Cost Sensitive Learning

Take misclassification costs into account during the construction and the evaluation of predictive models

Ricco RAKOTOMALALA

Outline

1. Cost sensitive learning: key issues
2. Evaluation of the classifiers
3. An example: CHURN dataset
4. Method 1: ignore the costs
5. Method 2: modify the assignment rule
6. Method 3: embed the costs in the learning algorithm
7. Other methods: Bagging and MetaCost
8. Conclusion
9. References

Cost sensitive learning
Key issues

Misclassification costs are inherent in the classification process

The goal of supervised learning is to build a model (a classification function) which connects Y, the target attribute, with (X1, X2,..),the input attributes. We want that the model is the most effective as possible.

$$Y = f(X_1, X_2, \ldots, X_J; \theta)$$

To quantify "the most effective as possible", we measure often the performance with the error rate. It corresponds to the probability of misclassification of the model.

$$ET = \frac{1}{card(\Omega)} \sum_{\Omega} \Delta[Y, \hat{f}(X, \hat{\alpha})]$$

$$où\ \Delta[.] = \begin{cases} 1\, si\, Y \neq \hat{f}(X, \hat{\alpha}) \\ 0\, si\, Y = \hat{f}(X, \hat{\alpha}) \end{cases}$$

But the error rate gives the same importance to all types of error. Yet, some types of misclassification may be worse than others. E.g. (1) Designate as "sick" a "healthy" person does not imply the same consequences than to designate as "healthy" a somebody who is ill. (2) Accuse of fraud an innocent person has not the same consequence than to neglect a fraudster.

This analysis is all the more important that the positive instances - positive class membership - that we want to detect are generally rare in the population (the ill persons are not many, the fraudsters are rare, etc.)

The misclassification cost matrix

(1) How to express the consequences of bad assignments?

→ *We use the misclassification cost matrix*

Case of the binary classification problem (K = 2) – *The most common*

$Y \backslash \hat{Y}$	$\hat{+}$	$\hat{-}$
$+$	α	β
$-$	γ	δ

Notes:

- Usually $\alpha=\beta=0$; but not always, sometimes $\alpha,\beta < 0$, the cost is negative i.e. a gain (e.g. give a credit to a reliable client)
- If $\alpha=\beta=0$ and $\gamma=\delta=1$, we have the usual scheme where the expected cost of misclassifications is equivalent to the error rate

(2) How to use the misclassification cost matrix for the evaluation of the classifiers?

→ *The starting point is always the confusion matrix*

→ *But we must combine this one with the misclassification cost matrix*

(3) How to use the cost matrix for the construction of the classifier?

→ *The base classifier is the one built without consideration of cost matrix*

→ *We must do better i.e. to obtain a better evaluation of the classifier by considering the misclassification cost*

Classifier evaluation metric:
the expected cost of misclassification (ECM)

The expected cost of misclassification (ECM)

$Y \backslash \hat{Y}$	$\hat{+}$	$\hat{-}$
$+$	a	b
$-$	c	d

The confusion matrix points out the quantity and the structure of the error i.e. the nature of the misclassification

$Y \backslash \hat{Y}$	$\hat{+}$	$\hat{-}$
$+$	α	β
$-$	γ	δ

The misclassification cost matrix quantifies the cost which is associated to each type of error

The expected cost of misclassification of the model (M)

$$C(M) = \frac{1}{n}\left(a \times \alpha + b \times \beta + c \times \gamma + d \times \delta\right)$$

We will use this metric to evaluate and compare the learning strategies.

Comments:

Its interpretation is not easy (unit of the cost?)...

Anyway, it allows to compare the performance of models

The lower is the ECM, the better is the model

The calculation must be performed on a test sample (or using resampling approaches such as cross-validation,...)

ECM – An example of calculation for comparing models

(M1)

		Prédite		
		+	-	Total
Observée	+	40	10	50
	-	20	30	50
	Total	60	40	100

$$C(M1) = \frac{1}{100}\left(40 \times (-1) + 10 \times 10 + 20 \times 5 + 30 \times 0\right) = 1.6$$

$Y \backslash \hat{Y}$	$\hat{+}$	$\hat{-}$
$+$	-1	10
$-$	5	0

- *The error rates are the same (ε = 30%)*
- *But when we take into account the costs, we observe that M1 is better than M2*
- *It is quite normal, M2 is wrong where this is the most costly (the number of false negative is 30)*

(M2)

		Prédite		
		+	-	Total
Observée	+	20	30	50
	-	0	50	50
	Total	20	80	100

$$C(M2) = \frac{1}{100}\left(20 \times (-1) + 30 \times 10 + 0 \times 5 + 50 \times 0\right) = 2.8$$

The error rate is a particular case of the ECM

The error rate is the ECM for which the misclassification cost matrix is the identity matrix.

