1 Topic

Describing the post-pruning process during the induction of decision trees (CART algorithm, Breiman and al., 1984 – C-RT component into TANAGRA).

Determining the appropriate size of the tree is a crucial task in the decision tree learning process. It determines its performance during the deployment into the population (the generalization process). There are two situations to avoid: the under-sized tree, too small, poorly capturing relevant information in the training set; the over-sized tree capturing specific information of the training set, which specificities are not relevant to the population. In both cases, the prediction model performed poorly during the generalization phase.

The trade-off between the tree size and the generalization performance is often illustrated by a graphical representation where we see that there is an "optimal" size of the tree (Figure 1). While the error on the training sample decreases as the tree size increases, the true error rate is stagnant, then deteriorates when the tree is oversized.

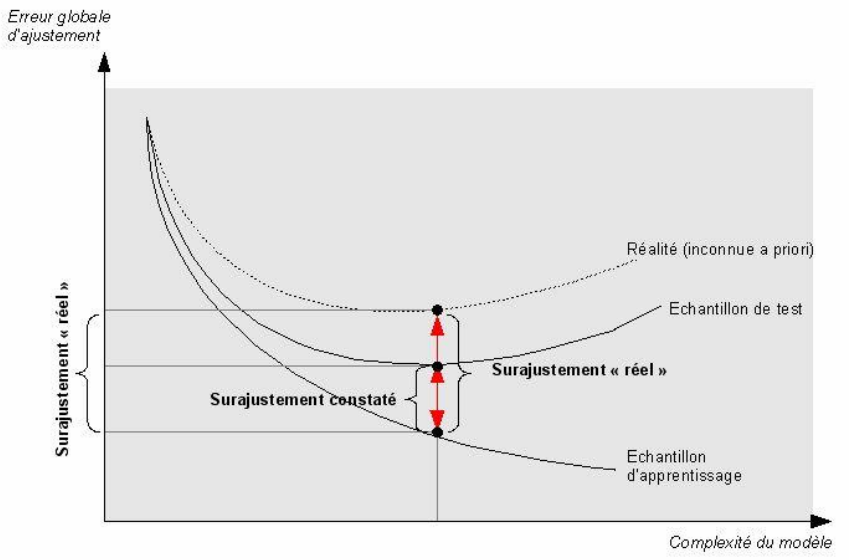


Figure 1 – Tree size and generalization error rate (Source: http://fr.wikipedia.org/wiki/Arbre_de_décision)

Determining the appropriate size of the tree is thus to select, among the many solutions, the more accurate tree with the smallest size. Simplifying decision tree is advantageous, beyond the generalization performance point of view. Indeed, a simpler decision is easier to deploy and the interpretation of the tree is also easier.

In their book, Breiman and al. (CART method, 1984) are the first which identify clearly the overfitting problem in the induction tree context. They propose the post-pruning process to avoid this problem. This idea was implemented later by Quinlan in the C4.5 method (1993), but in a different way.

Basically, the construction is performed in two steps. First, during the growing phase, in a top down approach, we create the tree by splitting recursively the nodes. Second, during the pruning phase, in

a bottom up approach, we prune the tree by removing the irrelevant branches i.e. we transform a node to a leaf by removing the subsequent nodes. This is during this second step that we try to select the most performing tree.

In the simplest version of CART, the training set is subdivided into two parts: the growing set, which used during the growing phase; and the pruning set, which used during the pruning phase. The aim is to search the optimal tree on this pruning set.

To avoid the overfitting on the pruning set, CART implements two strategies. (1) CART does not evaluate all the candidate subtrees in order to detect the best one. It uses the cost complexity pruning approach in order to highlight the candidate trees for the post-pruning. This process enables above all to insert a kind a smoothing in the exploration of the solutions. (2) Instead of the selection of the best subtree, this one which minimizes the error rate, CART selects the simplest tree based on the 1-SE rule i.e. the simplest tree for which the error rate is not upper than the best pruning error rate plus the standard error of the error rate. It enables to obtain a simpler tree and, in the same time, by preserving the generalization performance.

In this tutorial, we show to implement the CART approach into TANAGRA. We show also how to set the settings in order to control the tree size. We will study their influence on the generalization error rate.

2 Dataset

We use the ADULT_CART_DECISION_TREES.XLS[1] from the UCI Repository[2].

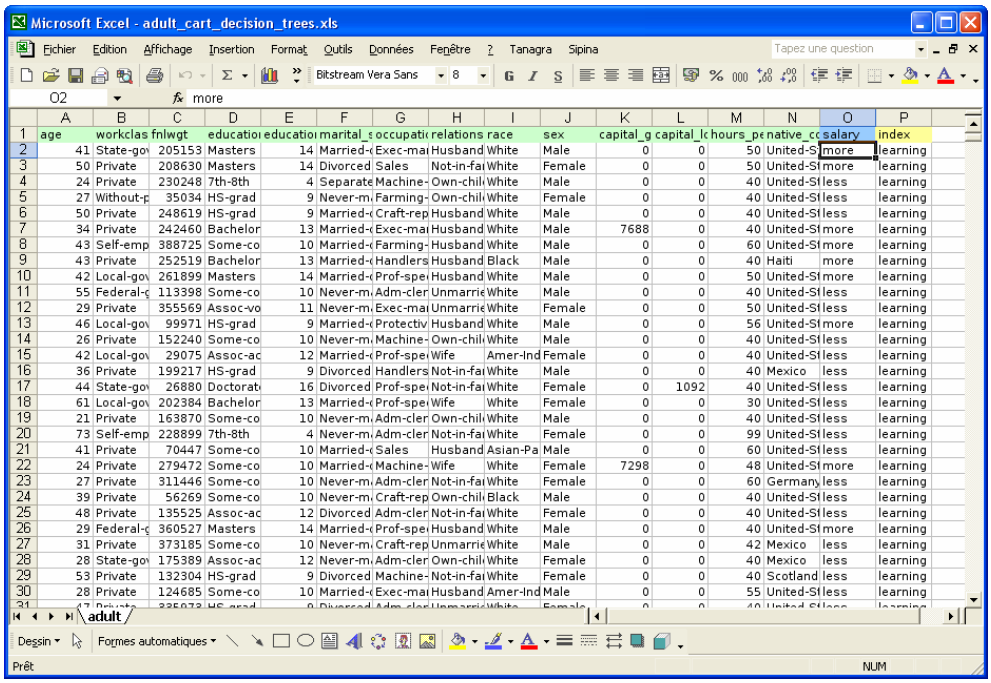
There are 48,842 instances and 14 variables. The target attribute is CLASS. We try to predict the salary of individuals (is the annual income is higher to 50,000$ or not) from their characteristics (age, education, etc.).

