

1 Topic

Mining frequent itemsets with the FREQUENT ITEMSETS component.

Searching regularities from dataset is the main goal of the data mining. They may have various representations. In the market basket analysis, we search the co occurrences of goods (items) i.e. the goods which are often purchased simultaneously. They are called "frequent itemset". For instance, one result may be "milk and bread are purchased simultaneously in 10% of caddies".

Frequent itemset mining is often presented as the preceding step of the association rule learning algorithm. At the end of the process, we highlight the direction of the relation. We obtain rules. For instance, a rule may be "90% of the customers which buy milk and bread will purchase butter also". This kind of rule can be used in various manners. For instance, we can promote the sales of milk and bread in order to increase the sales of butter.

In fact, frequent itemsets provide also valuable information. Detecting the goods which are purchased simultaneously enables to understand the relation between them. It is a kind of variant of the clustering analysis. We search the items which come together. For instance, we can use this kind of information in order to reorganize the shelves of the store.

In this tutorial, we describe the use of the FREQUENT ITEMSETS component under Tanagra. It is based on the Borgelt's "apriori.exe" program. We use a very small dataset. It enables to everyone to reproduce manually the calculations. But, in a first time, we describe some definitions about the frequent itemset mining process.

2 Frequent itemset mining

Our data file contains 10 instances (transactions) and 4 items¹.

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

Each row corresponds to a customer of an insurance company. Each column is an insurance contract. E.g. the customer n°1 has the contracts S1 and S3, etc. **The objective is to identify products (contracts) that are placed together.**

Item. An item corresponds to a product. We have 4 items (S1, S2, S3 and S4) here.

Support. The support of an item corresponds to the number of transactions in which it appears. For instance, the support of {S1} is 5. The support can be also computed as a percentage. In this case, we divide the absolute support by the number of the transactions e.g. $SUP(\{S1\}) = 5/10 = 50\%$.

¹ <http://www.dataminingarticles.com/closed-maximal-itemsets.html>

Itemset. An itemset is a set of items, e.g. $\{S_1, S_2\}$ is an itemset with the cardinal $CARD(\{S_1, S_2\}) = 2$. The support of an itemset counts the number of transactions in which the items appear simultaneously e.g. $SUP(\{S_1, S_2\}) = 3/10 = 30\%$. An itemset may be a singleton i.e. an itemset with a cardinal equal to 1.

Frequent itemset. An itemset is called frequent if its support is higher than a user-specified value. This last one is the main parameter of the algorithm. For instance, if we set the minimum support to 2 (or 20%) for our dataset, we obtain the following frequent itemsets (all non-grayed itemset into the graph).