Y \ $\hat{Y}$	$\hat{+}$	$\hat{-}$
+	0	1
−	1	0

		Prédite		
		+	-	Total
Observée	+	40	10	50
	-	20	30	50
	Total	60	40	100

$$C(M) = \frac{1}{100}(40\times 0 + 10\times 1 + 20\times 0 + 30\times 1) = 0.3$$
$$= \frac{20+10}{100} = 0.3$$

There is therefore two implicit assumptions in the error rate: all kind of errors have the same cost, which is equal to 1; a good classification does not produce a gain (negative cost)

Dealing with the multiclass problem (K > 2)

When K > 2, the expected cost of misclassification becomes

The element of the confusion matrix (n_{ik}) : Number of instance which are predicted as y_k, and which are in fact membership of y_i, where $\sum_i \sum_k n_{ik} = n$

The element of the misclassification cost matrix (c_{ik}) : The cost when we assign the value y_k to an individual which belongs to the class y_i

The expected cost of misclassification for the model M

$$C(M) = \frac{1}{n} \sum_i \sum_k n_{ik} \times c_{ik}$$

An example: customer churn

Preventing the loss of customers

Domain : Telephony sector

Goal : Detecting the clients which may leave the company

Target attribute: CHURN – o (yes : **+**) / n (no : **-**)

Input attributes : the customer behavior and use of the various services offered

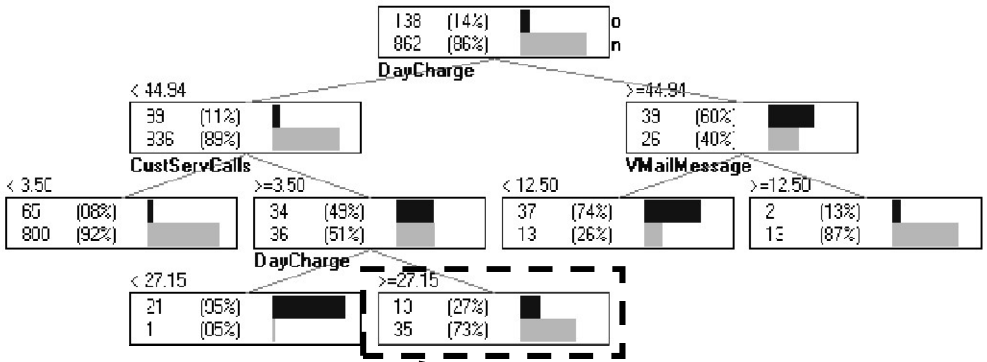
Samples: 1000 instances for the learning sample; 2333 for the test sample

Cost matrix

(*We can try different possibilities in practice*)

$Y \backslash \hat{Y}$	$\hat{+}$	$\hat{-}$
+	– 15	10
−	2	0

Decision tree learned from the dataset (*among the possible solutions*)



We focused on this leaf. We calculate the posterior class probabilities P(Y/X).

$$P(Y=+/\,DC<44.94; CSC\geq 3.5; DC\geq 27.15)=\frac{13}{48}=0.27$$

$$P(Y=-/\,DC<44.94; CSC\geq 3.5; DC\geq 27.15)=\frac{35}{48}=0.73$$

Method 1: ignore the costs

Method 1: neglect the misclassification cost matrix

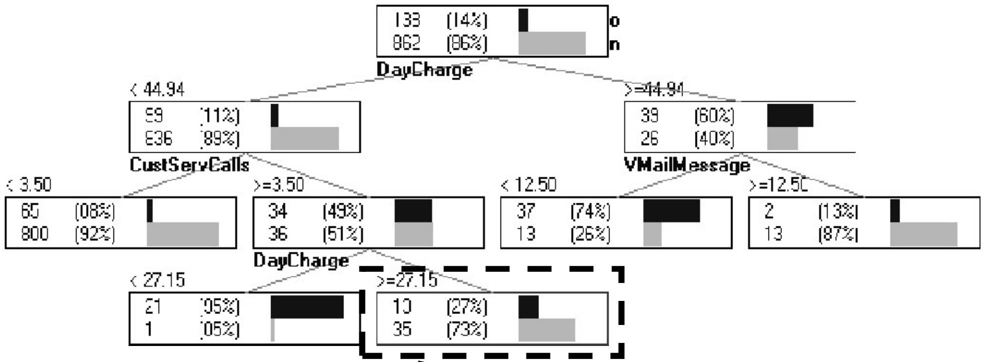
Method 1 :

- Neglect the misclassification costs during the construction of the classifier
- Neglect the misclassification costs when we assign the class to the individuals

i.e. we hope that the classifier which minimizes the error rate will minimize also the ECM

The assignment rule is based on the maximum of posterior class probabilities

$$y_{k*} = \arg\max_k P(Y = y_k / X)$$



$$P(Y = + / DC < 44.94; CSC \geq 3.5; DC \geq 27.15) = \frac{13}{48} = 0.27$$

$$P(Y = - / DC < 44.94; CSC \geq 3.5; DC \geq 27.15) = \frac{35}{48} = 0.73$$

If this rule is triggered when we try to assign a class to a new individual, then

