en œuvre du composant FREQUENT ITEMSETS de Tanagra, basé sur la bibliothèque « apriori.exe » de Borgelt<sup>1</sup>. Nous utilisons un petit jeu de données pour que tout un chacun puisse reconstituer manuellement les résultats produits par le logiciel. Mais, dans un premier temps, essayons d'explicitier les différentes notions liées à l'extraction des itemsets.

## 2 Extraction des itemsets

Notre fichier comporte 10 observations (transactions) et 4 items<sup>2</sup>.

S1	S2	S3	S4
1	0	1	0
0	1	0	0
0	0	0	1
0	1	1	1
0	1	1	0
0	1	1	0
1	1	1	1
1	0	1	0
1	1	1	0
1	1	1	0

<sup>1</sup> <http://www.borgelt.net/apriori.html> (Version 5.57)

<sup>2</sup> <http://www.dataminingarticles.com/closed-maximal-itemsets.html>

En ligne, nous avons des clients d'une compagnie d'assurance ; en colonne, des contrats (produits) associés à divers risques. Par exemple, le client n°1 a contracté les assurances S1 et S3, etc.

L'objectif est d'identifier les produits qui sont placés de concert.

**Item.** Un item correspond à un produit. Nous avons 4 items (S1, S2, S3 et S4) dans notre fichier.

**Support.** Le support d'un item est égal au nombre de transactions dans lesquelles il apparaît. Par exemple, le support de {S1} est égal à 5. Le support peut être exprimé également en termes relatifs. Dans ce cas, nous divisons le support (absolu) par le nombre total de transactions. Pour S1, nous avons  $SUP(\{S1\}) = 5/10 = 20\%$ .

**Itemset.** Un itemset est un ensemble d'items (ex. {S1,S2} est un itemset de cardinal  $CARD(\{S1, S2\}) = 2$ ). Le support d'un itemset comptabilise le nombre de transactions dans lesquelles les items apparaissent simultanément (ex.  $SUP(\{S1,S2\}) = 3/10 = 0.3$ ). Un itemset peut être composé d'un singleton (ex. {S1} avec  $SUP(\{S1\}) = 5/10$ ).

**Itemset fréquent.** Un itemset est dit fréquent si son support est supérieur à un seuil défini à l'avance, paramètre de l'algorithme de recherche. Dans notre exemple, en fixant le support minimum à 2 (ou 20% en relatif), nous observons dans le schéma suivant les itemsets fréquents (toutes les combinaisons non-grisées).