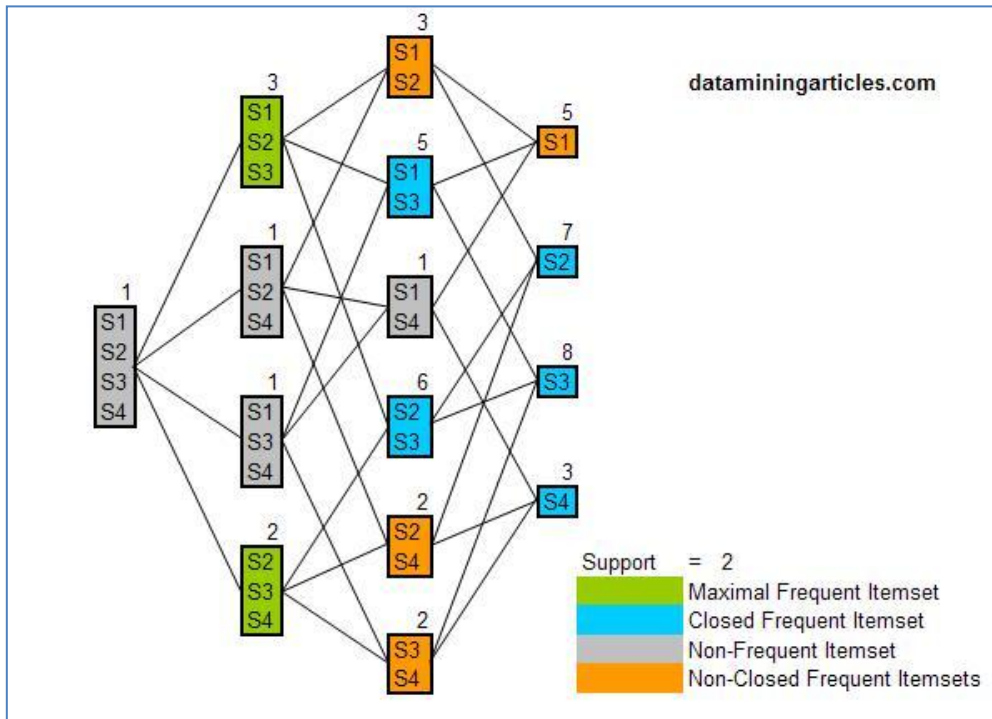


Figure 1 – Mining the frequent itemsets

Superset. A superset is an itemset defined in relation to another itemset. For instance, $B = \{S_1, S_2, S_3\}$ is a superset of $A = \{S_1, S_2\}$ because $CARD(A) < CARD(B)$ and $A \subset B$ i.e. all the items present into A are also present into B. If A is not frequent, then B is not frequent also. This property enables to decrease considerably the computation time during the extraction of frequent itemsets from a database.

Closed itemset. A frequent itemset is called "closed" if all its supersets have a support lower than its own support. For instance $\{S_1, S_3\}$ [$SUP(\{S_1, S_3\}) = 5/10$] is a closed frequent itemset because none of its supersets has the same support: $SUP(\{S_1, S_2, S_3\}) = 3/10$, $SUP(\{S_1, S_3, S_4\}) = 1/10$.

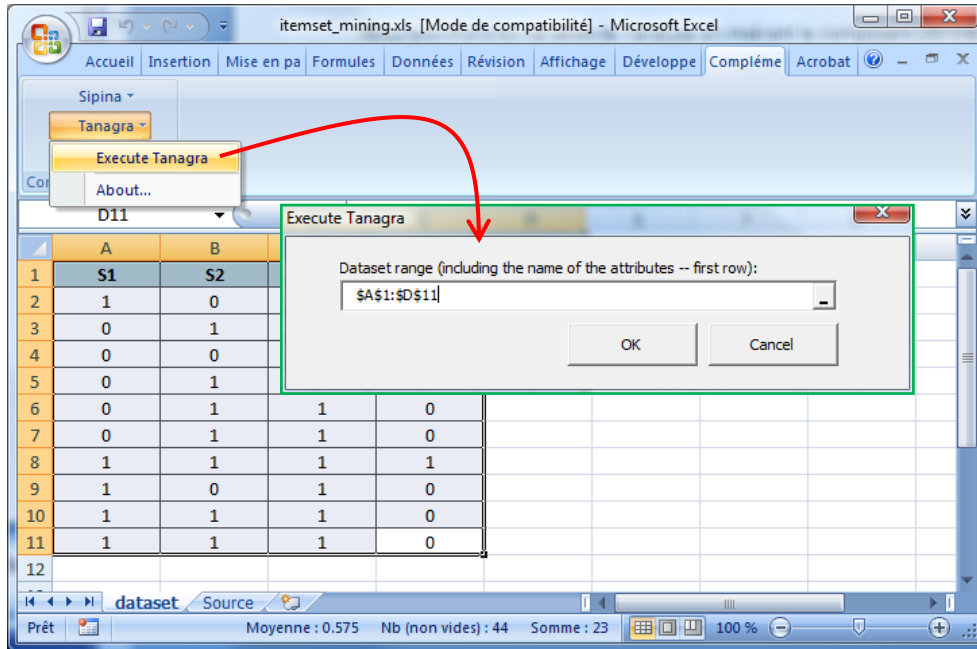
Maximal itemset. A frequent itemset is called "maximal" if none of its supersets is frequent. For instance, $\{S_1, S_2, S_3\}$ is maximal because it has only one superset $\{S_1, S_2, S_3, S_4\}$ and this last one is not frequent ($SUP(\{S_1, S_2, S_3, S_4\}) = 1/10 < 2/10$).

