A New Cost Sensitive Decision Tree Method : Application for Mammograms Classification

Walid Erray,[†] *and Hakim Hacid*[††]*,*

ERIC lab. University of Lyon 2, France, ERIC lab. University of Lyon 2, France

Summary
Make a decision has often many results and repercussions. These results do not have the same importance according to the considered phenomenon. This situation can be translated by the introduction of the cost concept in the learning process. In this article, we propose a method able to take into account the costs in the automatic learning process. We focus our work on the misclassification cost and we use decision trees as a supervised learning technique. Promising results are obtained using the proposed method.

Key words:
Cost Sensitive learning, Decision Trees.

Introduction

Supervised learning aims to build a prediction model able to predict the class of an object starting from its descriptive features. Several supervised learning methods were proposed like decision trees [2]. The quality of the prediction model depends on several parameters as its interpretation facility, its success rate, and its complexity.

Making a decision has after-effects. These after-effects can be more or less serious according to the considered phenomenon. For example, in the medical domain, classify a positive diagnostic as a negative one has more serious after-effects than making the opposite. Unfortunately, in the traditional learning methods, all the decisions are considered to have the same importance. To take into account this kind of situation, the cost sensitive learning was introduced. That is, for a learning, we can associate several types of cost. Turney [15] has identified ten, quote for example, the misclassification cost [2], and the test cost.

In this article, we are interested in the cost sensitive learning and we deal especially with the misclassification cost. So, we propose a method, based on decision trees, able to integrate the real cost. For that, we intervene on the various levels of the decision tree construction process. Throughout this article we will use the term "cost" to indicate the "real misclassification cost".

The rest of this article is organized as follows: the following section introduces the notations used throughout this paper as well as a brief description of the related work. After that, we introduce the basic version of our decision tree based learning method in Section 3. In Section 4, we give the improved version of our learning method for considering the costs. Section 5 presents the experiments performed to validate our approach. The application of the method on mammograms classification is discussed in Section 6. Finally, we conclude and give future directions in Section 6.

2. Notations And Related Work

Consider a set of data Ω composed by n items $I_1, I_2, ..., I_n$, I_n described by p features $V_1, V_2, ..., V_p$. In this article we focus exclusively on the two classes problems. We note these two classes C_1 and C_2. The total misclassification cost Θ of a prediction model Φ obtained using a given learning method, is calculated starting from the confusion matrix M_f (Table 1) obtained after validation (either traditional or cross validation), and the costs matrix M_c (Table 2).

	$C_1^{'}$	$C_2^{'}$
C_1	n_{11}	n_{12}
C_2	n_{21}	n_{22}

Table 1. Confusion Matrix

A confusion matrix contains an account of all the items of the dataset according to the predicted class by the given learning algorithm. So, $C_1^{'}$ and $C_2^{'}$ represent the predicted classes. The number of objects, actually belonging to class C_i $i = 1,2$ and whose model predicts their membership in the class $C_j^{'}$ $j = 1,2$ is n_{ij}.

	$C_1^{'}$	$C_2^{'}$
C_1	0	c_{12}
C_2	c_{21}	0

Table 2. Costs matrix

The costs matrix represents the misclassification costs c_{ij} which is the related cost of predicting, for an object

Manuscript received November 5, 2006.
Manuscript revised November 25, 2006.

actually belonging to class *i*, to be an object belonging to the class *j*. The total misclassification cost is given by the following equation: $\Theta = c_{12} \times n_{12} + c_{21} \times n_{21}$

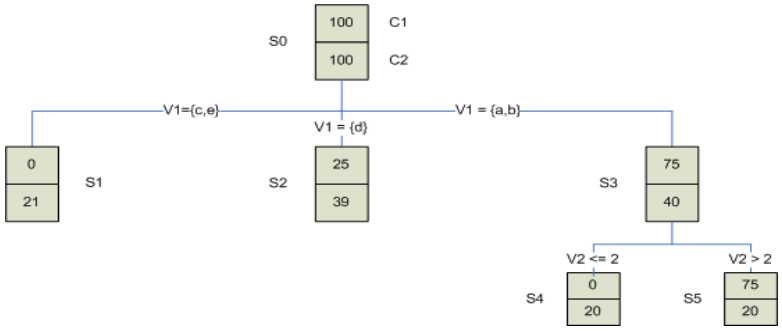

Figure 1. Example of a decision tree obtained by Basic Decision Tree Method

Weaker is Θ, better is the prediction model.