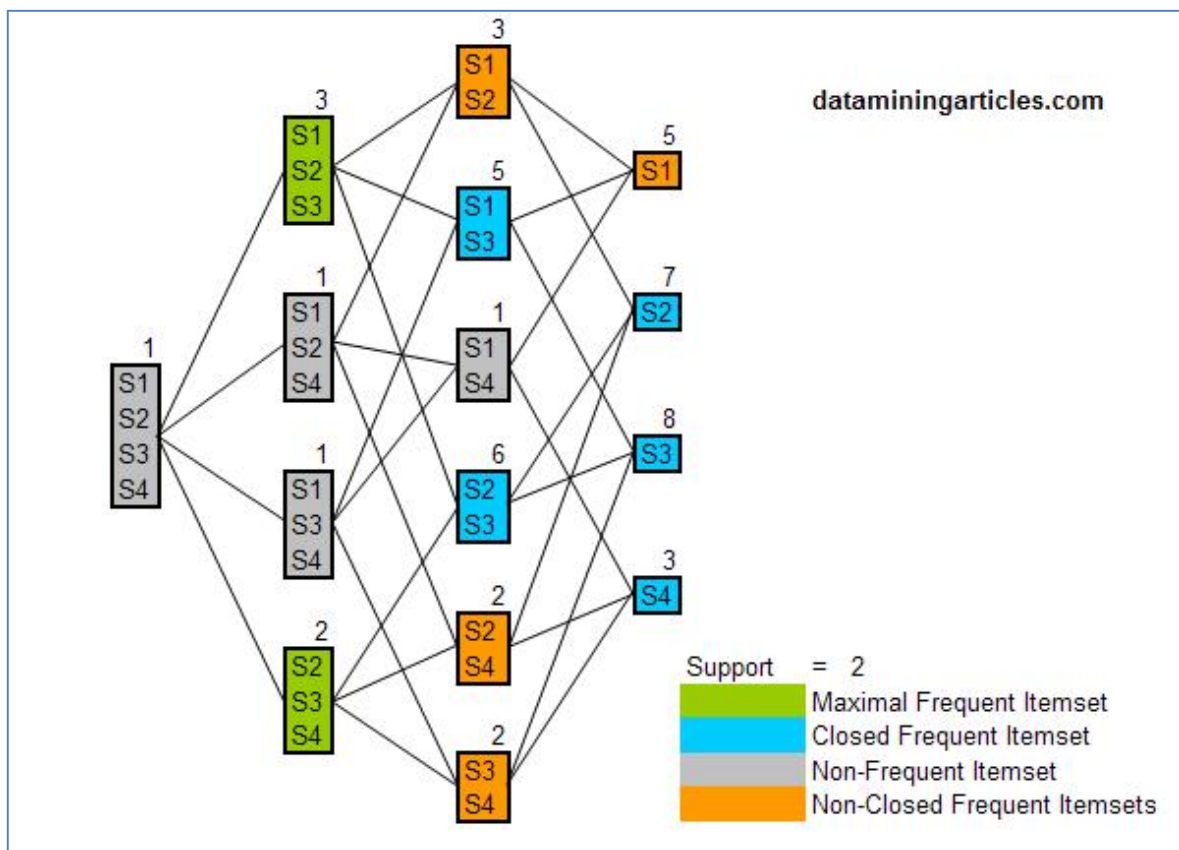


Figure 1 - Extraction des différents itemsets

**Superset.** Un superset est un itemset défini par rapport à un autre itemset. Prenons un exemple pour clarifier les idées : {S1, S2, S3} est un superset de {S1, S2}. Ainsi, de manière générale, B est un superset de A, si  $CARD(A) < CARD(B)$  et que  $A \subset B$  c.-à-d. on retrouve dans B tous les items de A. Remarquons une propriété très importante du support :  $SUP(B) \leq SUP(A)$ . En particulier, si A n'est

pas fréquent, alors B ne le sera pas également. Ce résultat permet de réduire considérablement l'espace de recherche lors de l'extraction des itemsets fréquents dans une base de données.

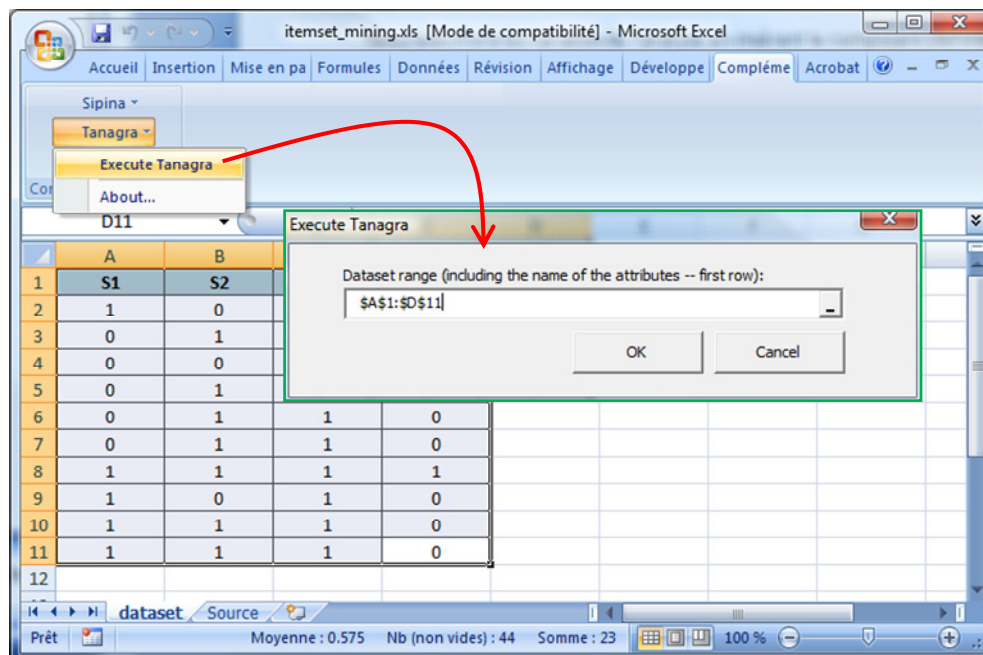
**Itemset fermé (closed itemset).** Un itemset fréquent est dit fermé si aucun de ses supersets n'a de support identique. Autrement dit, tous ses supersets ont un support strictement plus faible. Dans notre exemple ci-dessus,  $\{S_1, S_3\}$  est fermé car aucun de ses supersets n'a de support égal à  $5/10$  :  $SUP(\{S_1, S_2, S_3\}) = 3/10$ ,  $SUP(\{S_1, S_3, S_4\}) = 1/10$ .

**Itemset maximal (maximal itemset).** Un itemset est dit maximal si aucun de ses supersets n'est fréquent. Dans notre exemple ci-dessus,  $\{S_1, S_2, S_3\}$  est maximal car son superset  $\{S_1, S_2, S_3, S_4\}$  (il n'y en a qu'un) n'est pas fréquent avec un support de  $1/10$ .

**Itemset générateur (generator itemset).** Un itemset A est dit générateur s'il n'existe aucun itemset B tel que  $B \subset A$  et que  $SUP(B) = SUP(A)$ . Autrement dit, l'itemset est générateur si tous ses sous-itemsets ont un support strictement supérieur. Dans notre exemple,  $\{S_1, S_2, S_3\}$  de support  $4/10$  n'est pas générateur puisqu'on trouve  $\{S_1, S_2\}$  avec un support identique. Il en est de même en ce qui concerne  $\{S_1, S_3\}$  à cause de  $\{S_1\}$ . En revanche,  $\{S_2, S_4\}$ , de support  $2/10$ , est générateur parce que  $SUP(\{S_2\}) = 7/10$  et  $SUP(\{S_4\}) = 3/10$ .