The training set size is 10,000. They are used for the construction of the tree. In the CART process, this dataset will be subdivided into growing and pruning set. The test set size is 38,842. They are only used for the evaluation of the generalization error rate. We note that this part of the dataset (the test set) is never used during the construction of the tree, neither for the growing phase, neither for the pruning phase. The INDEX column enables to specify the belonging of an instance to the train or the test set.

Our goal is to learn, based on the CART methodology, a decision tree that is both effective (with the lowest generalization error rate) and simple (with the fewest leaves - rules - as possible).

[1] Accessible en ligne : http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/adult_cart_decision_trees.zip

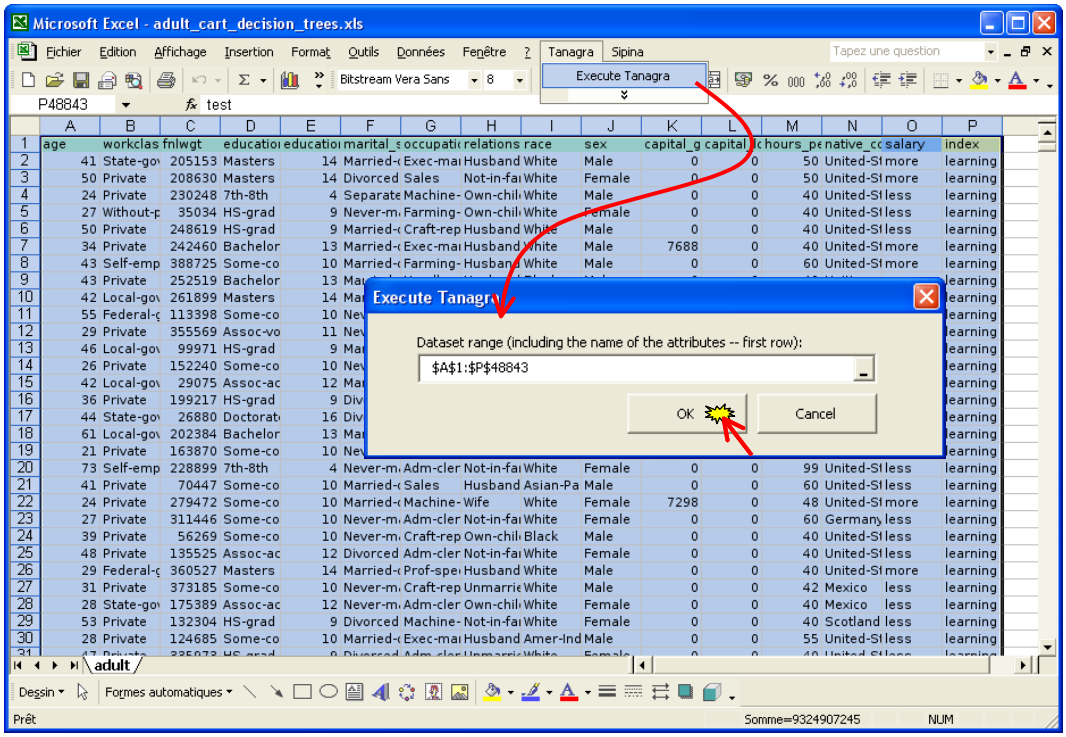
[2] http://archive.ics.uci.edu/ml/datasets/Adult



3 Learning a decision tree with the CART approach

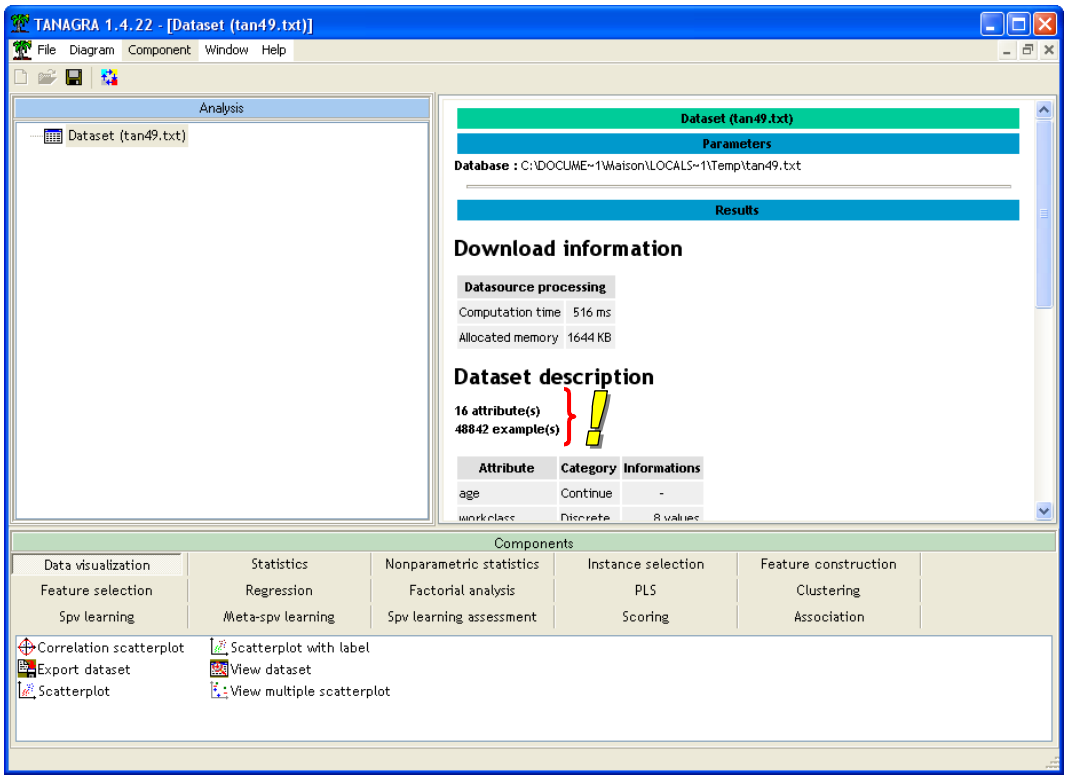
3.1 Importing the data file and creating a diagram

The simplest way to launch Tanagra is to open the data file into Excel. We select the data range; then we click on the Tanagra menu installed with the TANAGRA.XLA add-in[3]. After we checked the coordinates of the selected cells, we click on OK button.



