Subject

Show how to use Decision Lists (DL) with TANAGRA.

Decision Lists have been popular methods in 90's in machine learning scientific publications. They produce a list of sorted production rules such as "**IF** condition_1 **THEN** conclusion_1 **ELSE IF** condition_2 **THEN** condition_2 **ELSE IF**…".

Decision Lists and Decision Trees have a similar representation bias but not the same learning bias, DL use the "separate-and-conquer" principle instead of "divide-and-conquer" principle. They can produce more specialized rules but they can also lead to overfitting: setting the right learning parameters is very important for the decision lists algorithm.

The algorithm that we have implemented in TANAGRA is suggested by CN2 (Clark & Niblett, ML-1989). We have introduced two main modifications: (1) we use a hill-climbing algorithm instead of a best-first search; (2) a new parameter, minimal support of a rule, can be adjusted to avoid non-significant rules.

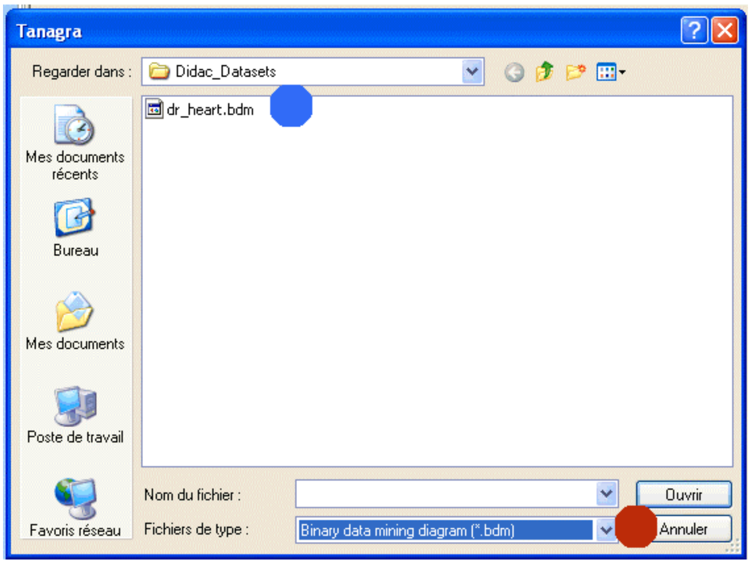
Dataset

HEART dataset; detect heart-disease from patient's characteristics.

Decision Lists

Download dataset

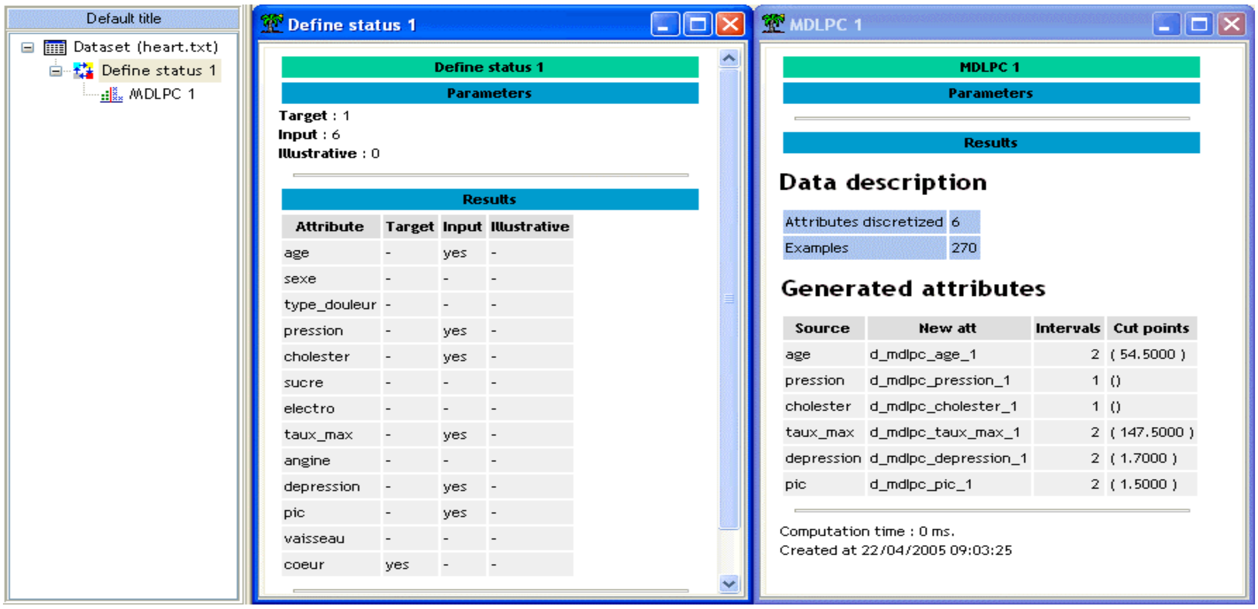
First of all, download the DR_HEART.BDM dataset (File / Open).