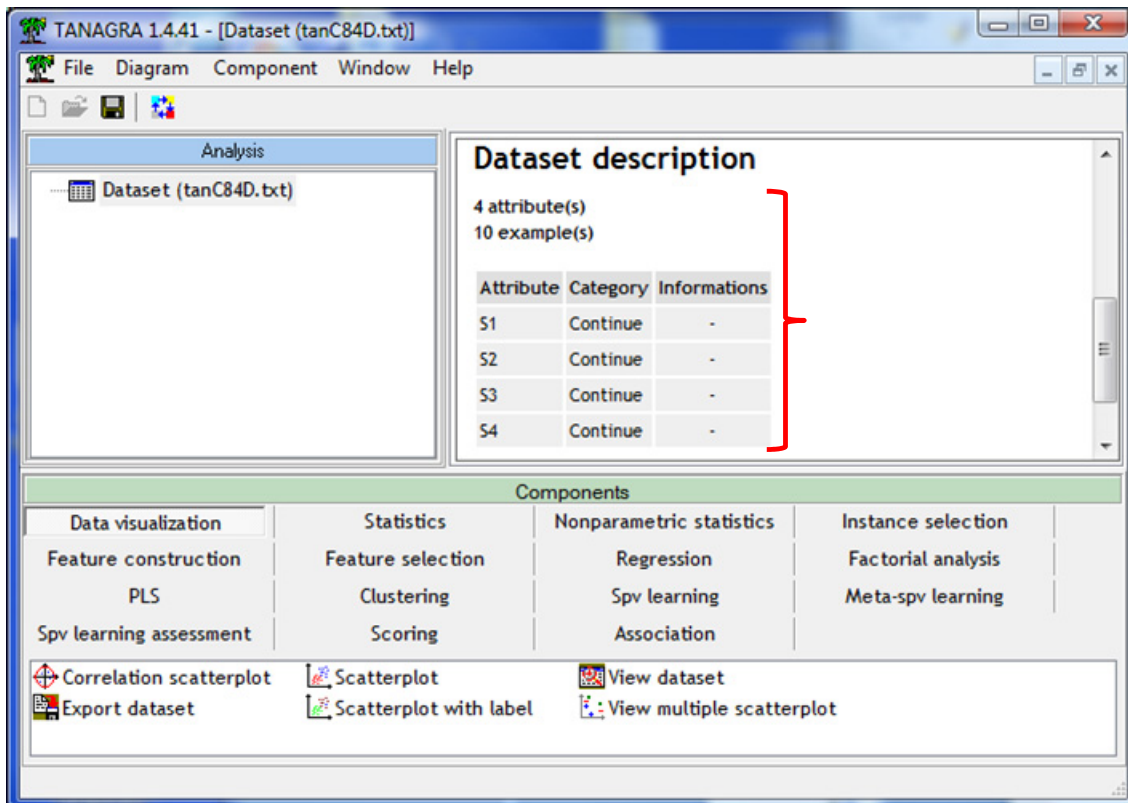
### 3 Extraction des itemsets à l'aide de Tanagra

Nous chargeons le fichier « [itemset\\_mining.xls](#) » dans le tableur Excel. Nous sélectionnons la plage de données, puis nous actionnons le menu TANAGRA / EXECUTE TANAGRA installé à l'aide de la macro complémentaire tanagra.xla<sup>3</sup>.