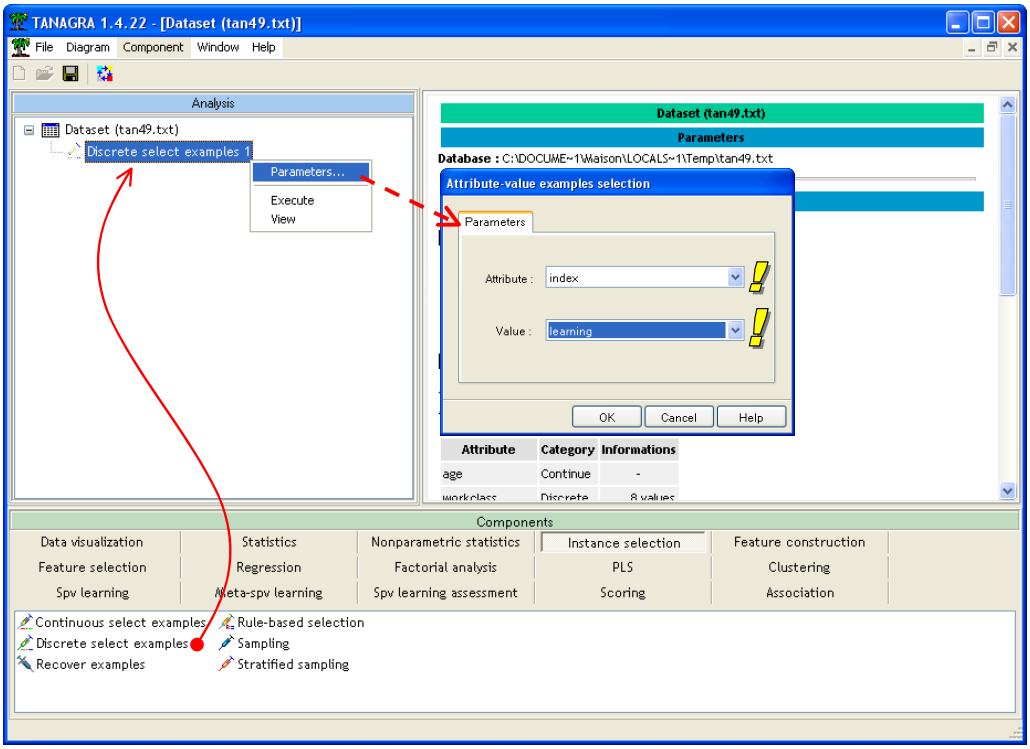
[3] See http://data-mining-tutorials.blogspot.com/2008/10/excel-file-handling-using-add-in.html

TANAGRA is automatically launched and the dataset imported. We have 48,842 instances and 15 columns (including the INDEX column).

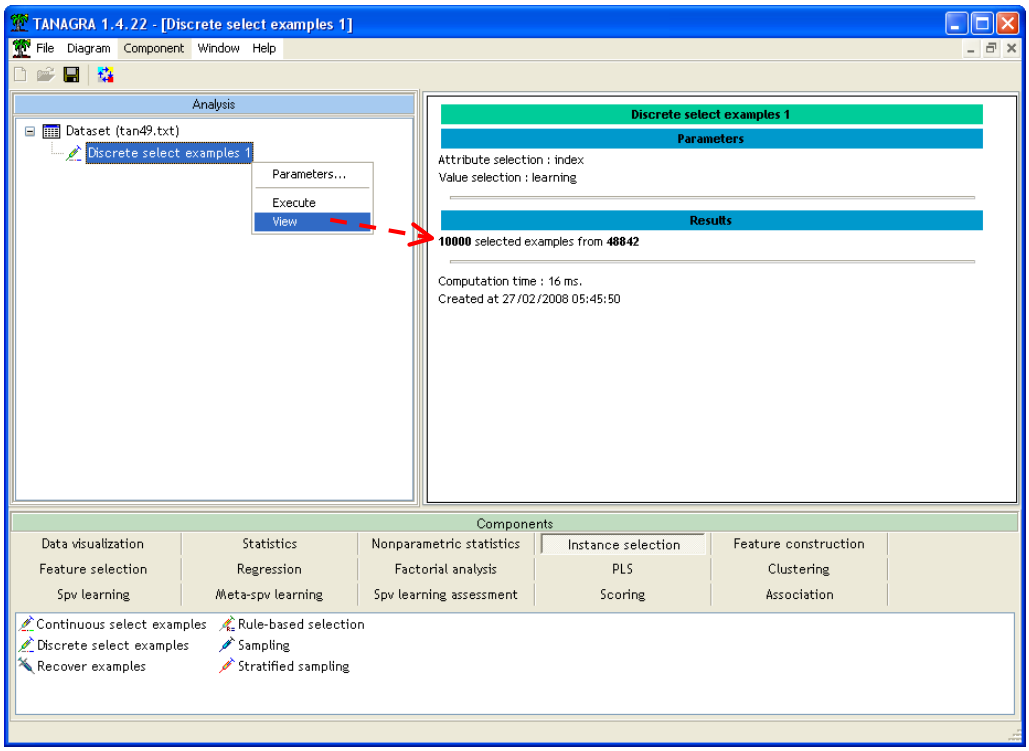


3.2 Specifying the train and the test sets

We add the DISCRETE SELECT EXAMPLES component (INSTANCE SELECTION tab). We click on the PARAMETERS menu. We set INDEX = LEARNING in order to select the train set.



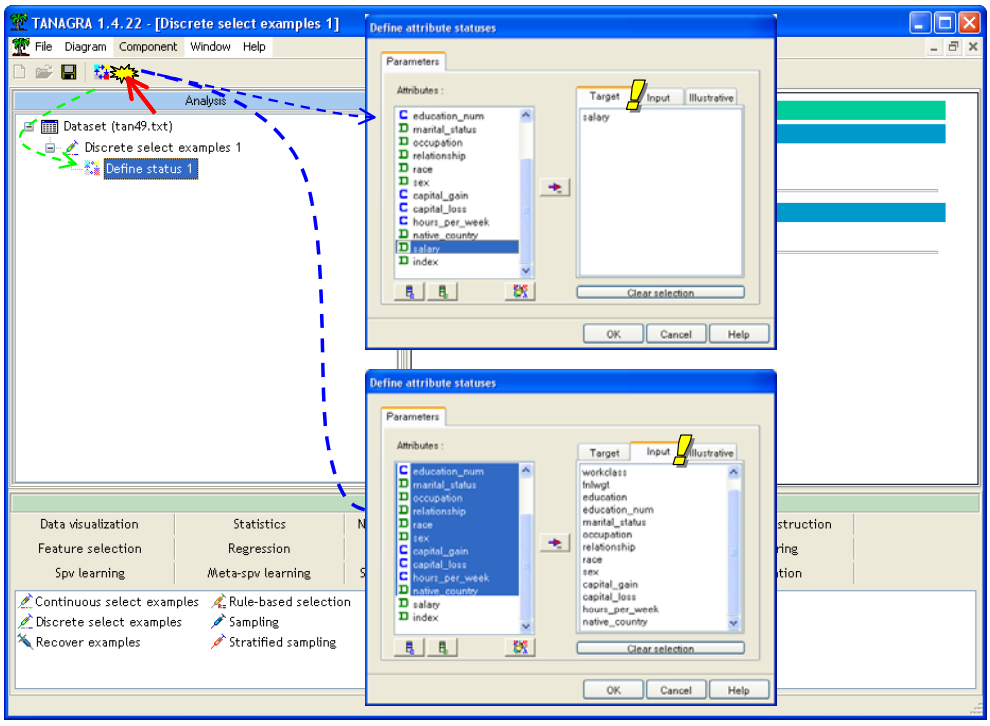
Then, we click on the VIEW menu: 10,000 examples are selected for the induction process.