Several work, allowing to take into account the real misclassification cost was proposed. We can classify these methods into three main categories: the consideration of the cost before the learning process (handling the learning set for example), the consideration of the cost during the learning process (the use of specific measures, post-pruning, etc.), and finally, the consideration of the cost after the learning process (decision rules manipulation for example).

In the case of the costs consideration before the learning process, if Ω is a balanced data set (50% of the objects belong to the class C_1, and 50% of the objects belong to the class C_2), then the prediction probability of the class C_1 as well as that of the class C_2 will be about 0.5. If the misclassification cost of C_1 (c_{12}) is more significant than the misclassification cost of C_2 (c_{21}), it will be necessary to increase the probability of predicting the class C_1 in order to reduce the total cost. An intuitive and simple solution is to increase the number of objects belonging to the class C_1 in the learning set [16]. So, to have a probability p. of predicting the class C_1, it is necessary to multiply its initial objects count by the term $\frac{p^*}{(1-p^*)}$ [2]. In the same category we can quote Metacost[3].

In the decision trees based methods context, the goal is to build, in an iterative manner, a succession of partitions which lead to a good model. In order to measure the quality of each partition, one can use an information measurement like the Shannon's entropy [13]. In order to take into account the cost, certain authors propose other costs sensitive measures like the proposal of Dummond and Holte in [4]. A cost sensitive pruning [1] can also give very interesting results by combining it with a classical learning or with a cost sensitive learning. Other methods make it possible to take into account the cost after the learning process. The goal is to handle the obtained prediction model in order to reduce the total cost [10] [6] [7].

3. BASIC DECISION TREE METHOD

The general principle of our method is rather similar to that of the other decision tree based methods. In fact, starting from the main partition P_0, representing the root S_0 of the tree and containing all the objects of Ω, the features $V_1, V_2, ..., V_p$ are used to build, in an iterative way, a succession of increasingly detailed partitions of Ω.

The described tree in Figure 1 generates a partition P_1 of four final nodes $s_1, s_2, s_4, and\ s_5$. The node s_4 corresponds to the objects of Ω with the modalities $V_1 = \{a, b\}$ and $V_1 \leq 2$. The transition from a partition P_t to another partition P_{t+1} is performed by splitting a node S_k using the feature V_i which generates the partition with the best quality. The tree construction process stops when no improvement (partition having better quality) can be obtained.

3.1 Association and selection in Basic Decision Tree Method

In order to produce a decision tree with a low complexity and a good quality, we adopt a popular principle used in ChAid[9] and CART [2]. This principle consists of gathering the values of the predictive features, having the same behaviour with respect to the predictive class during the node splitting.

The association principle, FaUR, used in our method is described in [5]. So, We start from the finest partition,

we seek in each iteration, the two best candidates columns to the fusion. These two columns are those whose fusion maximizes the total value t_s of Tschuprow measure [14] [17] based on the Chi2 measure. The algorithm stops when no fusion can increase the value of t_s.

$$t_s = \frac{\chi^2}{n\sqrt{(l-1)}}$$

$$\chi^2 = \sum_{i=1}^{2} \sum_{j=1}^{l} \frac{(n_{ij} - \frac{n_{i.}n_{.j}}{n})^2}{\frac{n_{i.}n_{.j}}{n}}$$

In the case of a quantitative feature, we start by sorting the values of the feature. Also, in this case, only the adjacent columns can be merged in order to obtain disjoints intervals.

The optimized version of FaUR [5] makes it possible to perform associations with a complexity of *O(l log l)*. This association method is applied to the contingency tables of all the features. As a finality, we select the feature V_j maximizing the t_s value of its contingency table.

3.2 Stopping Criteria

The tree construction (nodes splitting) is stopped if one of the two following conditions is satisfied:

- *Minimum objects count in a node*: We admit that a rule obtained from a leaf node is valid only if it is checked by a minimum number of objects (*effmin*). In other words, a leaf node whose minimum object count, of the dominant class, is lower than effmin will not be able to produce a valid rule. From that, any node which do not respect this constraint will not be developed. This is called the pre-pruning process.
- *Node homogeneity*: A node S_k is considered to be homogeneous if it contains only objects belonging to the same class C_i. In this case, any node obtained from the splitting of S_k will, automatically, have the same conclusion as S_k, i.e. C_i. Thus, it is useless to continue the splitting of such a node.