Generator itemset. A is a generator itemset if it does not exist an itemset B such as $B \subset A$ and $SUP(B) = SUP(A)$. In other words, an itemset is generator if all its sub-itemsets have a support strictly higher than its own support. For instance, $\{S_1, S_2, S_3\}$ (support = $4/10$) is not generator because $\{S_1,$

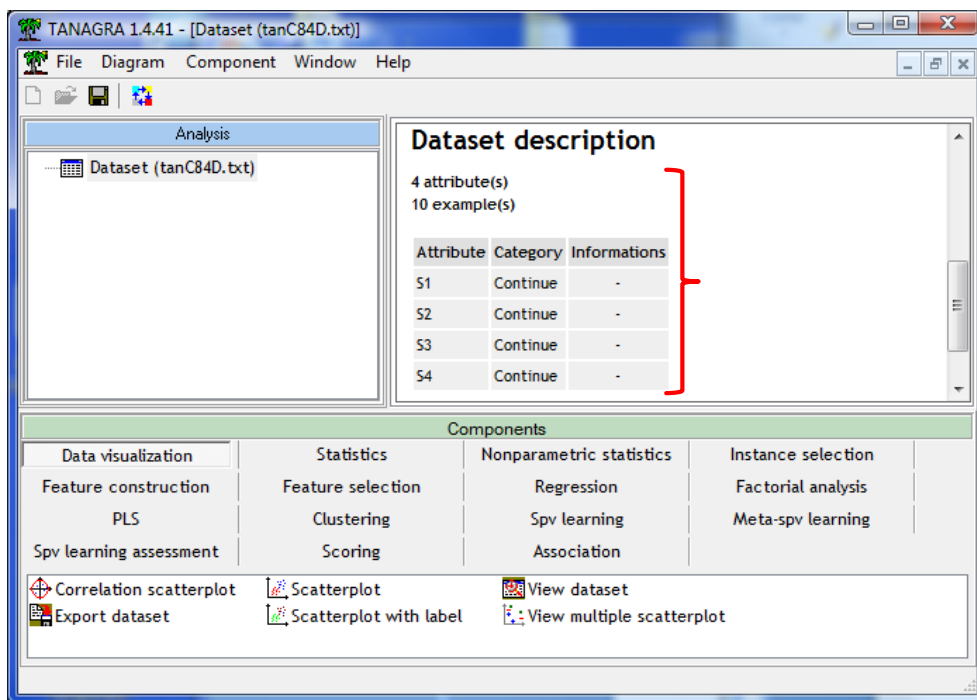
S_2 , one of its sub-itemsets, have the same support (4/10). Whereas $\{S_2, S_4\}$ (support = 2/10) is generator because $SUP(\{S_2\}) = 7/10$ and $SUP(\{S_4\}) = 3/10$.

3 Itemset mining with Tanagra

We load the « [itemset mining.xls](#) » data file into Excel. We select the data range; then we click on the TANAGRA / EXECUTE TANAGRA menu which is installed with the “tanagra.xla” add-in².