3.3 Target variable and input variables

We want to specify the problem to analyze.

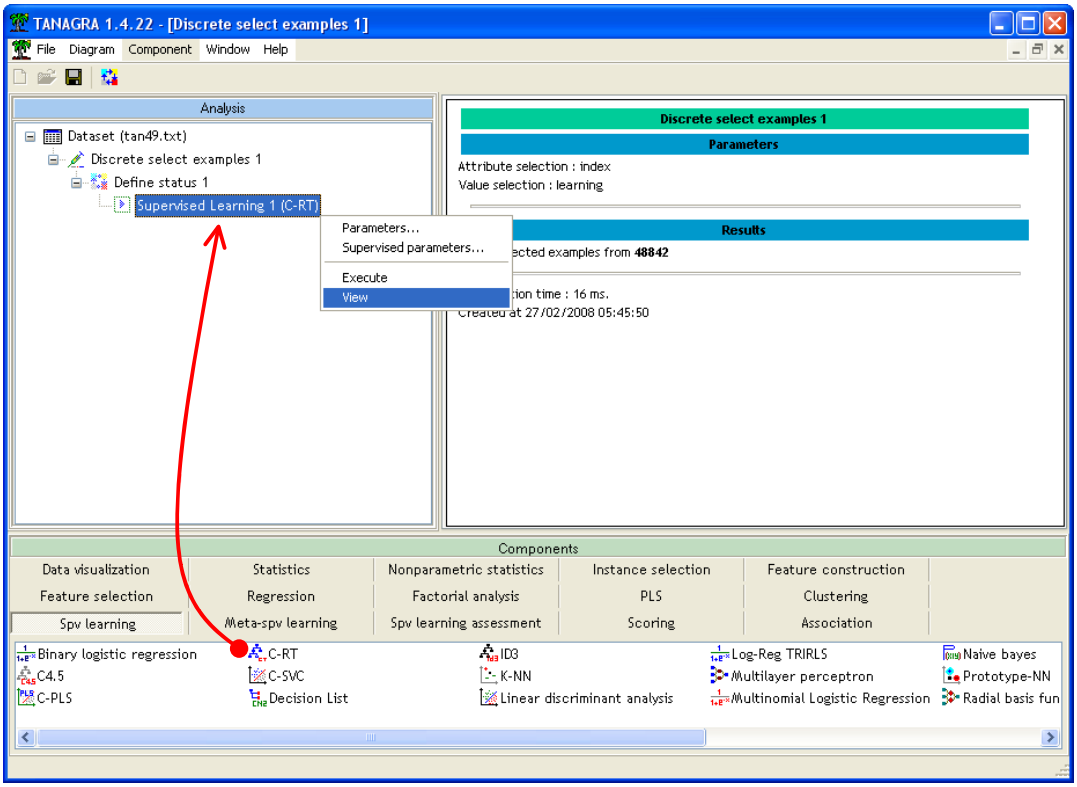
We add the DEFINE STATUS component into the diagram. We set CLASS as TARGET; all the other variables (except the INDEX column) as INPUT.



3.4 Learning a decision tree with the C-RT component

The C-RT component is an implementation of the CART algorithm, as it is described in the Breiman's book (Breiman and al., 1984). We use the GINI index as an indicator of goodness of split in the growing phase, and a separate pruning set is used in the post-pruning process.

We add the C-RT component (SPV LEARNING tab) into the diagram. We click on the VIEW menu.



Let us describe the various sections of the report supplied by Tanagra.

3.4.1 Confusion matrix

Classifier performances

Error rate	0.1490
Values prediction	**Confusion matrix**

Value	Recall	1-Precision
more	0.5474	0.2431
less	0.9453	0.1295

	more	less	Sum
more	1298	1073	2371
less	417	7212	7629
Sum	1715	8285	10000

The confusion matrix is computed on the whole training set (growing + pruning). On our dataset, the error rate is 14.9%. We know that because it is computed on the learning set, the resubstitution error rate is often (not always) optimistic.

3.4.2 Subdivision of the learning set into growing and pruning sets

Data partition

Growing set	6700
Pruning set	3300

Next, Tanagra displays the repartition of the learning set (10,000 instances) into growing (6,700) and pruning sets (3,300).

3.4.3 Trees sequence

Trees sequence (# 32)

N°	# Leaves	Err (growing set)	Err (pruning set)
32	1	0.2363	0.2388
28	6	0.1466	0.1539
20	39	0.1193	0.1479
1	205	0.0904	0.1700

The next table shows the candidate trees for the final model selection. For each tree, we have the number of leaves, the error rate on the growing set, and the error rate on the pruning set:

- The largest tree has 205 leaves, with an error rate of 9.04% on the growing set, and 17% on the pruning set.
- The optimal tree according to the pruning set contains 39 leaves, with an error rate of 14.79%.
- But, C-RT, based on the 1-SE principle, prefers the tree with 6 leaves with an error rate of 15.39% (on the pruning set). According the CART authors, this procedure enables to reduce dramatically the size of the selected tree (the initial tree contains 205 leaves!), without a diminution of the generalization performance. We will describe more deeply this approach below.