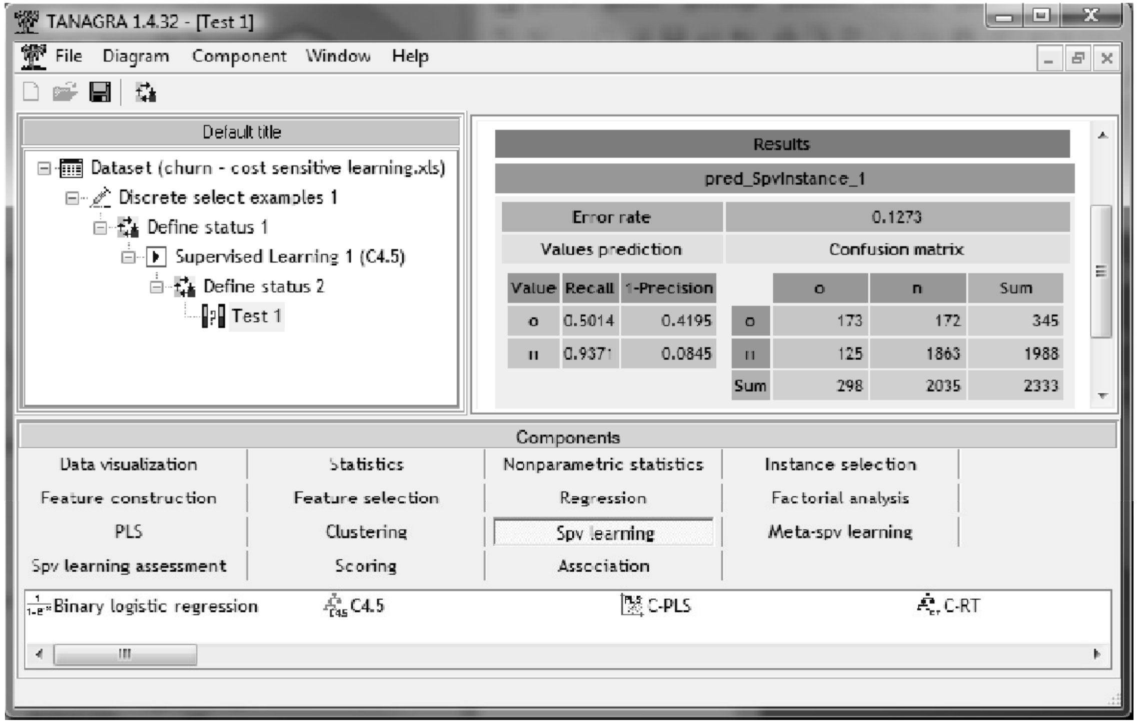
$$\hat{Y} = no$$

We predict "churn = no"

Method 1 : “CHURN” problem

1000 instances: training sample

2333 instances: test sample



Misclassification

cost matrix

Y \ $\hat{Y}$	$\hat{+}$	$\hat{-}$
+	– 15	10
–	2	0

$$C(M_1) = \frac{1}{2333}\left(-15\times 173 + 10\times 172 + 2\times 125 + 0\times 1863\right)$$
$$= -0.2679$$

This is the reference score i.e. by incorporating the cost in one way or another into the learning strategy, we must do better.

Method 2: modify the assignment rule

Method 2: modify the assignment rule

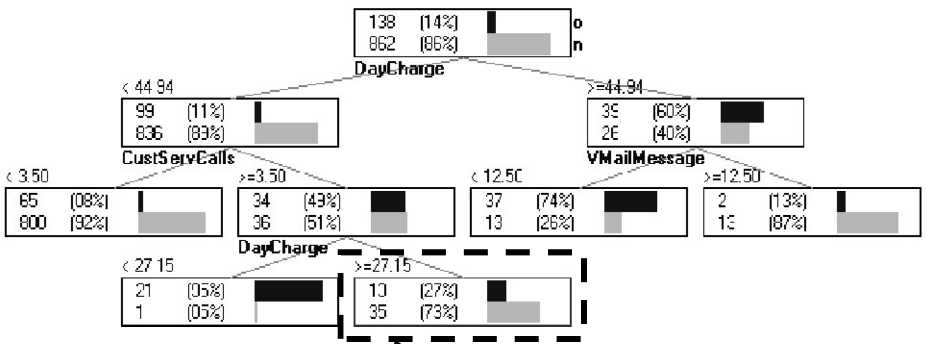
Method 2:

- Neglect the misclassification costs during the construction of the classifier
- Use the misclassification cost and the posterior class probabilities for the prediction

Rule: select the label which minimizes the expected cost

We calculate the expected cost for the prediction of each label. We select the one which minimizes the cost.

$$y_{k^*} = \arg\max_k C(y_k / X) = \arg\max_k \left[\sum_i P(Y = y_i / X) \times c_{ik} \right]$$



$$P(Y = + / X) = 0.27$$
$$P(Y = - / X) = 0.73$$

Misclassification cost matrix

Y \ $\hat{Y}$	$\hat{+}$	$\hat{-}$
+	-15	10
−	2	0

Expected cost for the prediction: Y = +

$$C(+/X) = -15 \times 0.27 + 2 \times 0.73 = -2.59$$

Expected cost for the prediction: Y = -

$$C(-/X) = 10 \times 0.27 + 0 \times 0.73 = 2.7$$

The least costly prediction is **Y = +**

Yet, this is not the label with the maximum posterior probability.