Discretization of continuous features

DL cannot handle continuous input attributes, we must discretize them, for instance with MDLP supervised algorithm.

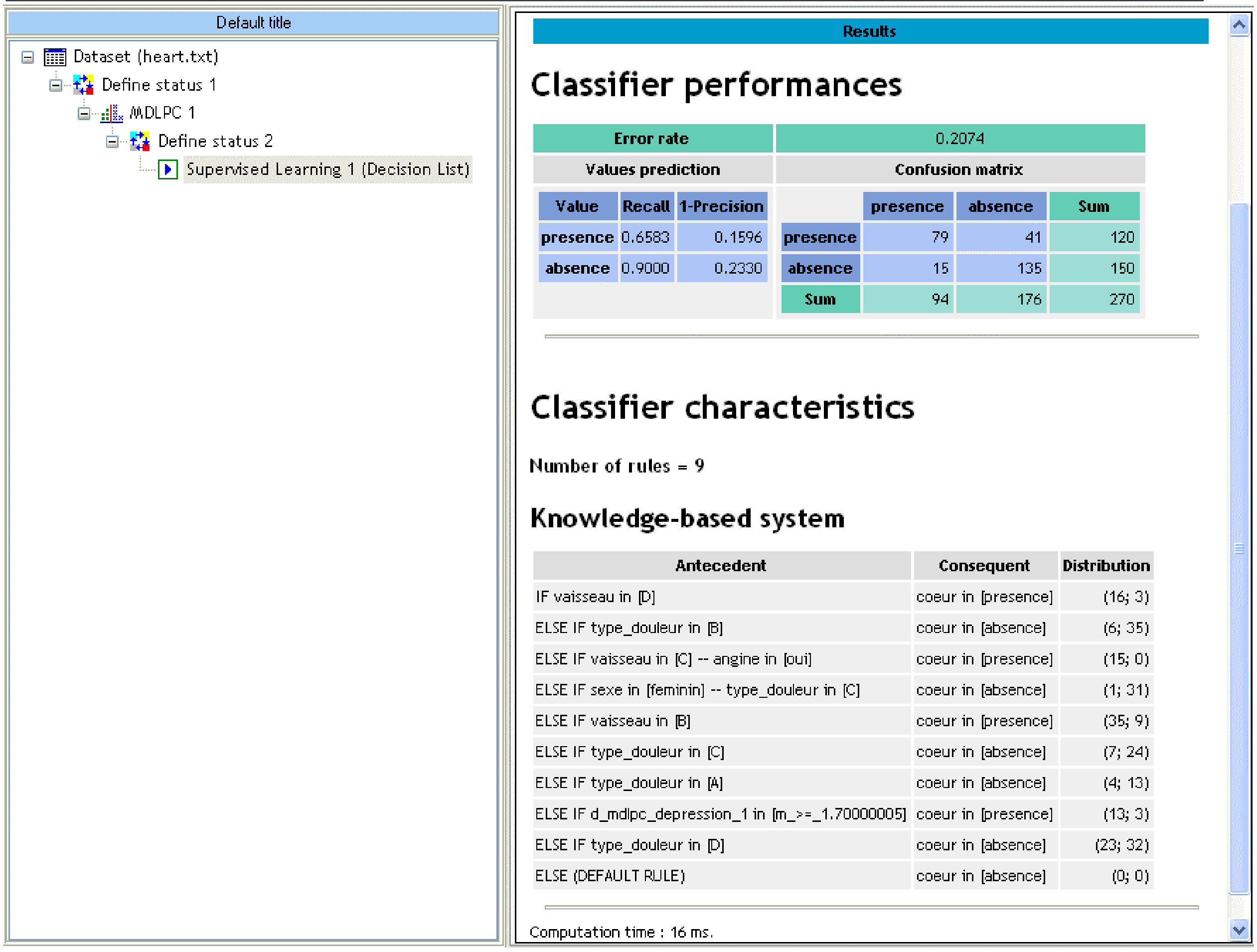
Set as INPUT all continuous attributes, and set COEUR as TARGET. Add the MDLPC component in the stream diagram.