3.3 Basic Decision Tree Method Algorithm

Algorithm 1 summarizes the operations performed in the proposed method.

Obtaining the rules starting from a decision tree is made on the decision tree's leaves. Let us recall that a rule is composed by a conjunction of conditions and a conclusion.

The conditions are obtained by traversing the tree from the root to the leaves, each traversed node brings a condition to the rule. A conclusion generated by a leaf node is considered as valid if the minimum object count constraint is satisfied (the object count of the dominant class is higher than the predefined threshold). In the opposite, the conclusion is determined by the parent node.

```
L : set of free nodes
A = V1, . . . , Vp : set of features
Div(k, j) : features obtained after splitting Sk with Vj ;
L = S0
while (L ≠ ∅ ) do
    Let Sk ∈ L
    if ((Sk ≠ homogne) and (Sk ≠ effmin)) then
        tsmax = −1;
        best = −1;
        for i = 1 to p do
            Get(Ti), Ti : the contingency table of Vi;
            FaUR(Ti);
            tsi = ts(Ti)
            if (tsi > tsmax) then
              best = i;
              tsmax = tsi;
            end if
        end for
        Split Sk with Vbest
        L = L + Div(k, best)
    end if
    L = L − Sk
end while
```

Algorithm 1. Basic decision tree method algorithm

Consider the example of Figure 1. If *effmin = 25*, then the rule generated by the node S_5 will conclude on C_1 because this node respects the minimum objects count constraint. However, the conclusion of the rule generated by the node S_4 inherits from that of its parent node (S_3) because it does not respect the minimum objects count constraint. The application of the described stages produces a decision tree having a rather similar aspect to the classical decision trees. In the following, we'll introduce our main contribution for the consideration and the integration of the costs in the learning process by extending the above described method.

4. COST SENSITIVE DECISION TREE

The final goal of our work is to propose a learning algorithm sensitive to the real misclassification costs. As we quoted it before, this can be carried out on three levels: before the learning process, during the learning process, and after the learning process. In what concern us, we do not intervene before the learning, i.e. no data handling is made.

Using the decision trees as a learning method, our contribution intervene at the construction level of the tree, at the post-pruning level, and at the generation of the rules level.

More concretely, the idea is to intervene, first, locally, on the node level during the processing of the contingency tables, during the selection of the splitting features, and during the pre-pruning. After that, we act on a global level, and this, during the pruning (post-pruning) and during the decision rules generation.

4.1 Local level

4.1.1 A new measure for contingency table association and splitting feature selection

The features' modalities association in the contingency table as well as the choice of a splitting feature depend on the value of the Tschuprow measure.

These two operations aim to maximize the value of this measure. However, the problem related to this measure is that it considers the costs of the classes equivalent and are equal to 1. In other words, it does not take into account the affected costs to the classes.

In order to consider the costs, we propose a new quality measure, t_{cost}, based on the *Tschuprow* measure and introducing a new element representing the cost. The measure is illustrated by the following formula:

$$t_{cost} = ts - E$$

$$E = \begin{cases} if\ (n_{12}=n_{21}) \begin{cases} if\ (c_{12}<c_{21})\ Then\ E_1 \\ else\ E_2 \end{cases} \\ else\ E_3 \end{cases}$$

$$t_{cost} = t_s - E$$

where:

$$E_1 = c_{12} \times n_{12}$$

$$E_2 = c_{21} \times n_{21}$$

$$E_3 = \sum_{j=1}^{2} (E_{31} + E_{32})$$

$$E_{31} = c_{12} \times Min(Max_i n_{ij} - n_{1j};1) \times n_{1j}$$

$$E_{32} = c_{21} \times Min(Max_i n_{ij} - n_{2j};1) \times n_{2j}$$

4.1.2 Pre-pruning

This task is ensured by the introduction of the minimum objects count concept. We already introduced a first vision of the minimum objects count concept (*effmin*) in the previous sections. This concept translates the conditions of the decision-making at a given node. The principle is that a conclusion on a class C_1 at a node S_k must be checked by a preset minimum object count by disregarding the costs.

Let us consider the two classes C_1 and C_2. If c_{12} is higher than c_{21}, this means that a bad decision on C_2 (classify an object belonging to C_1 in C_2) has more after-effects than a bad decision on C_1 (classify an object belonging to C_2 in C_1). We translate this by a stronger penalization of the decision-making on C_2. This penalization can be translated by the fact that the necessary objects count for a conclusion on C_2 must be higher than that for the class C_1.