Method 2: some comments

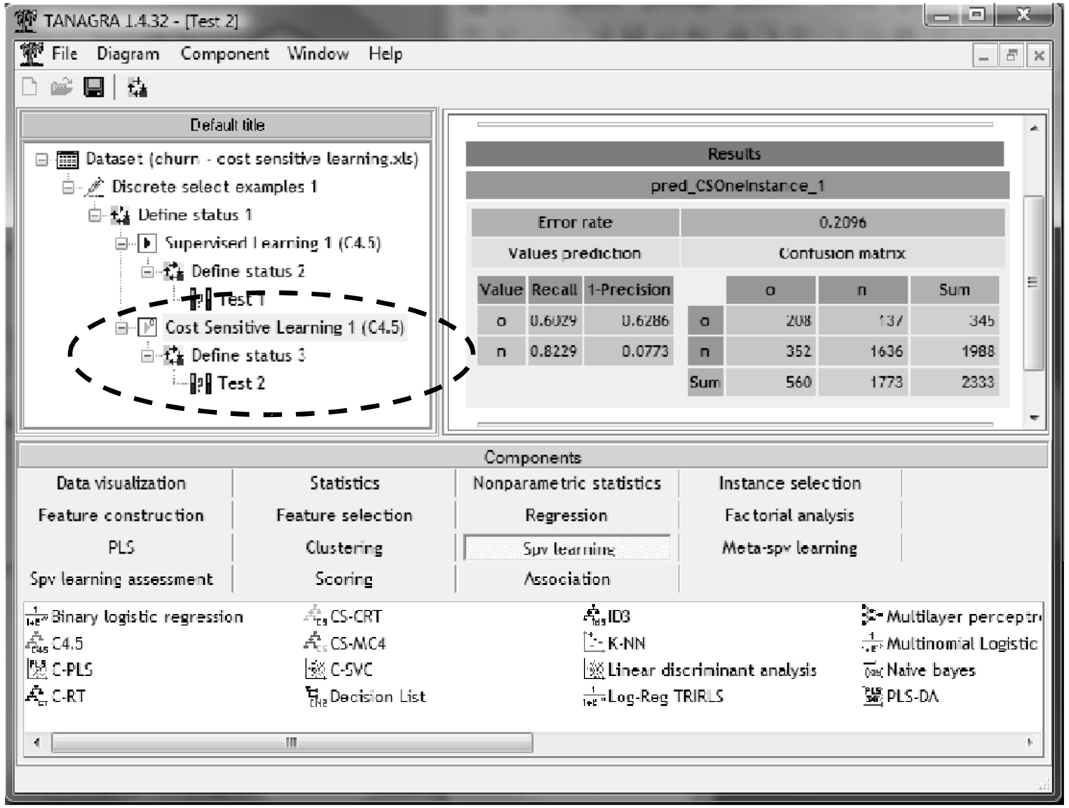
(1) This strategy is adaptable to any supervised learning algorithm (logistic regression, discriminant analysis, etc.) as long as it provides a reliable estimate of posterior class probabilities P(Y/X)

Exercise: See the detail of calculations for the logistic regression for instance

(2) When the cost matrix is the identity matrix, this strategy minimizes the error rate: this is a “real” generalization

Exercise : Apply the assignment rule with an identity matrix to the previous example

Method 2: “CHURN” problem



This is exactly the same tree as before. Only the assignment rule on the leaves was modified in order to take into account the cost matrix.

Y \ $\hat{Y}$	+	−
+	− 15	10
−	2	0

$$C(M_2) = \frac{1}{2333}\left(-15\times 208 + 10\times 137 + 2\times 352 + 0\times 1636\right)$$
$$= -0.4483$$

The improvement is dramatic, without a modification of the classifier!

Method 2 : “CHURN” problem - Comparison of the confusion matrices

Method 1

0.1273			
Confusion matrix			
	o	n	Sum
o	173	172	345
n	125	1863	1988
Sum	298	2035	2333

Method 2

0.2096			
Confusion matrix			
	o	n	Sum
o	208	137	345
n	352	1636	1988
Sum	560	1773	2333

Cost matrix

$Y \backslash \hat{Y}$	$\hat{+}$	$\hat{-}$
+	− 15	10
−	2	0

- The error rate is worse for M2. This result is expected because M2 does not try to minimize this metric.

- The number of true positive (TP) is higher for M2 (208 vs. 173 for M1) because this is the most advantageous situation (cost = -15)
- M2 has more false positive (FP) (352 vs.125 for M1) which are comparatively less penalizing (cost = 2)
- Since we increase the number of true positive, we have mechanically less false negative (FN = 137 vs. 172 for M1) (cost = 10). Therefore, the expected misclassification cost is lesser.