3.4.4 Tree description

Tree description

Number of nodes	11
Number of leaves	6

Decision tree

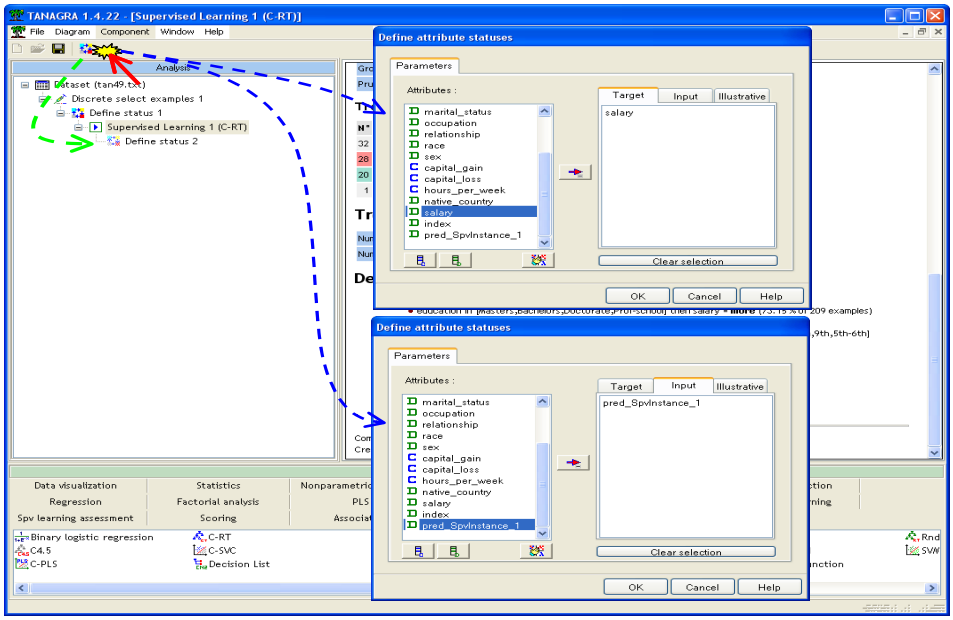
- relationship in [Husband,Wife]
  - education in [Masters,Bachelors,Doctorate,Prof-school] then salary = **more** (73.15 % of 209 examples)
  - education in [7th-8th,HS-grad,Some-college,Assoc-voc,Assoc-acdm,11th,10th,Preschool,12th,1st-4th,9th,5th-6th]
    - capital_gain < 5095.5000
      - capital_loss < 1794.0000 then salary = **less** (72.96 % of 1953 examples)
      - capital_loss >= 1794.0000 then salary = **more** (71.79 % of 78 examples)
    - capital_gain >= 5095.5000 then salary = **more** (96.43 % of 112 examples)
- relationship in [Not-in-family,Own-child,Unmarried,Other-relative]
  - capital_gain < 7073.5000 then salary = **less** (94.87 % of 3608 examples)
  - capital_gain >= 7073.5000 then salary = **more** (93.22 % of 59 examples)

The final section of the report describes the induced decision tree.

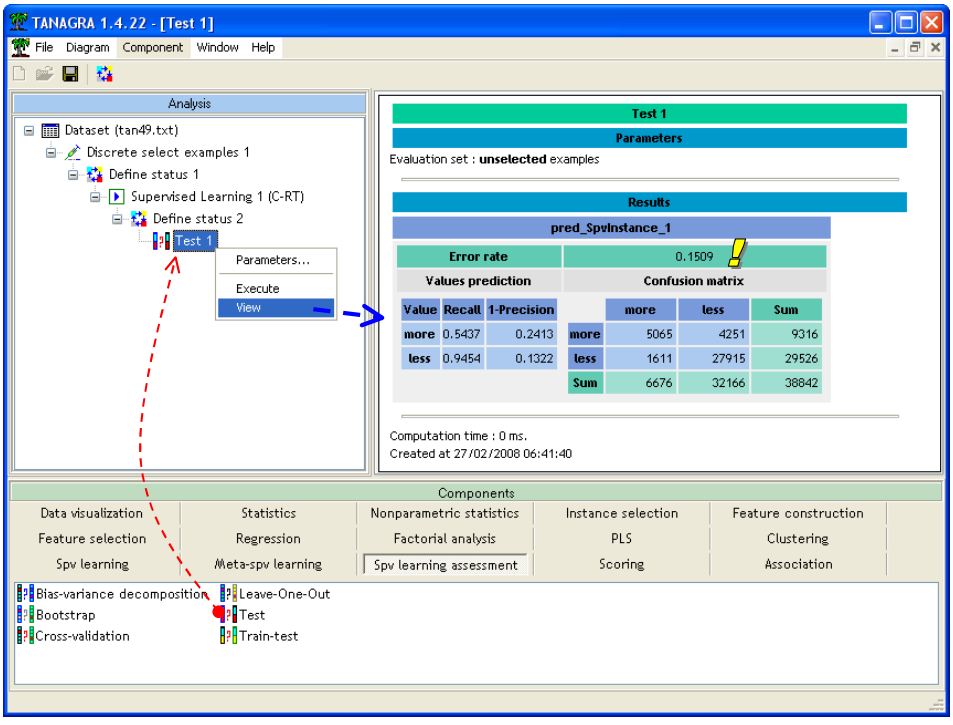
3.5 Evaluation on the test set

Both the growing and the pruning sets are used during the tree construction. They cannot give an honest estimate of the error rate. For this reason, we use a third part of the dataset for the model assessment: this is the test set.

We insert the DEFINE STATUS component into the diagram, we set SALARY as TARGET, and the predicted values computed from the decision tree (PRED_SPVINSTANCE_1) as INPUT.



Then, we add the TEST component (SPV LEARNING ASSESSMENT tab). By default, it computes the confusion matrix, and thus the error rate, on the previously unselected instances i.e. the test set.



We click on the VIEW menu. The test error rate is 15.09%, computed on 38,842 instances.

This is an estimated value of course. But it is rather reliable since it is computed on a large sample; the confidence interval of the error rate is [0.1473; 0.1545] for a 95% confidence level.

4 Some variants about the tree selection

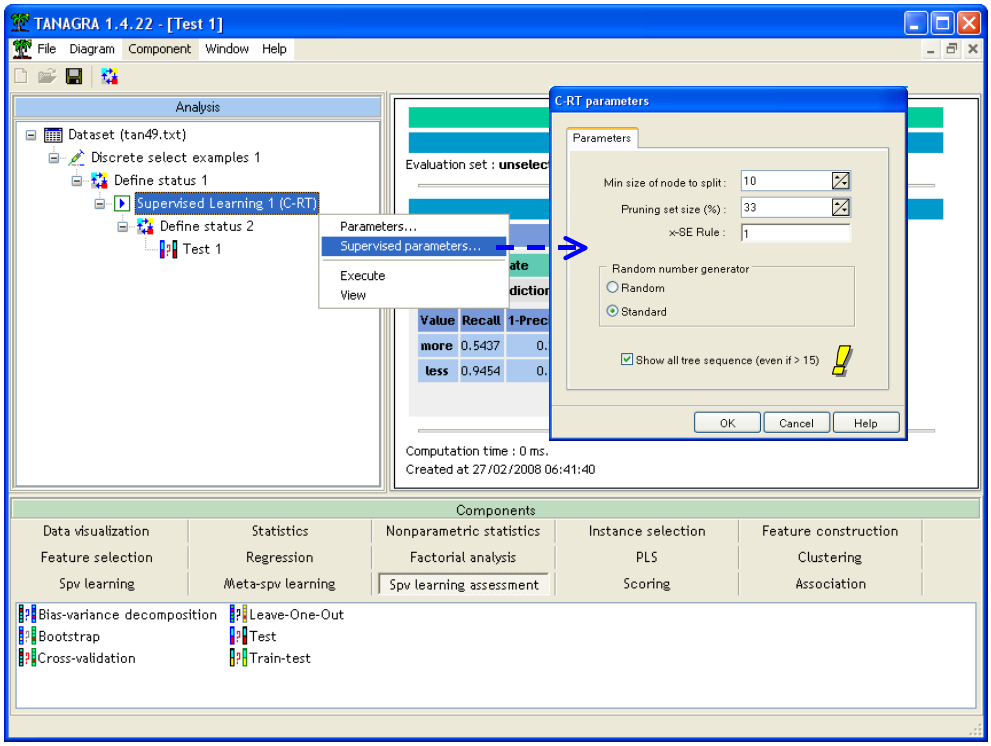
4.1 The x-SE RULE principle

Why do we not select the optimal tree on the pruning set?