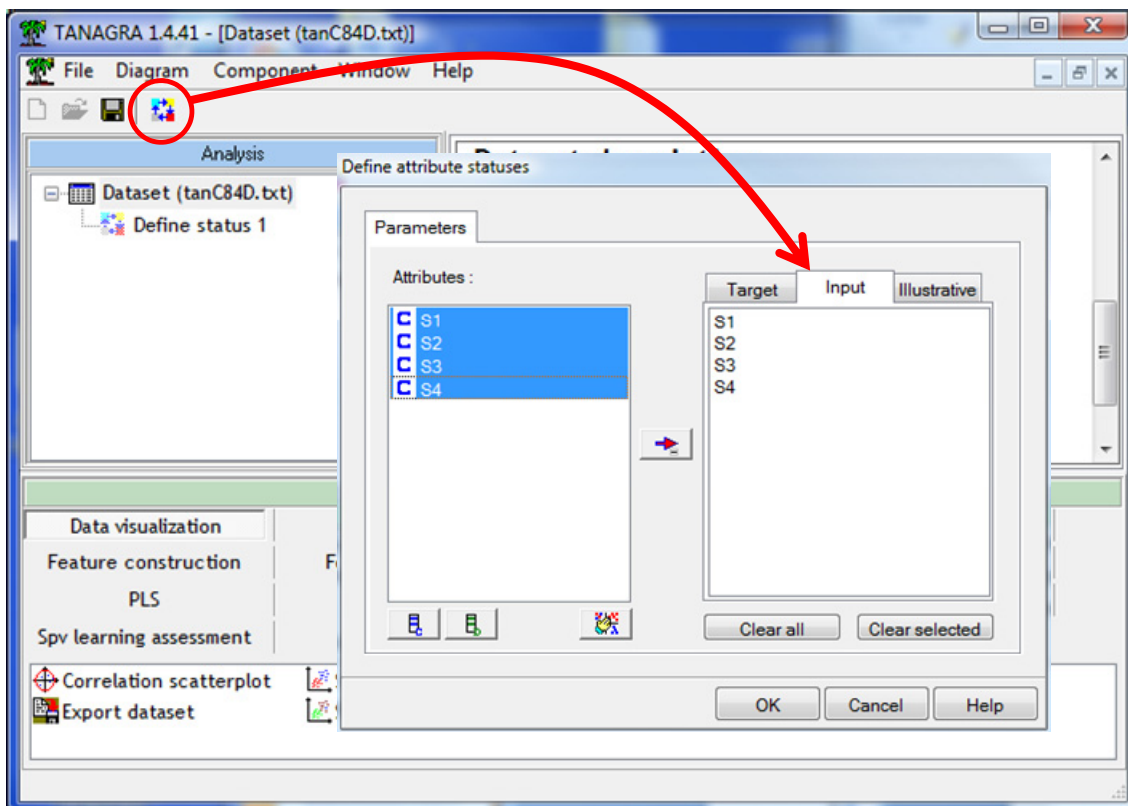


Tanagra est automatiquement démarré et les données chargées.

<sup>3</sup> Voir <http://tutoriels-data-mining.blogspot.com/2010/08/ladd-in-tanagra-pour-excel-2007-et-2010.html> pour l'installation et l'utilisation de la macro dans Excel 2007 et 2010. Elle est également fonctionnelle sous les versions antérieures d'Excel (<http://tutoriels-data-mining.blogspot.com/2008/03/importation-fichier-xls-excel-macro.html> - Excel 97 à 2003). Enfin, une procédure analogue existe pour OpenOffice et LibreOffice (voir <http://tutoriels-data-mining.blogspot.com/2011/07/tanagra-addon-pour-openoffice-33.html>).

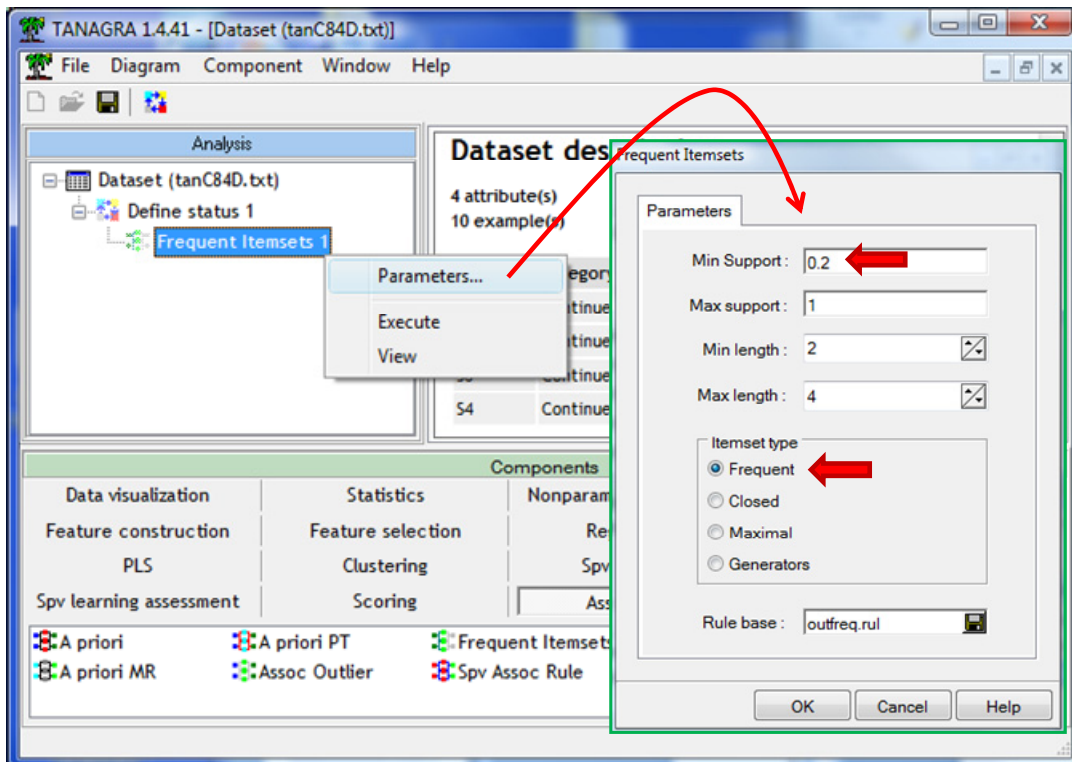


Nous spécifions les variables de l'analyse en insérant le composant DEFINE STATUS. Nous plaçons toutes les variables en INPUT.