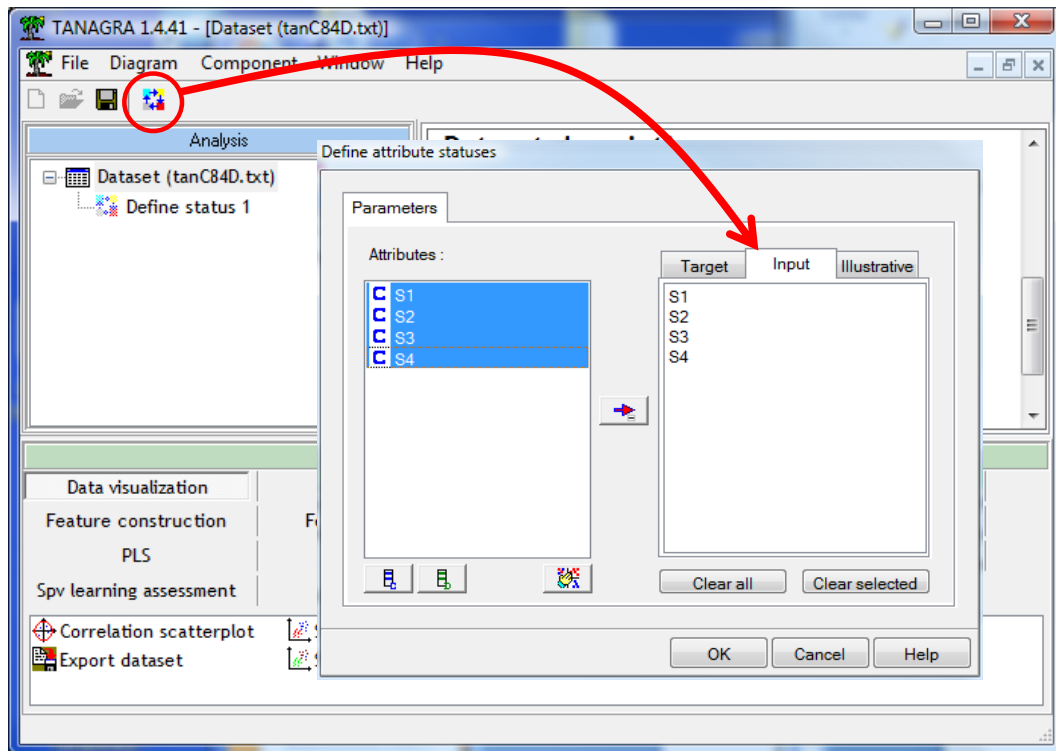


Tanagra is automatically started and the dataset is loaded.



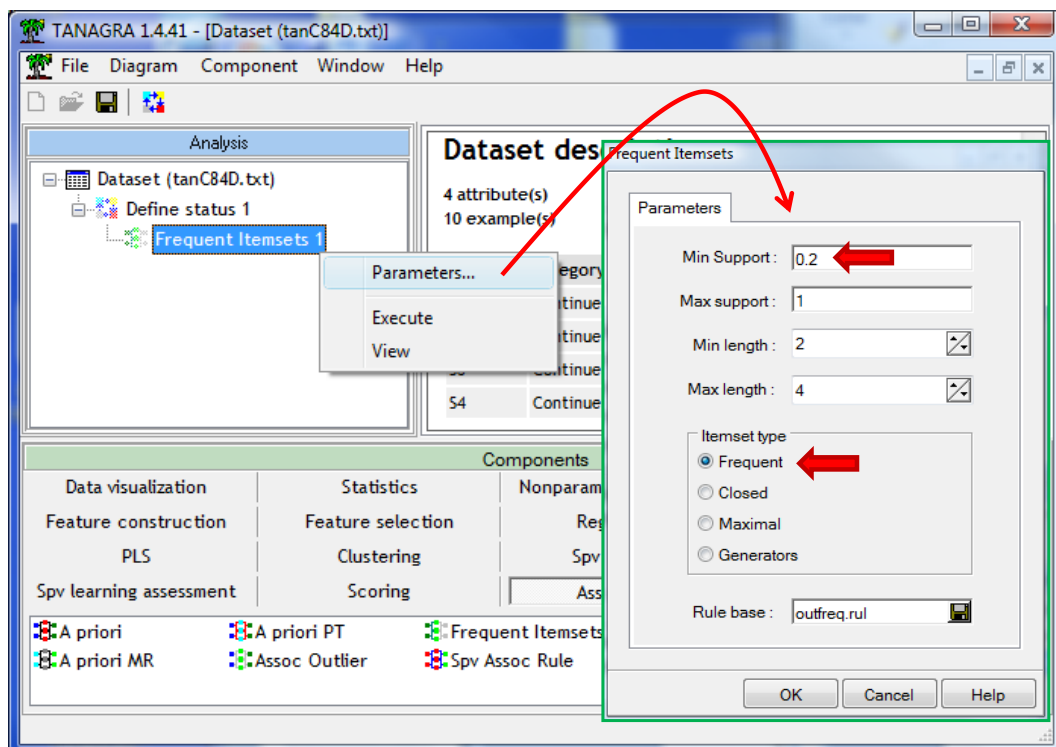
² <http://data-mining-tutorials.blogspot.fr/2010/08/tanagra-add-in-for-office-2007-and.html> for Excel;
<http://data-mining-tutorials.blogspot.fr/2011/07/tanagra-add-on-for-openoffice-calc-33.html> for Open Office.

We select the variables for the analysis using the DEFINE STATUS component. We set all the variables as INPUT ones.