Learning DL

Set as INPUT all discrete attributes (including the discretized attributes) except COEUR, which is the TARGET attribute.

Add learning components, DL produces a set of rules, resubstitution error rate is 20% and we obtain 9 rules.



Classifier performances

Error rate	0.2074
Values prediction	Confusion matrix

Value	Recall	1-Precision
presence	0.6583	0.1596
absence	0.9000	0.2330

	presence	absence	Sum
presence	79	41	120
absence	15	135	150
Sum	94	176	270

Classifier characteristics

Number of rules = 9

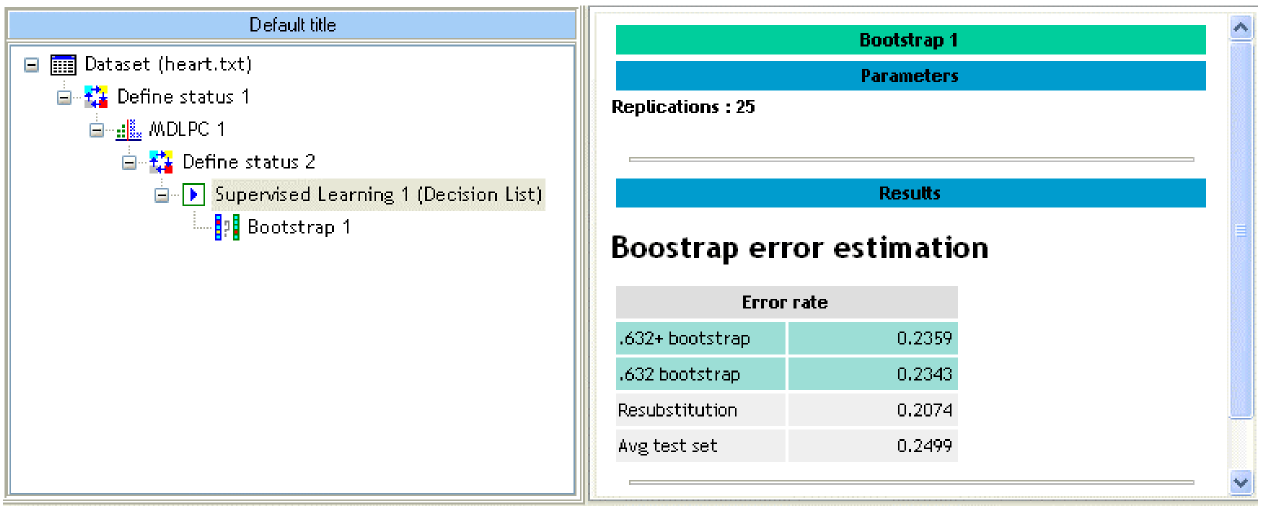
Knowledge-based system

Antecedent	Consequent	Distribution
IF vaisseau in [D]	coeur in [presence]	(16; 3)
ELSE IF type_douleur in [B]	coeur in [absence]	(6; 35)
ELSE IF vaisseau in [C] -- angine in [oui]	coeur in [presence]	(15; 0)
ELSE IF sexe in [feminin] -- type_douleur in [C]	coeur in [absence]	(1; 31)
ELSE IF vaisseau in [B]	coeur in [presence]	(35; 9)
ELSE IF type_douleur in [C]	coeur in [absence]	(7; 24)
ELSE IF type_douleur in [A]	coeur in [absence]	(4; 13)
ELSE IF d_mdlpc_depression_1 in [m_>=_1.70000005]	coeur in [presence]	(13; 3)
ELSE IF type_douleur in [D]	coeur in [absence]	(23; 32)
ELSE (DEFAULT RULE)	coeur in [absence]	(0; 0)

Computation time : 16 ms.