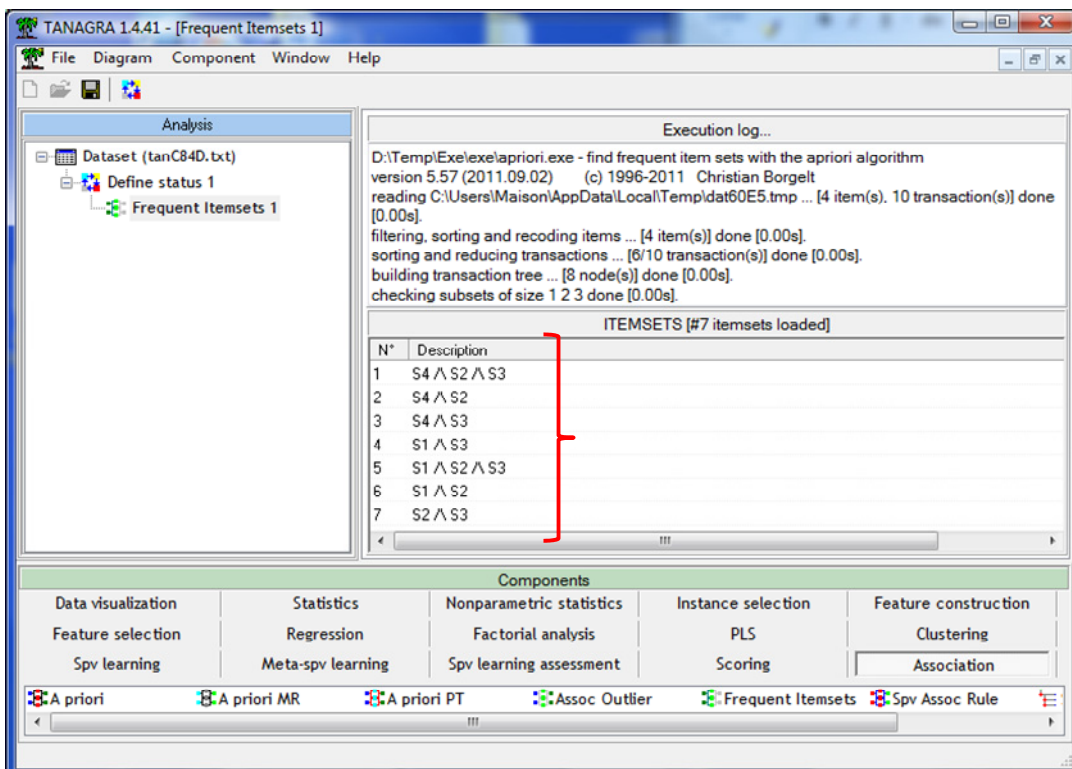


### 3.1 Itemsets fréquents

Pour produire les itemsets fréquents, nous ajoutons le composant FREQUENT ITEMSETS dans le diagramme. Nous actionnons le menu PARAMETERS afin de spécifier les paramètres de l'analyse.



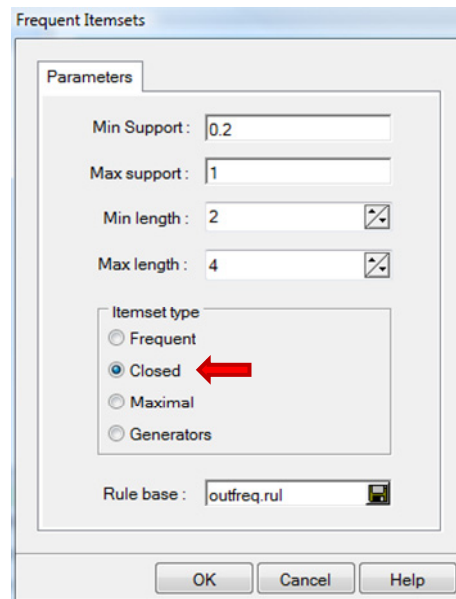
Nous descendons le support minimum à 20%. Nous ne modifions pas les autres paramètres. Par défaut, l'outil extrait les itemsets fréquents de cardinal compris entre MIN LENGTH = 2 et MAX LENGTH = 4. Les itemsets composés d'un singleton ne sont donc pas générés.