Method 3: embed the cost matrix within the learning algorithm

Method 3: modification of the learning algorithm

Method 3 :

• Use explicitly the cost matrix during the construction of the classifier

• And of course, use the misclassification cost matrix for the prediction in order to minimize the expected cost

Rule: select the label which minimizes the expected cost

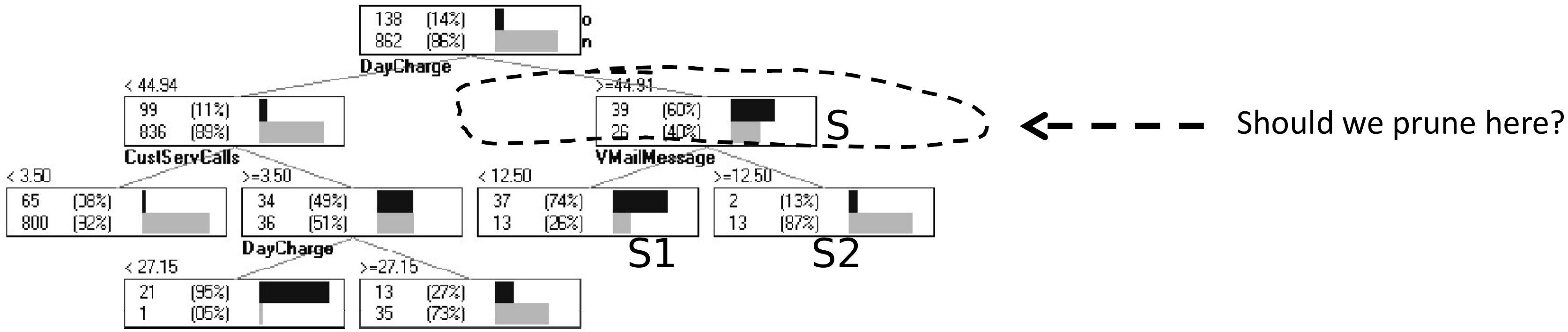
Main challenge: only few methods can be modified (in a simple way)

The decision tree algorithm is one of the few approaches that can incorporate the costs into the learning process: we focused on the post-pruning phase here.

Method 3: Cost-sensitive Decision Tree (CS-MC4, CART, etc.)

Growing phase: use the usual splitting measure (Shannon entropy, Gini index, etc.)

Post-pruning: use an approach which takes into account the costs



Post-pruning: compare the expected cost of misclassification at the node S with the weighted average of the expected costs on the leaves S1 and S2.

The idea is very close to that of C4.5 post-pruning algorithm except that instead of working on errors (pessimistic error), we work on misclassification costs. We must also penalize leaves with a few number of instances.

Method 3: CS-MC4

(1) Estimation of the posterior class probabilities with the Laplace smoothing

$$P(Y = y_k / S) = \frac{n_{ks} + \lambda}{n_s + \lambda \times K}$$

The higher is λ, the smoother is the estimation.

Usually, we set λ = 1 (see Laplace's Rule of Succession)

(2) Calculate the misclassification cost for the node

$$C(S) = \min_k C(y_k / S)$$
$$= \min_k \left[\sum_i P(Y = y_i / S) \times c_{ik} \right]$$

For a node S :

(a) Calculate the expected cost for each label

(b) Select the conclusion which minimizes the cost

(c) This cost corresponds the cost of the node

(3) Prune leaves if the weighted average of their costs is higher than the cost of preceding node.

Prune from a node:

(a) If the predictions of all the child nodes are identical with the node

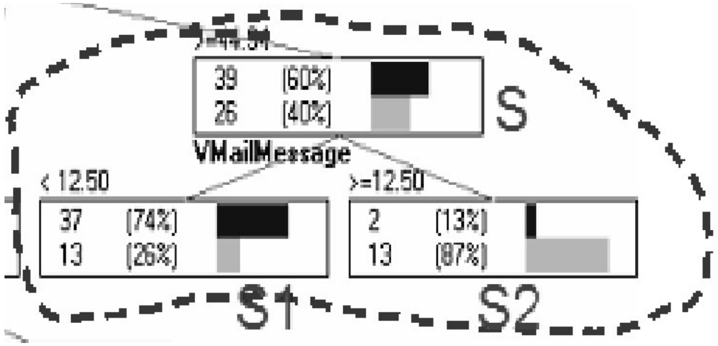
OR

(b) if $C(S) \leq \frac{n_{s_1}}{n_s} C(S_1) + \frac{n_{s_2}}{n_s} C(S_2)$

Method 3: CS-MC4 – An example

Cost matrix

Y \ $\hat{Y}$	+	−
+	−15	10
−	2	0



Node S

$$C(+/S)=\frac{39+1}{65+2}\times(-15)+\frac{26+1}{65+2}\times(2)=-8.15$$
$$C(-/S)=\frac{39+1}{65+2}\times(10)+\frac{26+1}{65+2}\times(0)=5.97$$