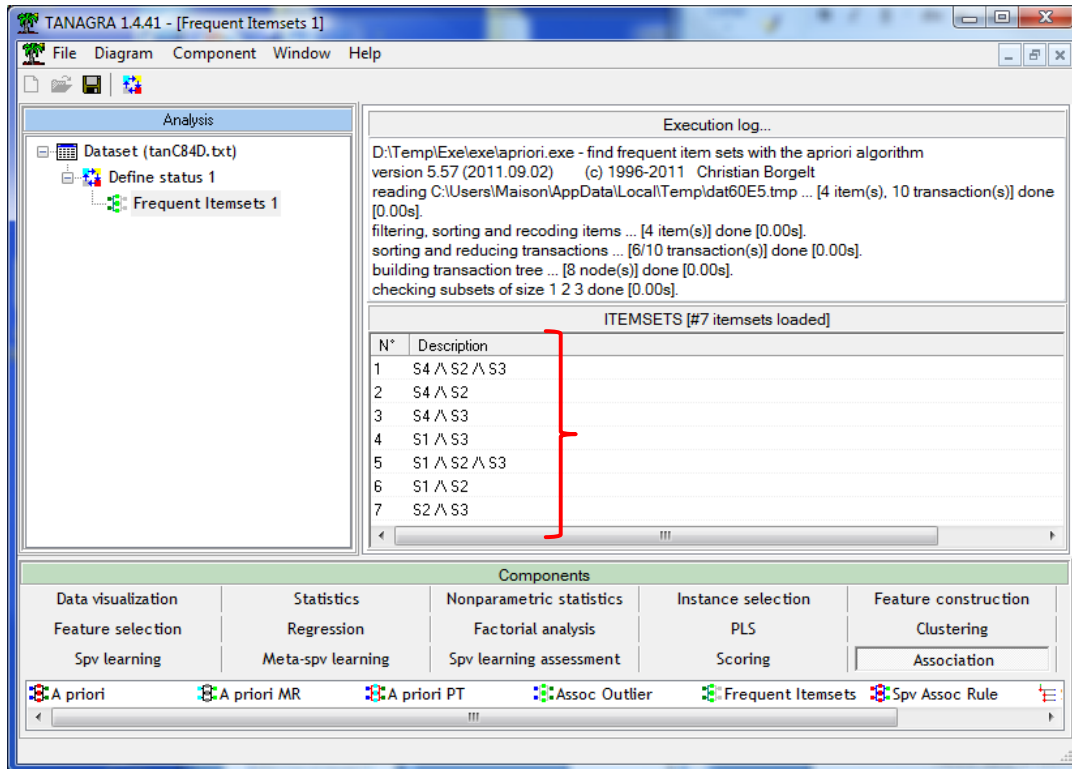


3.1 Frequent itemsets

To extract the frequent itemsets, we add the component into the diagram. We click on the PARAMETERS contextual menu to specify the settings of the analysis.



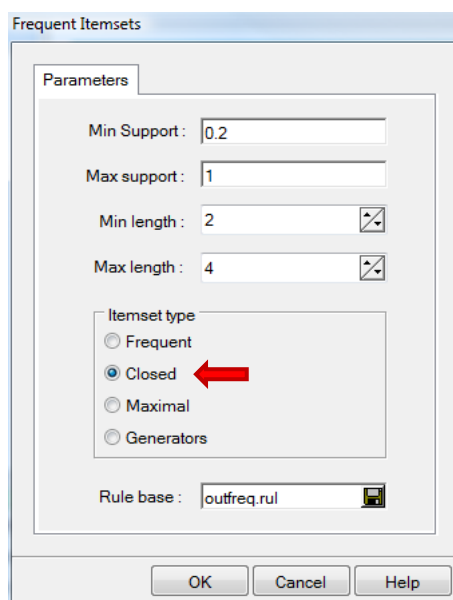
We set the minimal support to 20%. We do not modify the other parameters. By default, the tool extracts the frequent itemsets of cardinal between MIN LENGTH = 2 and MAX LENGTH = 4. So, the itemsets composed of a singleton are not generated.