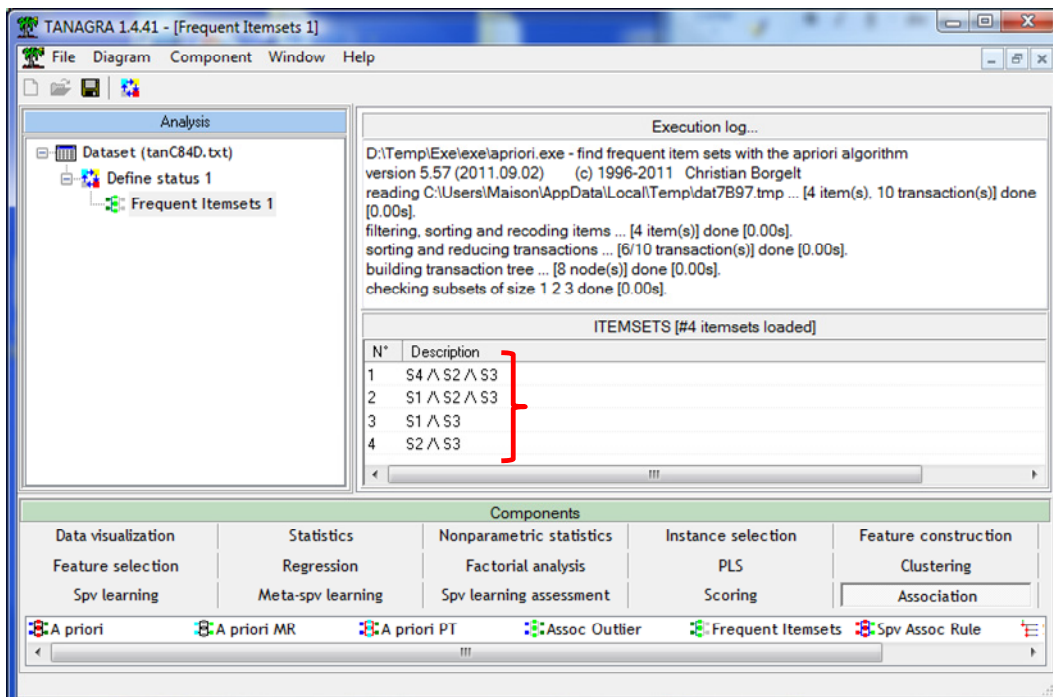
Il ne nous reste plus qu'à cliquer sur le menu contextuel VIEW. Le module « apriori.exe » est lancé en sous main dixit la fenêtre de suivi. Les itemsets – au nombre de 7 – s'affichent dans la partie inférieure. Les résultats concordent avec le graphe des itemsets (les itemsets non-grisés dans Figure 1), à l'exception des itemsets de cardinal 1 que nous avons sciemment mis de côté.

### 3.2 Itemsets fermés

Pour obtenir les itemsets fermés, nous re-paramétrons le composant en actionnant le menu contextuel PARAMETERS. Nous choisissons la seconde option (ITEMSET TYPE = CLOSED).

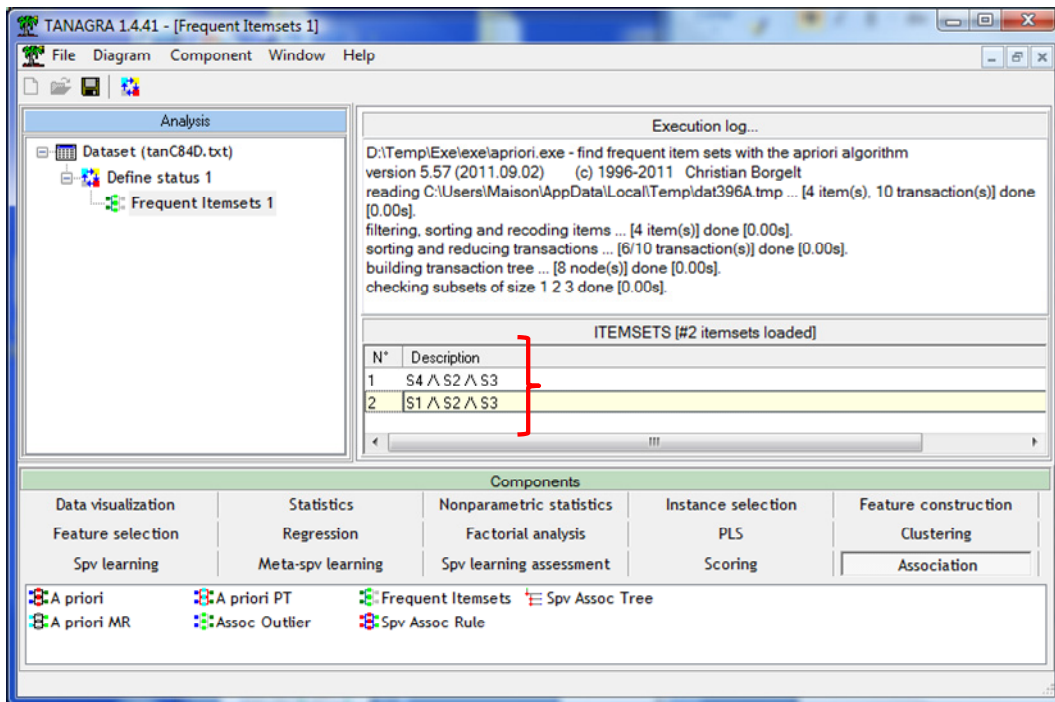


Nous validons et nous cliquons sur VIEW. Nous obtenons 4 itemsets (en vert et bleu dans la Figure 1, à l'exception des singletons).