In order to integrate this concept in our method, we introduce an additional minimum objects count (*effmin2*). This concept corresponds to the minimum objects count that a rule concluding on C_2 must satisfy to be considered as valid.

To penalize the class C_2, we set its own minimum objects count (*effmin2*). Intuitively, the minimum objects count assigned to each class is inversely proportional to its cost.

We illustrate this concept on the example of Figure 2 where we consider *effmin*(C_1) = 10 and *effmin2*(C_2) = 25. So, splitting the node S_1 will continue since we have a possibility of obtaining nodes that will conclude on C_1 and respecting *effmin* although S_1 does not respect *effmin2*. In the node S_2, the number of objects belonging to the class C_2 is lower than *effmin2*. Split this node cannot produce any more nodes concluding on C_2 while respecting the *effmin2* constraint. For that, the splitting is not useful any more, it is then stopped. With regard to the node S_3, the class C_1 violates the minimum objects count constraint what prevents the continuation of the splitting task.

4.2 Global level

4.2.1 Post-pruning

The post-pruning is a significant operation and is necessary to obtain a tree with a rather good quality and to prevent a high complexity of the tree. Indeed, some leaves of the tree can sometimes be useless (too specialized tree). In such a situation, a post-pruning is applied.

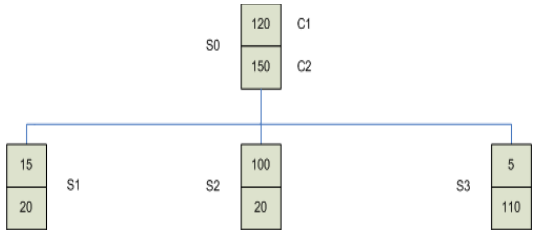

Figure 2. Illustration of the pre-pruning principle

At this level, we consider the real cost θ_{test}, calculated on a test data set, combined with the complexity of the tree $\pi(\pi = \alpha \times leaves\ count)$ (α : predefined by the user) to perform the pruning. The goal is to minimize the total cost, $\theta_{tot} = \theta_{test} + \pi$.

4.2.2 Generation of the rules

Consider the decision tree illustrated in Figure 3 with the following misclassification costs: $c_{12} = 5$ and $c_{21} = 1$.

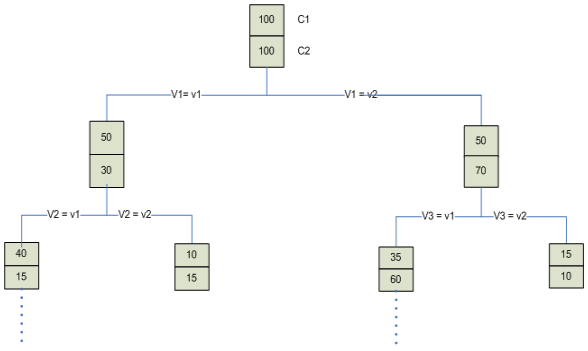

Figure 3. Illustration of the rules generation principle

Classically, if the minimum objects count constraint is $eff\min = 15$ objects, then among the rules we can obtain:

- $R_1 : if\,(V_1 = v_1) AND (V_2 = v_2) THEN\ \ Class = C_2$
- $R_2 : if\,(V_1 = v_2) AND (V_3 = v_2) THEN\ \ Class = C_1$

In the standard case, we notice that the two conclusions were taken with the same number of objects (15 objects in this case). This means that we have the same probability to do an error in both cases. However, by taking into account the costs, the consequences of the rule R_1 are more important than the consequences of the rule R_2. In this case, it is necessary that the rules concluding on C_2 have less probability of doing a mistake on the class C_1.

By considering the previous example, if one sets the minimum objects count related to C_2, *effmin2* = 30 objects while leaving objects count related to C_1 (*effmin* = *15*), then the conclusion of the rule R1 will be not valid. In this case, we consider that the conclusion of the rule R1 inherits from the conclusion of the parent node. Thus, We will have the two following rules:

- $R_1 : if\,(V_1 = v_1) AND (V_2 = v_2) THEN\ \ Class = C_1$
- $R_2 : if\,(V_1 = v_2) AND (V_3 = v_2) THEN\ \ Class = C_1$

So, in order to introduce the costs for the decision-making in a node, we use two different minimum objects count: The first one is used to make a decision on C_1, and the second one is used to make a decision on C_2.