$\Rightarrow$ $\hat{Y}=+$, $C(S)=-8.15$

Leaf S1

$$C(+/S1)=\frac{37+1}{50+2}\times(-15)+\frac{13+1}{50+2}\times(2)=-10.42$$
$$C(-/S1)=\frac{37+1}{50+2}\times(10)+\frac{13+1}{50+2}\times(0)=7.31$$

$\Rightarrow$ $\hat{Y}=+$, $C(S1)=-10.42$

Leaf S2

$$C(+/S2)=\frac{2+1}{15+2}\times(-15)+\frac{13+1}{15+2}\times(2)=-1.0$$
$$C(-/S2)=\frac{2+1}{15+2}\times(10)+\frac{13+1}{15+2}\times(0)=1.76$$

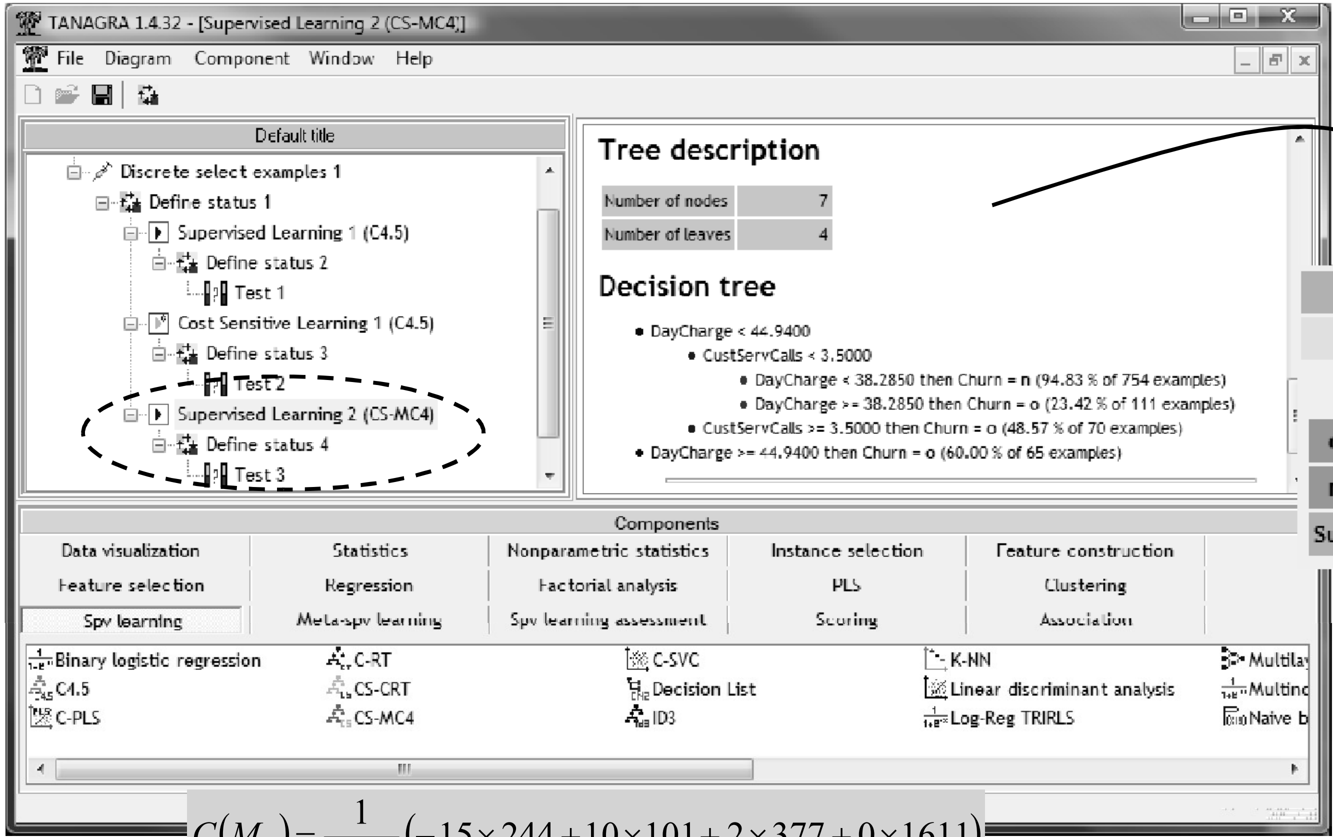
$\Rightarrow$ $\hat{Y}=+$, $C(S2)=-1.0$

!

Here, we prune from S because (a) all the nodes have the same conclusion.

(b) We note however that we have not a reduction of the costs: C(S)=-8.15 vs. 50/65 x C(S1) + 15/65 x C(S2)=-8.25

Method 3: CS-MC4 – “CHURN” problem



0.2049

Confusion matrix

	o	n	Sum
o	244	101	345
n	377	1611	1988
Sum	621	1712	2333

$$C(M_3) = \frac{1}{2333}\left(-15\times 244 + 10\times 101 + 2\times 377 + 0\times 1611\right)$$
$$= -0.8127$$

It greatly improves the results!