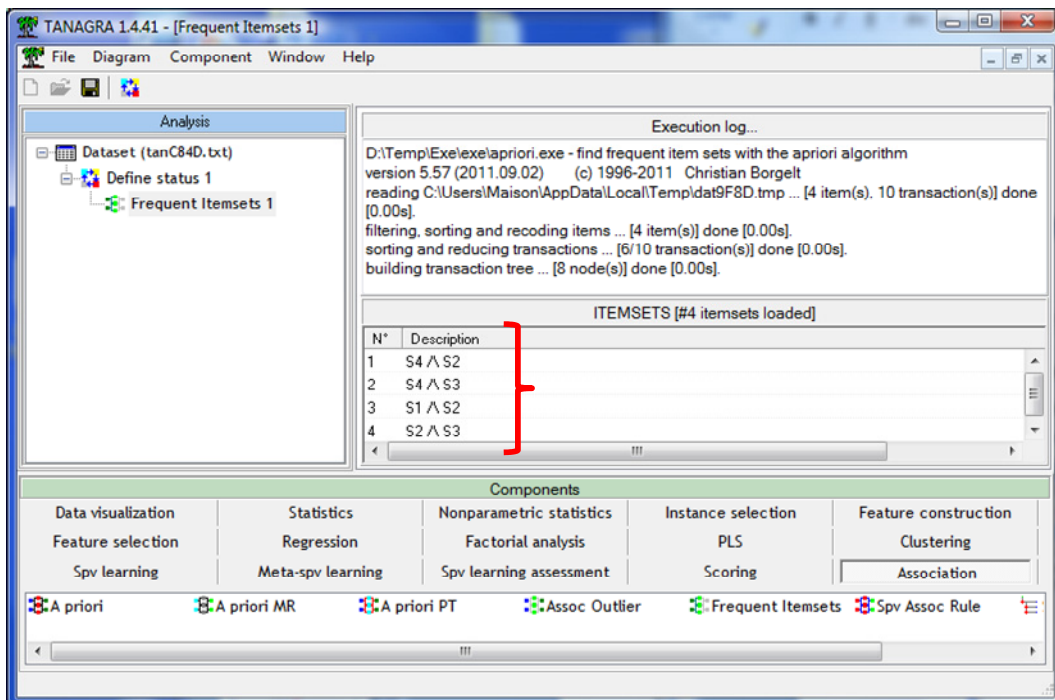
### 3.3 Itemsets maximaux

Nous réitérons le même schéma avec l'option (ITEMSET TYPE = MAXIMAL) dans la boîte de paramétrage. Nous obtenons à la sortie 2 itemsets, ceux qui sont en vert dans le graphe (Figure 1).



### 3.4 Itemsets générateurs

Enfin, pour obtenir les itemsets générateurs, nous sélectionnons la dernière option (ITEMSET TYPE = GENERATORS) dans la boîte de paramétrage. Nous obtenons 4 itemsets de cardinal égal à 2.



## 4 Extraction des itemsets à l'aide de R (package « arules »)

La procédure « **apriori** » du package « **arules**<sup>4</sup> » est également basé sur la bibliothèque de Borgelt. Il est donc possible de reproduire les résultats ci-dessus en introduisant les commandes et les paramètres adéquats dans le logiciel R. Nous décrivons succinctement les principales commandes dans ce qui suit, en leur associant les résultats affichés par l'outil.

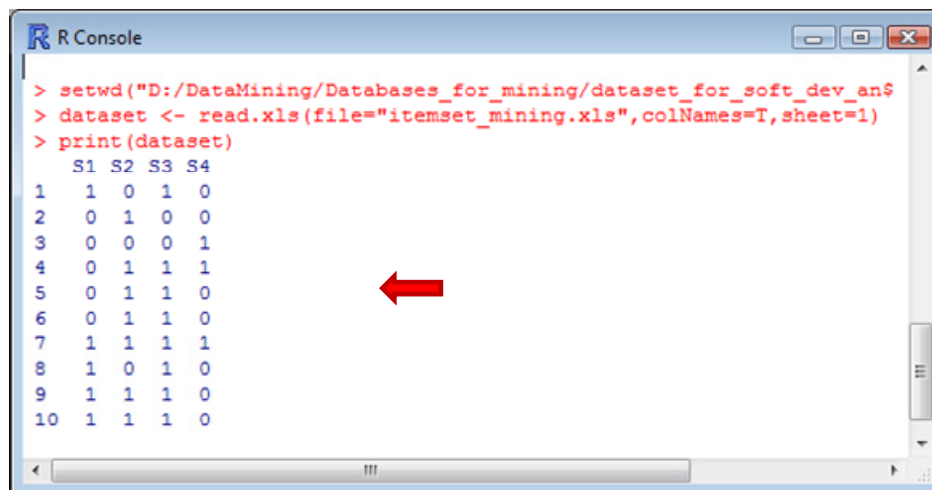
### 4.1 Importation des données

Nous utilisons la commande « **read.xls** » du package « **xlsReadWrite**<sup>5</sup> » pour lire le fichier de données « **itemset\_mining.xls** ». Voici les instructions associées.

```
#vider la mémoire
rm(list=ls())

#importation des données
library(xlsReadWrite) #chargement de la bibliothèque
setwd("...") #modifier le répertoire de travail
#la première ligne correspond au nom des items
#les données sont situées dans la première feuille du classeur XLS
dataset <- read.xls(file="itemset_mining.xls",colNames=T,sheet=1)
#affichage des valeurs
print(dataset)
```

R nous fournit les informations suivantes.



```
R Console
> setwd("D:/DataMining/Databases_for_mining/dataset_for_soft_dev_an$
> dataset <- read.xls(file="itemset_mining.xls",colNames=T,sheet=1)
> print(dataset)
  S1 S2 S3 S4
1  1  0  1  0
2  0  1  0  0
3  0  0  0  1
4  0  1  1  1
5  0  1  1  0
6  0  1  1  0
7  1  1  1  1
8  1  0  1  0
9  1  1  1  0
10 1  1  1  0
```

### 4.2 Extraction des itemsets fréquents

Pour extraire les itemsets fréquents, nous chargeons au préalable la bibliothèque, nous spécifions les paramètres de l'analyse, enfin nous lançons les calculs. Attention, pour que la procédure comprenne bien que les données se présentent sous la forme d'une matrice 0/1 de présence-absence des items dans les transactions, nous devons effectuer un transtypage avec la commande **as.matrix(...)**.