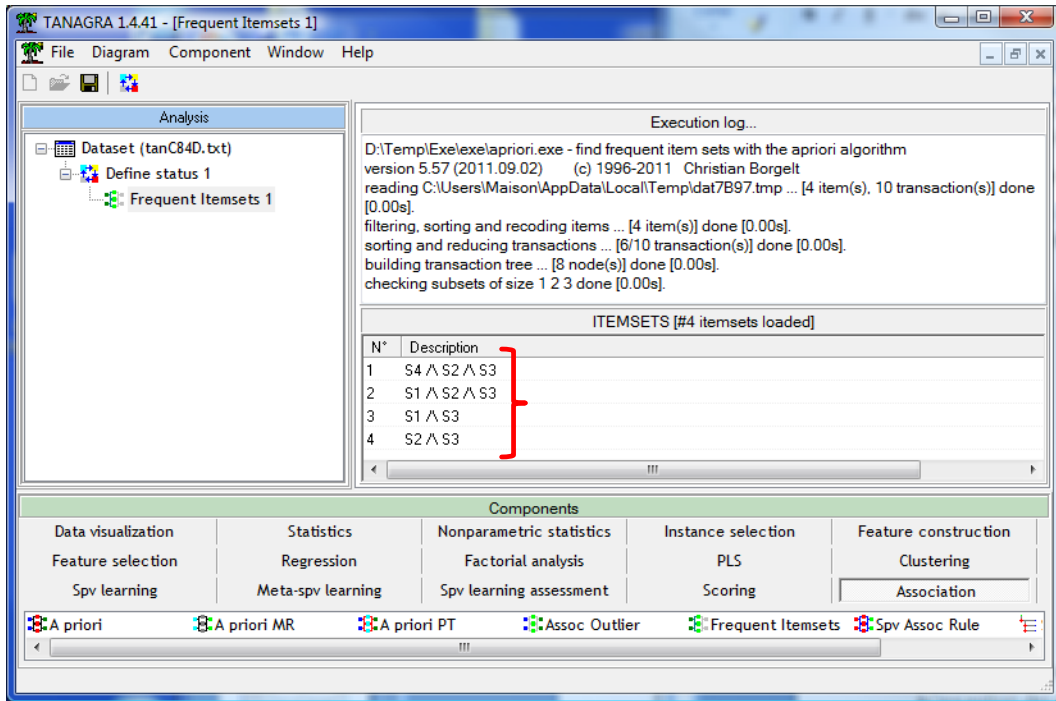
We click on the VIEW menu to start the analysis. The "apriori.exe" program is launched. The extracted itemsets are displayed into the visualization window.

3.2 Closed itemsets

To obtain the closed itemsets, we modify the settings by clicking again on the PARAMETERS menu. We set the following values (especially **ITEMSET TYPE = CLOSED**).

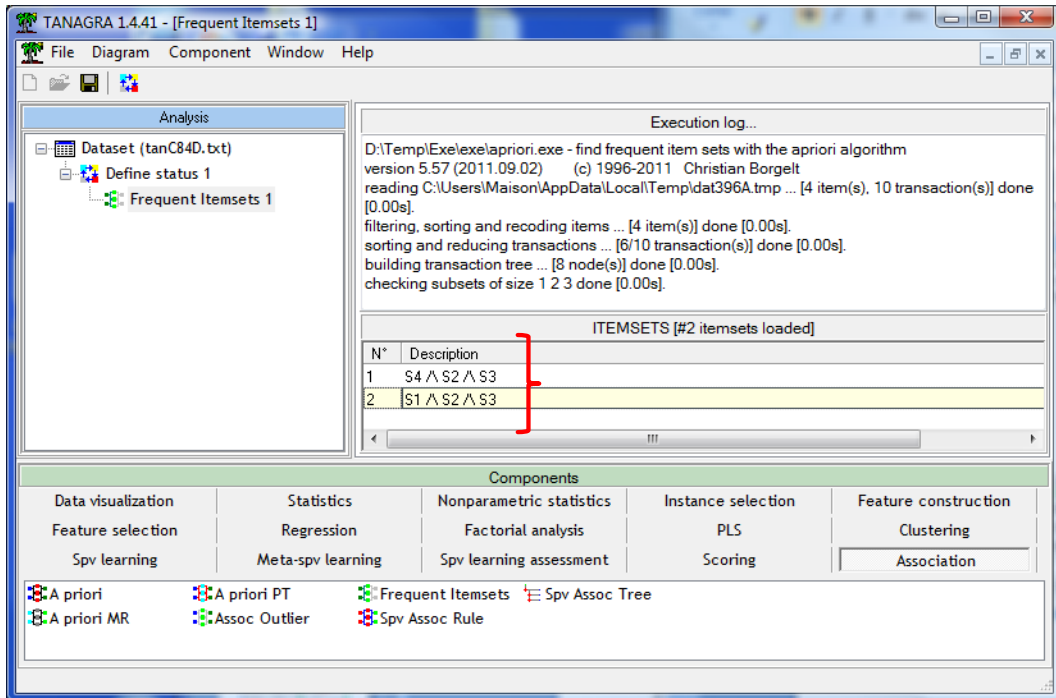


We validate our choice. We click on the VIEW menu, we obtain 4 closed itemsets.