By taking into account the presented elements in this paper, we integrated the cost concept in the decision tree construction process. The suggested method makes it possible to support a class having a high cost, and also makes it possible to keep a good quality of the tree in order to prevent the loss of the decision on the other class. The following section introduces the performed experiments to validate the proposed approach.

5. EXPERIMENTS AND RESULTS

5.1 Parameters Evolution

Setting up the initial values of the parameters is often problematic for the user. In our case, the second minimum objects count (*effmin2*) parameter initialization can be difficult. In order to give an idea on the possible values to set for this parameter, we will give the evolution and the possible relation between the different parameters.

To perform these experiments, we used breast-cancer data set described in [12]. We balanced this data set in order to have the same decision probability at the beginning of the evaluation. The general principle of the tests is as follows:

we make correspond for each cost various values of minimum objects count (*effmin2*). For the two test series1,

we set the cost c_{12} of the class C_1 at 1. We vary the cost of the second class C_2 using values from 5 up to 150. Also, for each value of c_{21}, we vary effmin2 from 5 up to 150. *effmin* as for it is defined to 5 for all the experiments. We recover at each iteration, the obtained gain.

To show the interest of the method, we made the tests in the first case by supporting class 1 ($C_1 = 1$, $C_2 = 2$) and in the second case by supporting the class 2 ($C_1 = 2$, $C_2 = 1$). Curves of figures 4 and 5 show the obtained results.

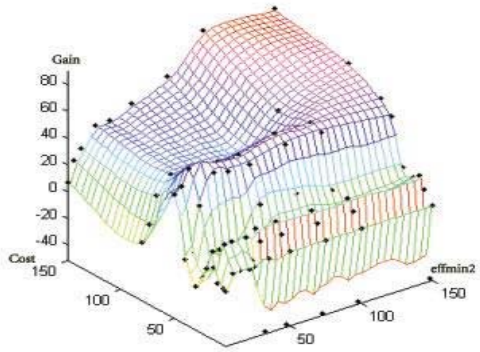

Figure 4. Evolution of the gain according to the cost and the minimum objects count for the class C2 = 1

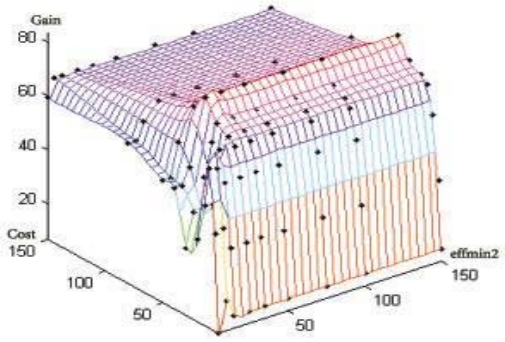

Figure 5. Evolution of the gain according to the cost and the minimum objects count for the class C2 = 2

We can notice that the curves have approximately the same behaviour for the two tests series (for the two classes). Globally, one can say that the minimum objects count depends on the associated cost to a class. Indeed, We notice that the obtained gain, for a given cost, increases by increasing *effmin2*. However, from a certain value of *effmin2*, the gain does not increase any more and can even be deteriorated (Figure 5).

5.2 Tests and results

For the effective evaluation, we took datasets from the UCI Irvine repository [12], and we use the following ones: Australiancredit, Breast-Cancer, Heart and White-House-Votes-84. These data sets are two classes problems. We balanced the datasets by taking the totality of the objects of the class having less objects count.

We performed three tests series on each data set and on each class. The three tests series correspond respectively to costs 5, 10 and 15. Also, we used three methods: *C4.5* [11], *BDTM*, and *CSDTM*. For *CSDTM*, we use *effmin2* with the values 5, 10 and 15. Table of Figure 8 (in the end of this paper) summarizes the obtained results in term of real misclassification cost.

Initially, we can affirm that our initial method offers better results compared to C4.5, for these datasets. In addition, the CSDTM method makes it possible, in the majority of the cases, to reduce the total cost. Certainly, the cost does not decrease for all the *effmin2*' values (heart, c_{21} = 15, $C_2 = 1$, *effmin2* = 15), but there is, at least, a value of effmin2 for which the cost decreases.

These results show certainly the utility and the interest of the method. In the next section, we describe and discuss an application of the suggested method on mammograms classification.