The first reason is that we must not transfer the overfitting from the growing set to the pruning set. The second reason is that a deeper study of the error rate curve according to the tree size shows that we can select many solutions. It is more suitable to select the simplest tree for the deployment and the interpretation.

4.1.1 Error rate curve according to the tree size

To obtain the detailed values of the error rate according to the tree size, we click on the SUPERVISED PARAMETERS menu of the SUPERVISED LEARNING 1 (C-RT) component. We activate the SHOW ALL TREE SEQUENCE option.



We click on the VIEW menu. The detailed values of the error rate are given in the "Tree Sequence" table now (Tableau 1). We can obtain a graphical representation of these values (Figure 2).

We note that the tree with 6 leaves is very close, according the pruning error rate, to the optimal tree. The difference seems not significant.

N°	# Leaves	Err (growing set)	Err (pruning set)
32	1	0.2363	0.2388
31	3	0.1748	0.1806
30	4	0.1593	0.1664
29	5	0.1516	0.1567
28	6	0.1466	0.1539
27	7	0.1446	0.1497
26	11	0.1391	0.1488
25	16	0.1340	0.1488
24	21	0.1293	0.1485
23	22	0.1284	0.1488
22	28	0.1248	0.1485
21	33	0.1221	0.1494
20	39	0.1193	0.1479
19	43	0.1176	0.1479
18	46	0.1164	0.1506
17	51	0.1148	0.1521
16	73	0.1082	0.1552
15	80	0.1064	0.1558
14	83	0.1057	0.1555
13	91	0.1039	0.1552
12	94	0.1033	0.1552
11	107	0.1009	0.1567
10	116	0.0994	0.1570
9	139	0.0960	0.1606
8	154	0.0942	0.1642
7	158	0.0937	0.1642
6	168	0.0927	0.1648
5	174	0.0921	0.1670
4	188	0.0910	0.1679
3	194	0.0907	0.1691
2	199	0.0906	0.1700
1	205	0.0904	0.1700

Tableau 1 – Tree sequence description - Growing / pruning error rate

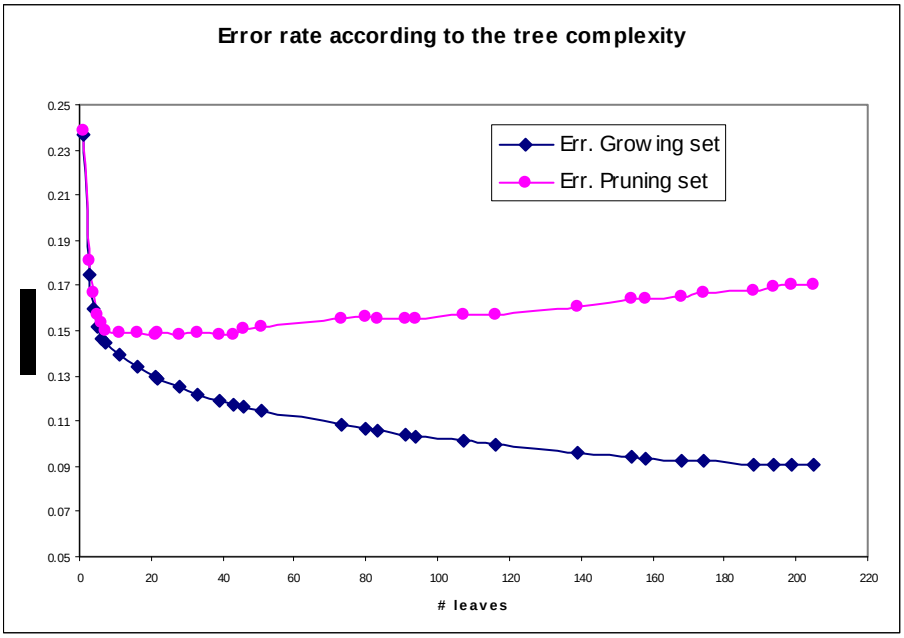


Figure 2 - Evolution of the error rate according to the tree size

4.1.2 The 1-SE RULE tree selection

How C-RT selects the tree with 6 leaves? The idea is to select the simplest tree for which the pruning error rate is not significantly higher than the one of optimal tree. For this, it computes a value which is similar to the higher limit of the confidence interval of the error rate of the optimal tree.

In our case, the optimal tree has 39 leaves, with an error rate of $\varepsilon = 0.1479$. The estimated standard error is

$$\sigma = \sqrt{\frac{\varepsilon \times (1-\varepsilon)}{n}} = \sqrt{\frac{0.1479 \times (1-0.1479)}{3300}} = 0.00617977$$

The upper limit defined by the 1-SE RULE ($\theta = 1$) is

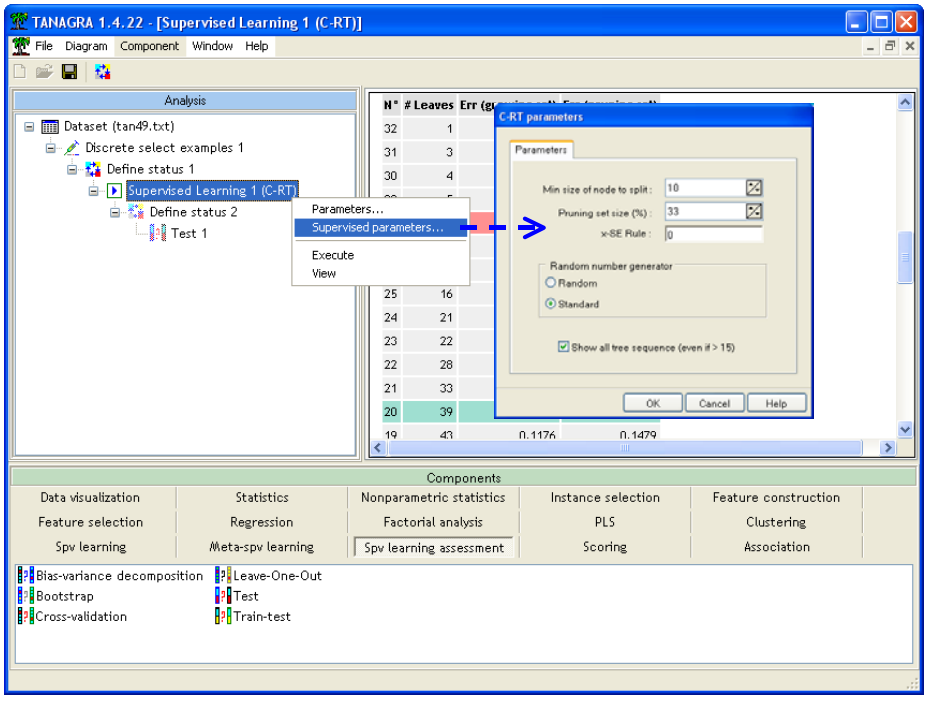
$$\varepsilon_{seuil} = \varepsilon + \theta \times \sigma = \varepsilon + 1 \times \sigma = 0.1541$$

Thus, we search in the table above the simplest tree for which the pruning error rate is not higher than this limit. It is the tree n°28 with 6 leaves; the pruning error rate is 0.1539.