Error rate evaluation

To obtain less biased error rate estimation, we use the BOOSTRAP component; the "true" error rate is 24%

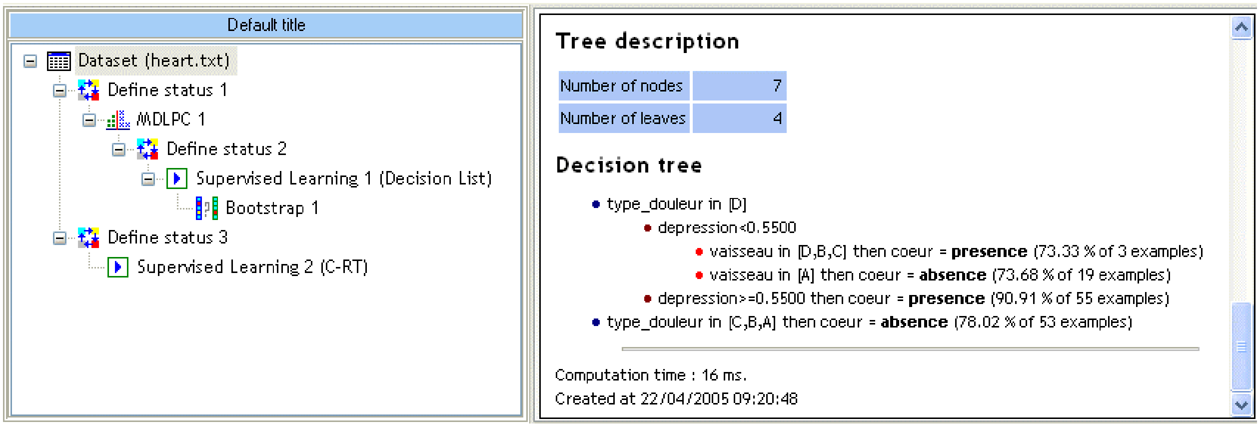


Boostrap error estimation

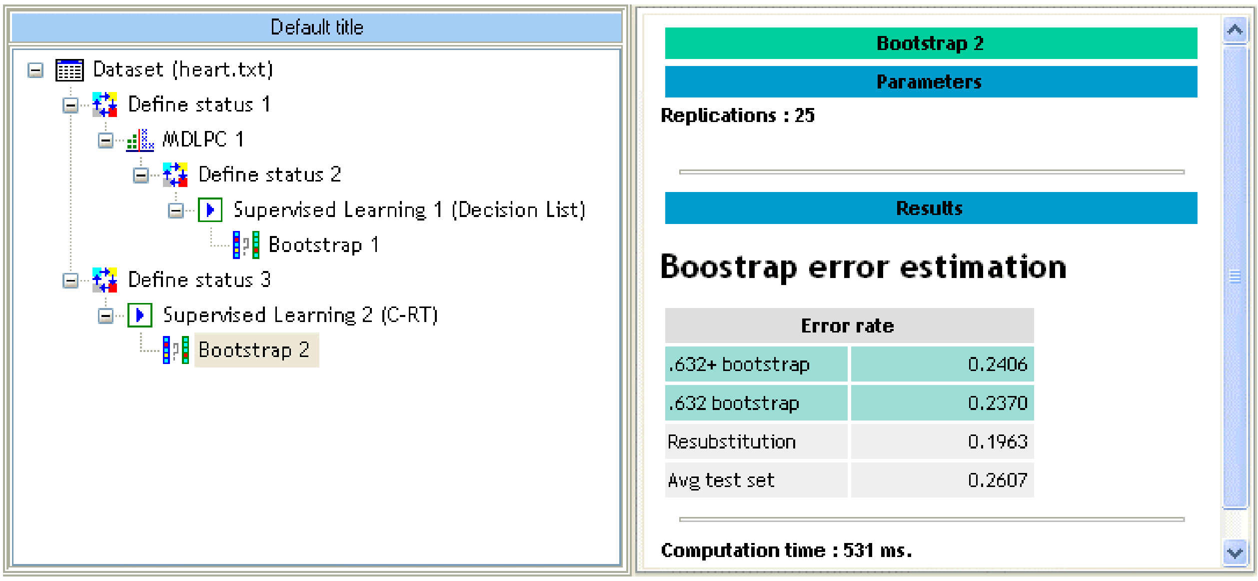
Error rate	
.632+ bootstrap	0.2359
.632 bootstrap	0.2343
Resubstitution	0.2074
Avg test set	0.2499

Comparison with C-RT decision tree algorithm

We want to compare these results with a popular decision tree algorithm (C-RT, Breiman & al., 1984). Because this method can handle both continuous and discrete input attributes, we do not need to discretize the continuous attributes before learning.



The estimated error rate with the BOOTSTRAP is 24%.



The majority of comparisons on benchmark datasets show that these methods -- "Decision Lists" and "Decision Trees" -- lead to very similar performances.

I think the Decision Lists are less known of general public essentially because they have never been implemented in popular data mining software.