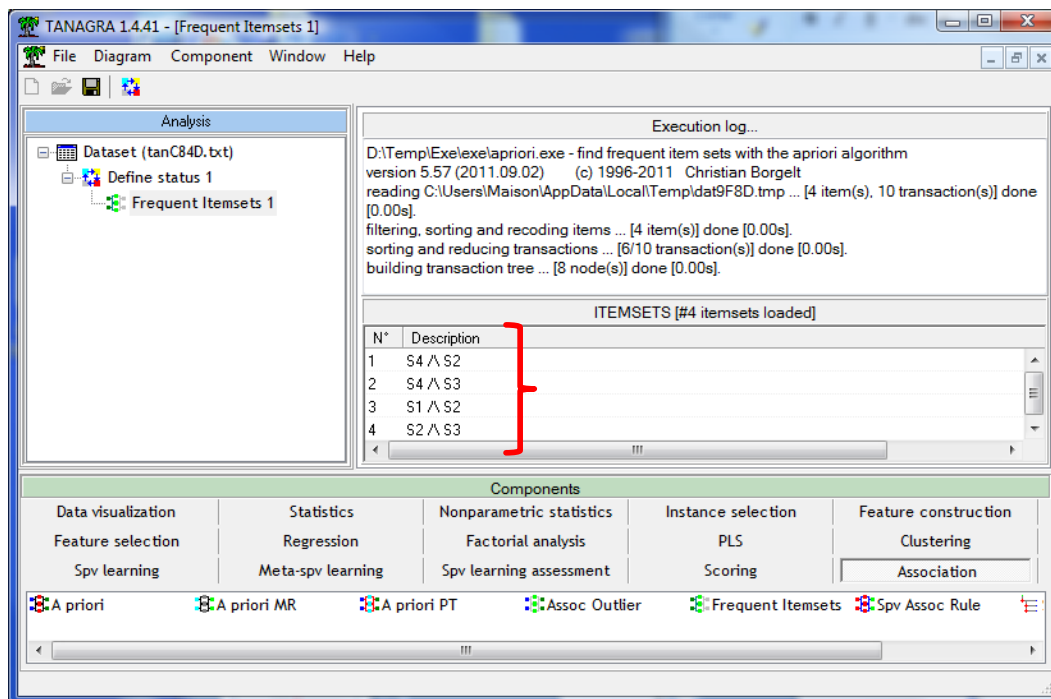
3.3 Maximal itemsets

We set new parameters (**ITEMSET TYPE = MAXIMAL**) into the dialog settings. We obtain 2 maximal itemsets.



3.4 Generator Itemsets

Last, to obtain generator itemsets, we set (**ITEMSET TYPE = GENERATORS**) into the dialog settings. We obtain 4 itemsets.



4 Extracting itemsets with R (« arules » package)

The "apriori" procedure based on the "arules" package³ uses also the Borgelt's program. We can thus obtain the same results as above if we set the same settings for the calculations.