Improvement is based on an increase of the True Positive = 244

We note that the error rate is worse than M1, but this is not the matter.

Reminder

$$C(M_1) = -0.2679$$
$$C(M_2) = -0.4483$$
$$C(M_3) = -0.8127$$

Some other approaches

Other approaches: Cost-sensitive meta-learning with Bagging

Learning (P: number of classifiers)

For p = 1 to P

Sample with replacement (n among n)

Learn the classifier Mp

End For

Prediction for one instance ω

For p = 1 to P

Set the prediction with Mp → $\boxed{\hat{Y}_p(\omega)}$

End For

According to the proportions observed on the P's predictions, we have an estimate of

$$\boxed{P\left[Y = y_k / X(\omega)\right]}$$

Make the prediction which minimizes the cost by taking into account the misclassification cost matrix.

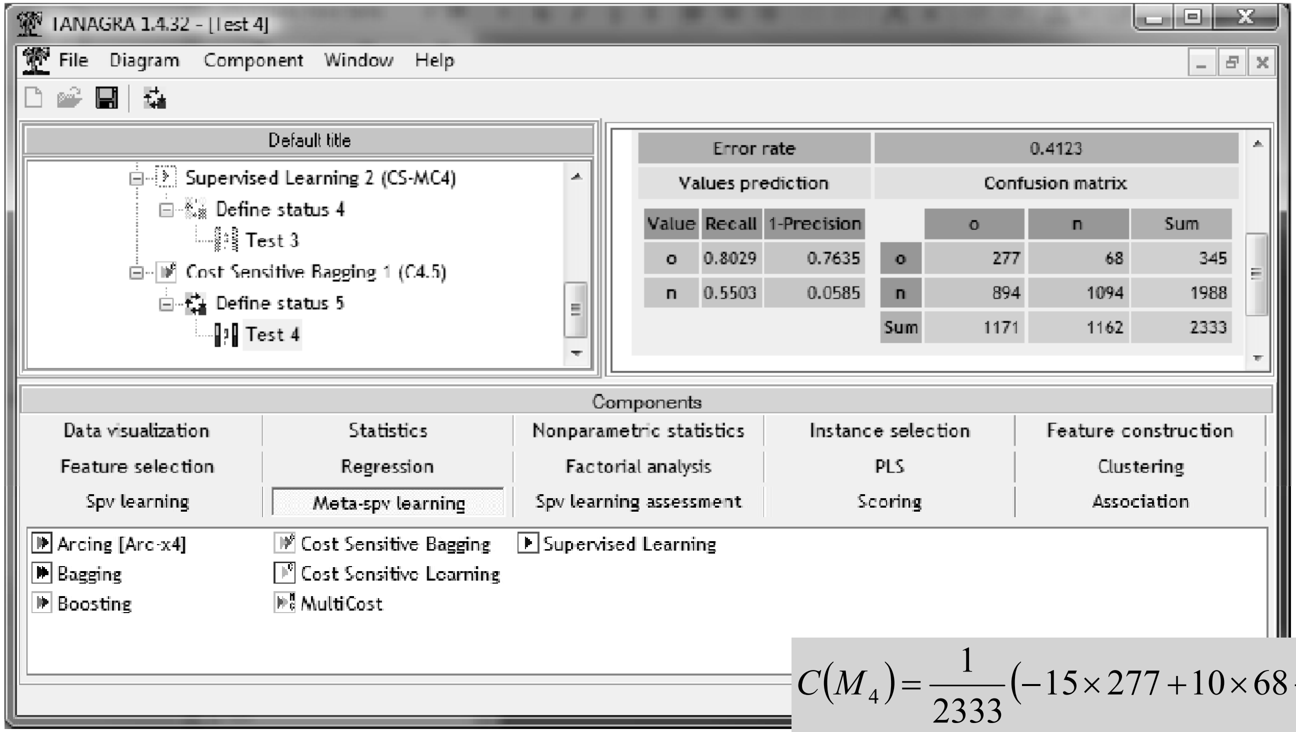
Pros

- The meta-classifier is often better than the individual classifier
- The approach is generic, it is applicable regardless of the underlying learning method
- It works even if the base classifiers do not provide a correct estimate of P (Y/X)

Cons

- If P is large, the calculation can be prohibitive
- The mechanism of the classification is not "readable" (we do not identify what is the underlying reason of a prediction)

Other approaches: CHURN problem - Bagging



$$C(M_4) = \frac{1}{2333}\left(-15\times 277 + 10\times 68 + 2\times 894 + 0\times 1094\right)$$
$$= -0.7231$$

*<u>Note:</u> One can include [**Tanagra**] the misclassification cost matrix for the predictions of the base classifiers Mp*

Other approaches: MetaCost

Idea: Make use of the performance of the Bagging, but provide only a single classifier as output (thus "readable") - Based on a re-labelling mechanism of individuals

Learning (P : number of classifiers)