4.1.3 Accuracy of the 0-SE RULE ($\theta = 0$) tree on the test set

We see above that the test error rate of the tree defined by the 1-SE rule is 15.07% (section 3.5). What about the performance of the optimal tree (with 39 leaves)? Is it better or worse?

We click on the SUPERVISED PARAMETERS menu of the SUPERVISED LEARNING 1 (C-RT). We specify the 0-SE RULE for the tree selection.



We click on the VIEW menu. Not surprisingly, the optimal tree is the selected tree (39 leaves).

Trees sequence (# 32)

N°	# Leaves	Err (growing set)	Err (pruning set)
32	1	0.2363	0.2388
31	3	0.1748	0.1806
30	4	0.1593	0.1664
29	5	0.1516	0.1567
28	6	0.1466	0.1539
27	7	0.1446	0.1497
26	11	0.1391	0.1488
25	16	0.1340	0.1488
24	21	0.1293	0.1485
23	22	0.1284	0.1488
22	28	0.1248	0.1485
21	33	0.1221	0.1494
20	39	0.1193	0.1479

The error rate on the whole learning set (growing + pruning) is 0.1287.

Classifier performances

Error rate	0.1287

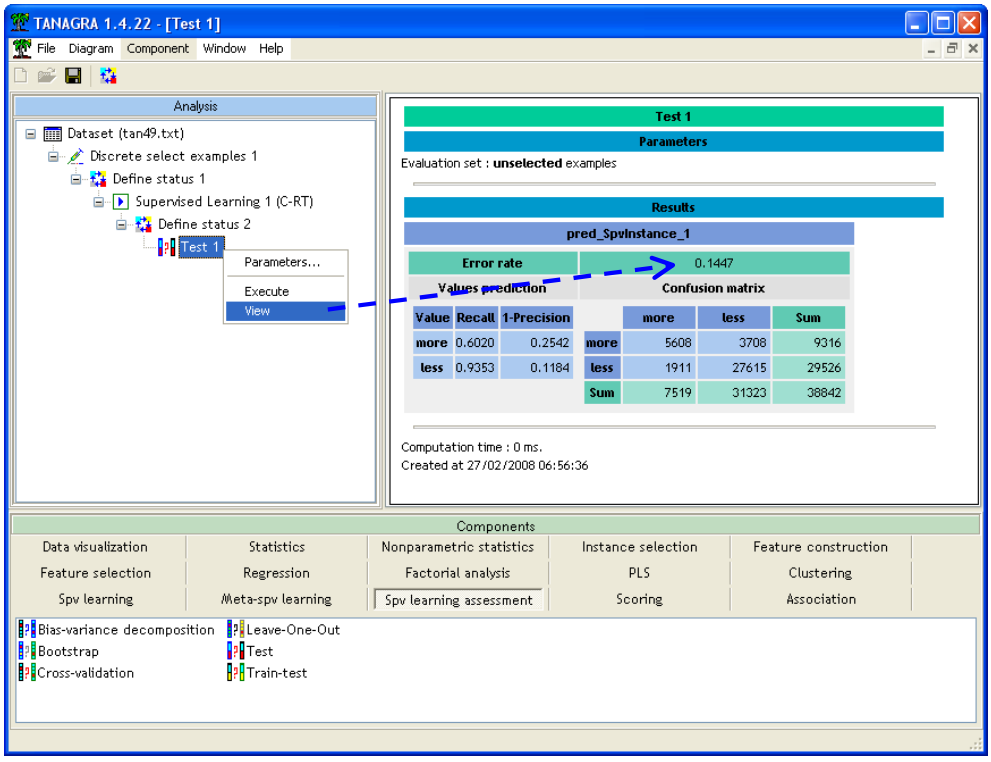
Values prediction

Value	Recall	1-Precision
more	0.6326	0.2171
less	0.9455	0.1077

Confusion matrix

	more	less	Sum
more	1500	871	2371
less	416	7213	7629
Sum	1916	8084	10000

To obtain the test error rate, we click on the VIEW menu of the TEST 1 component into the diagram.



The test error rate of the optimal tree (with 39 leaves) is 0.1447. Its confidence interval for the 95% confidence level is [0.1412; 0.1482]. This tree, which is much larger than the tree defined with the 1-SE rule principle (39 leaves vs. 6 leaves), is not significantly better (Section 3.5, page 8 – the confidence interval was [0.1473; 0.1545]).