4.1 Data file importation

We use the « [read.xls](#) » command of the « [xlsReadWrite](#)⁴ » package to read the data file.

```
#clear la mémoire
rm(list=ls())

#data importation
library(xlsReadWrite) #loading the package
setwd("...") #modifying the directory

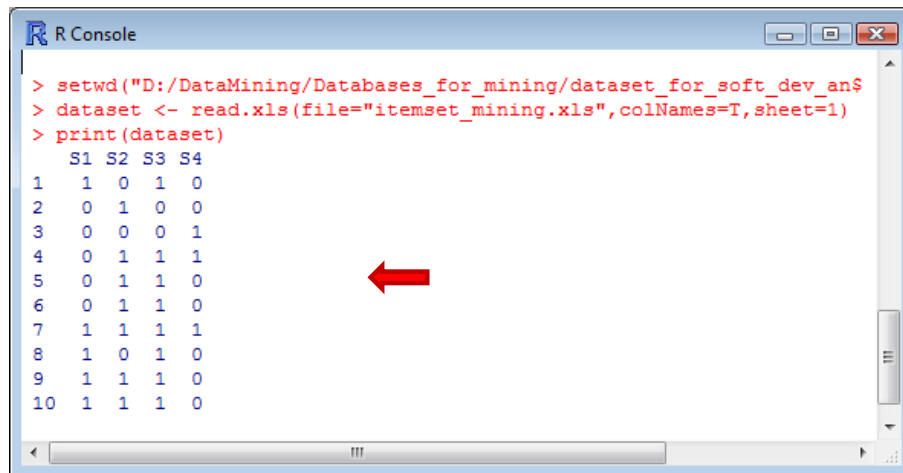
#the first row corresponds to the name of the variables
#the dataset is located in the first sheet
dataset <- read.xls(file="itemset_mining.xls",colNames=T,sheet=1)

#printing the values
print(dataset)
```

R provides the following results.

³ <http://cran.r-project.org/web/packages/arules/index.html>

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



```

> 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 Extracting the frequent itemsets

To extract the frequent itemsets, we load the "arules" library; then, we set the parameters of the analysis; last, we launch calculations. We note that we must transform the data frame into a matrix with the `as.matrix(...)` command.

```
#loading the package
```

```
library(arules)
```

```
#set the settings
```

```
params <- list(supp = 0.2, minlen = 2, maxlen = 4, target="frequent itemsets")
```

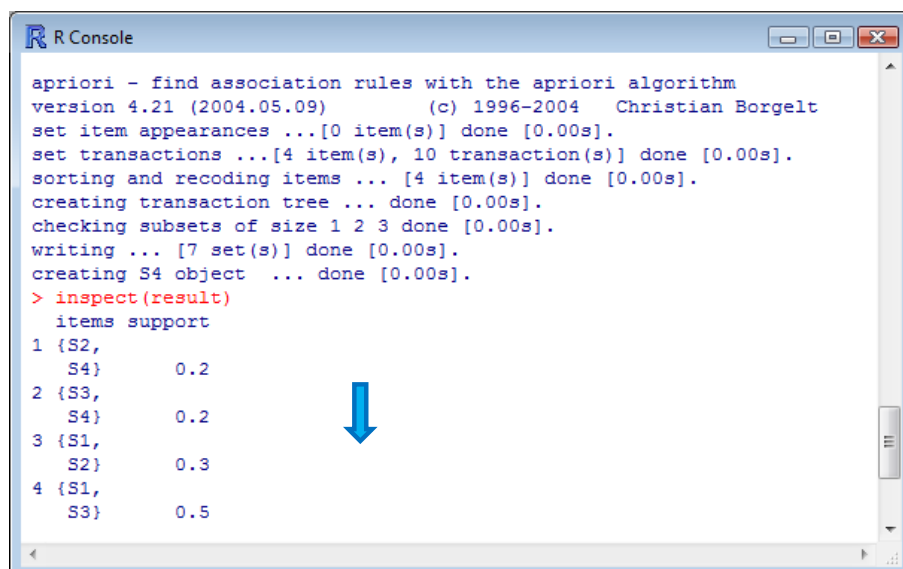
```
#computing the rules
```

```
result <- apriori(as.matrix(dataset), parameter = params)
```

```
#printing the itemsets
```

```
inspect(result)
```

R shows the itemsets. We obtain the same results as Tanagra.



```

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,      0.2
   S4}
2 {S3,      0.2
   S4}
3 {S1,      0.3
   S2}
4 {S1,      0.5
   S3}

```

Note: We note that the package "arule" is based on the 4.21 version of "apriori.exe". The extraction of the generator itemsets is not available.

4.3 Extracting the other kinds of itemsets

The results being the same that those of TANAGRA, we describe only the parameters of calculations for each type of itemset in this sub-section.

#mining the closed itemsets

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

#mining the maximal itemsets

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

5 Conclusion

We update the external library "apriori.exe" of Borgelt (version 5.57) in the Tanagra 1.4.41. It is definitely faster⁵. We added in this occasion a new component for the extraction of the frequent itemsets (FREQUENT ITEMSETS) which is also based on the same program.

⁵ <http://data-mining-tutorials.blogspot.fr/2011/09/priori-pt-updated.html>