<sup>4</sup> <http://cran.r-project.org/web/packages/arules/index.html>

<sup>5</sup> <http://cran.r-project.org/web/packages/xlsReadWrite/index.html>

```

#chargement du package
library(arules)

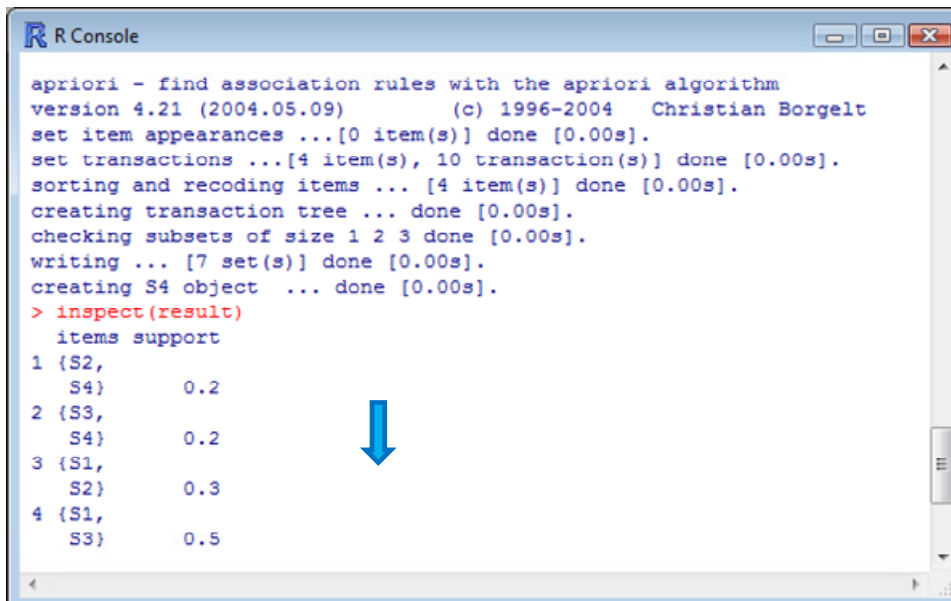
#paramétrage de la méthode
params <- list(supp = 0.2, minlen = 2, maxlen = 4, target="frequent itemsets")

#lancement des calculs
result <- apriori(as.matrix(dataset), parameter = params)

#affichage des itemsets
inspect(result)

```

R affiche les itemsets, les résultats concordent en tous points à ceux de Tanagra. Ce n'est guère étonnant puisque les deux outils s'appuient sur la même bibliothèque de calcul.



```

R Console
apriori - find association rules with the apriori algorithm
version 4.21 (2004.05.09)      (c) 1996-2004  Christian Borgelt
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[4 item(s), 10 transaction(s)] done [0.00s].
sorting and recoding items ... [4 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [7 set(s)] done [0.00s].
creating S4 object ... done [0.00s].
> inspect(result)
  items support
1 {S2,
  S4}      0.2
2 {S3,
  S4}      0.2
3 {S1,
  S2}      0.3
4 {S1,
  S3}      0.5

```

**Remarque :** Notons cependant que le package « arule » est basé sur la version 4.21 de « apriori.exe » de Borgelt. L'extraction des itemsets générateurs n'est pas disponible.

### 4.3 Extraction des autres types d'itemsets

Les résultats étant de la même teneur que ceux de Tanagra, nous nous contentons de décrire les paramètres de calculs pour chaque type d'itemset dans cette sous-section.

```

#extraction des itemsets fermés
params <- list(supp = 0.2, minlen = 2, maxlen = 4, target="closed frequent itemsets")
result <- apriori(as.matrix(dataset), parameter = params)
inspect(result)

#extraction des itemsets maximaux
params <- list(supp = 0.2, minlen = 2, maxlen = 4, target="maximally frequent itemsets")
result <- apriori(as.matrix(dataset), parameter = params)
inspect(result)

```

## 5 Conclusion

La sortie de la version 1.4.41 de Tanagra a été l'occasion de mettre à jour la bibliothèque externe « apriori.exe » de Borgelt (version 5.57), nettement plus performante en termes de temps de traitements<sup>6</sup>. Nous en avons profité pour introduire un nouveau composant de génération des itemsets fréquents (FREQUENT ITEMSETS), toujours basée sur la même bibliothèque, que nous décrivons dans ce didacticiel.

---

<sup>6</sup> Voir <http://tutoriels-data-mining.blogspot.com/2011/09/mise-jour-de-apriori-pt.html>