4.2 Selection of a specific tree

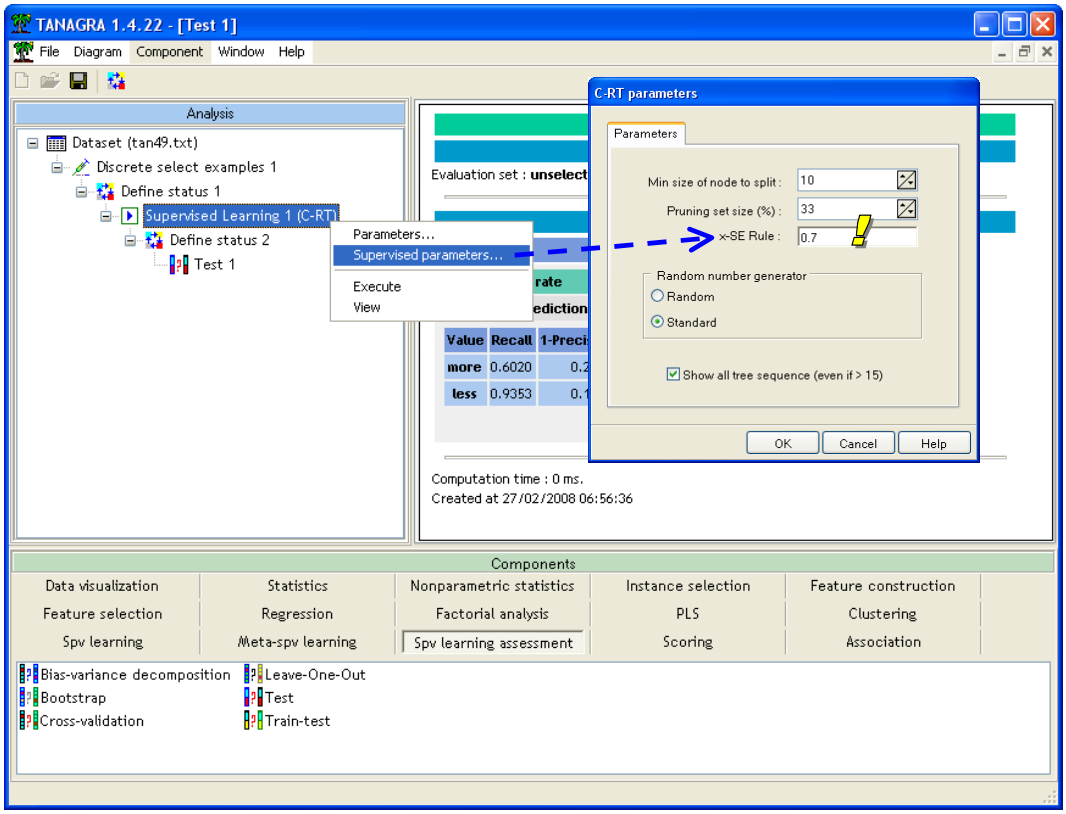
Another way to select the final tree is to use the error rate curve above (Figure 2). According to the error rate related to each candidate tree (Tableau 1) and our domain knowledge, we can set the appropriate value θ in order to obtain a specific tree.

4.2.1 Specifying the parameter θ

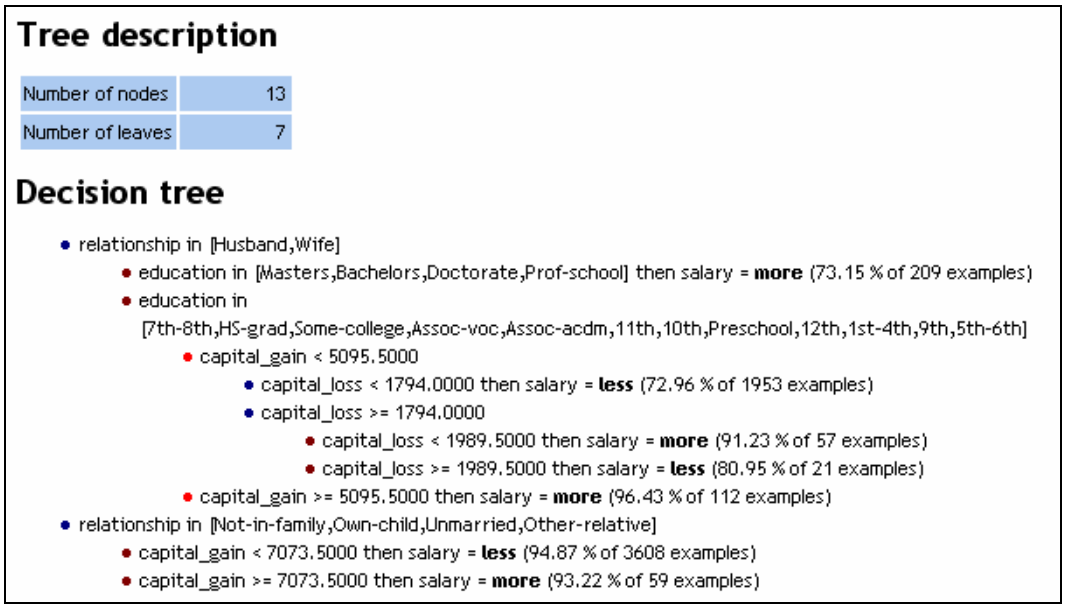
Given the error curve (Figure 2), we want to obtain the tree with 7 leaves (tree n°27). Its pruning error rate is 0.1497. To obtain this tree, we define the parameter theta θ so that the threshold lies between the tree with 7 leaves (pruning error rate = 0.1497) and the tree with 6 leaves (pruning error rate = 0.1539). Through trial and error, it appears that theta = 0.7 is a suitable value, the upper limit becomes

$$\varepsilon_{seuil} = 0.1479 + 0.7 \times 0.006 = 0.1522 .$$

We click on the SUPERVISED PARAMETERS of the SUPERVISED LEARNING 1 (C-RT) component, we set θ = 0.7.



The obtained contains actually 7 leaves.



Tree description

Number of nodes	13
Number of leaves	7

Decision tree

- relationship in [Husband,Wife]
  - education in [Masters,Bachelors,Doctorate,Prof-school] then salary = **more** (73.15 % of 209 examples)
  - education in [7th-8th,HS-grad,Some-college,Assoc-voc,Assoc-acdm,11th,10th,Preschool,12th,1st-4th,9th,5th-6th]
    - capital_gain < 5095.5000
      - capital_loss < 1794.0000 then salary = **less** (72.96 % of 1953 examples)
      - capital_loss >= 1794.0000
        - capital_loss < 1989.5000 then salary = **more** (91.23 % of 57 examples)
        - capital_loss >= 1989.5000 then salary = **less** (80.95 % of 21 examples)
    - capital_gain >= 5095.5000 then salary = **more** (96.43 % of 112 examples)
- relationship in [Not-in-family,Own-child,Unmarried,Other-relative]
  - capital_gain < 7073.5000 then salary = **less** (94.87 % of 3608 examples)
  - capital_gain >= 7073.5000 then salary = **more** (93.22 % of 59 examples)

Note: According to the tools, we can handle another parameter than theta (e.g. the complexity parameter for R software, "rpart" package). But, in all cases, the goal is to select the "suitable" tree from the error rate curve.

4.2.2 Generalization performance of the tree with θ = 0.7

Last, we want to evaluate this tree on the test set. We click on VIEW menu of TEST 1. We obtain 0.1489. Its confidence interval at the 95% confidence level is [0.1454; 0.1524].

The following table summarizes the various evaluated configurations.

Theta-SE RULE	#Leaves	Err.Test	95% Conf.Interval
1	6	0.1509	0.1473 ; 0.1545
0.7	7	0.1489	0.1454 ; 0.1524
0	39	0.1447	0.1412 ; 0.1482

Clearly, the tree with 6 leaves (θ = 1) is enough to get a sufficient level of performance.

5 Conclusion

Among the many variants of decision trees learning algorithms, CART is probably the one that detects better the right size of the tree.

In this tutorial, we describe the selection mechanism used by CART during the post-pruning process. We show also how to set the appropriate value of the parameter of the algorithm in order to obtain a specific (a user-defined) tree.