(1) Learn a set of classifiers with the BAGGING approach
(2) Classify each instance of the learning sample (reclassifying the learning data)
(3) Use these predictions as labels for the construction of an unique classifier → we obtain the final classifier

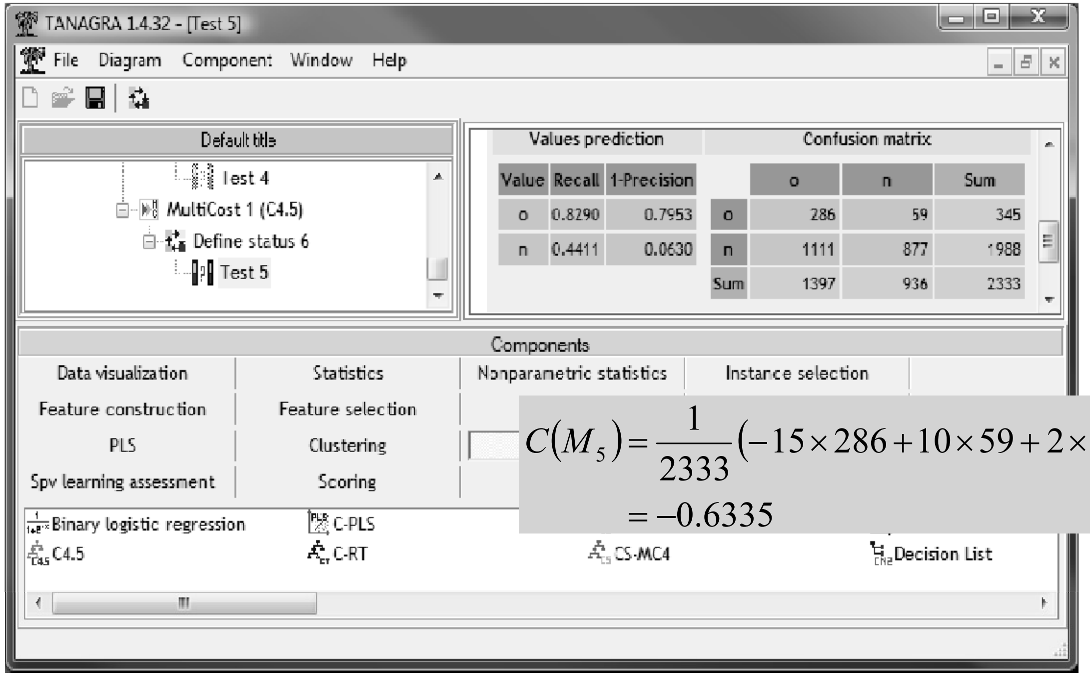
Pros

• **One unique classifier is obtained. The interpretation of the model is the same as for the usual learning scheme.**
• The approach is generic, it is applicable regardless of the learning algorithm.

Cons

• But there is no guarantee that the final unique model has same level of performance as the meta-classifier
• If P is large, the calculation can be prohibitive

Other methods: “CHURN” problem - MetaCost



$$C(M_5) = \frac{1}{2333}\left(-15\times 286 + 10\times 59 + 2\times 1111 + 0\times 877\right)$$
$$= -0.6335$$

For information purpose: cross tabulation between the original labels (observed) and the modified labels (used for the construction of the final classifier)

Cross-tab

	o	n	Sum
o	138	0	138
n	299	563	862
Sum	437	563	1000

All positive instances are kept positives

299 negative instances are re-labeled as positives

*<u>Note:</u> One can include [**Tanagra - MULTICOST**] the misclassification cost matrix for the predictions of the base classifiers Mp*

Conclusion

Comparison of the methods on the “CHURN” dataset

Method	ECM	Comments
M1 (ignore the costs)	-0.2679	This is the baseline solution. We must do better than this approach.
M2 (taking into account the costs during the prediction phase only)	-0.4483	We have the same model than M1. But we apply differently the model when we assign a class to one instance. This works only if the classifier can provide the class membership probabilities P(Y/X).
M3 (taking into account the costs during the construction of the classifier)	-0.8127	This is the best solution for the CHURN dataset. But only a few methods can be directly modified in order to take into account the costs (decision tree for our dataset).
M4 (Bagging)	-0.7231	Generic and powerful. But the meta-classifier is a black-box model. We do not perceive the underlying concept connecting the class attribute Y to the descriptors X.
M5 (MetaCost)	-0.6335	It tries to take advantage of the Bagging while providing an unique interpretable model for the classification. The performance reflects this intermediate position. It is applicable regardless of the base learning method.

Note: Other approaches based on re-weighting of instances exist…

References

Papers

There are a lot of papers online. Set "*cost sensitive learning*" in a web search engine.

Tutorials

Tanagra, "Cost-sensitive learning – Comparison of tools", March 2009.

http://data-mining-tutorials.blogspot.fr/2009/03/cost-sensitive-learning-comparison-of.html

Tanagra, "Cost-sensitive Decision Tree", November 2008.

http://data-mining-tutorials.blogspot.fr/2008/11/cost-sensitive-decision-trees.html