6. MAMMOGRAMS CLASSIFICATION

6.1 Description of the database

.We used a data set issued from the DDSM(DIGITAL Database for Screening Mammography) of the University of the South Florida [8]. This database includes 2620 breast cancer cases representing malignant and benign cancers (cf. Figure 6). Each mammogram was studied by a doctor and zones comprising cancerous zones are identified.

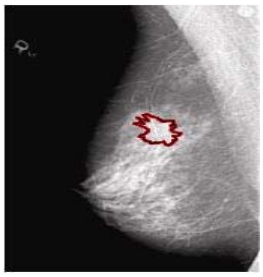

Figure 6. Example of the studied mammograms

6.2 Processing the database for features extraction

In order to study this database, each mammogram is divided into several circles of equal sizes allowing to cover the totality of the mammogram (cf. Figure 7). Then, we extract from each circle some characteristics corresponding to the colour histogram (256 features). Thus, each circle will be considered as an item (object). If the circle crosses the cancerous zone, then it will be marked as a malignant item (*Class=malignant*), and if its not the case it will be marked as a benign item (*Class=benign*). Thus, our target (class) feature will have two possible values: benign and malignant. At the end, we consider a dataset with 12.417 items.

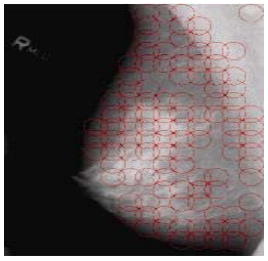

Figure 7. Processing of the mammograms

6.3 Costs introduction

The bad classification of a malignant circle is more serious than a bad classification of a benign circle. In order to decrease the number of bad classifications of the "malignant" items, we will apply the proposed method.

A cross validation (10 iterations with a training on 90% circles and a validation on the 10% of the remaining circles) made using the basic decision tree method (costs of bad classifications = 1) give the following confusion matrix (*cf.* Table 3).

	Benign	Malignant
Benign	9911	2134
Malignant	36	336

Table 3. confusion Matrix obtained using Arbogodai[17]

We will perform series of tests in order to observe the evolution of the bad classifications on the class "malignant" according to the misclassification cost. Also, we will show the interest of using a measure including the cost during the selection of the best splitting feature at each node and during the association of the features modalities.

An effective method should decrease the number of misclassification on the class "malignant" if the cost of this bad classification increases. Thus, we distinguish three stages:

- In the first stage, we will use the standard version of Tschuprow measure during the features' modalities association and the proposed version of *Tschuprow* during the best splitting feature selection. Any increasing of *effmin2* will be performed.
- In a second stage we introduce the cost at the selection time (*costS*) or during association (*costA*) while defining *effmin2* to 20.
- In the last stage, we introduce the cost at the selection and the association levels while increasing *effmin2*.

Step 1 : effmin2=2 and we vary the misclassification cost of malignant circles during the selection stage only. We obtain the results described into table 5 and 5 respectively.

	Benign	Malignant
Benign	10065	1980
Malignant	46	326

Table 4. Confusion matrix with CostS=5

	Benign	Malignant
Benign	9938	2107
Malignant	37	335

Table 5. Confusion matrix with CostS= 100

We note a deterioration of the results. Indeed the number of miss classifications of the malignant circles increases. These results confirm the previously obtained ones: Taking into account the costs in the used measure is not enough to decrease the misclassification rate while having *effmin = effmin2*.

Step 2 : effmin2=20 and we vary the misclassification cost of the malignant circles.

By defining *effmin2* to 20 and increasing the cost, we obtain the results of tables 6, 7, 8 and 9 respectively.

	Benign	Malignant
Benign	8300	3745
Malignant	11	361

Table 6. Confusion matrix with CostS=5

The obtained results show a reduction in the number malignant circles bad classification when increasing *effmin2* and costS/costA.

	Benign	Malignant
Benign	7802	7243
Malignant	8	364

Table 7. Confusion matrix with CostS=100

	Benign	Malignant
Benign	8617	3428
Malignant	12	360

Table 8. Confusion matrix with CostA=5

	Benign	Malignant
Benign	8175	3870
Malignant	11	361

Table 9. Confusion matrix with CostA=100

	Benign	Malignant
Benign	9252	2793
Malignant	23	349

Table 10. Confusion matrix with Cost=5, effmin2=5

Step 3: We vary effmin2 and the misclassification cost of the malignant circles.

By increasing effmin2, costS (=cost) and costA (=cost), we obtain the results showed in tables 10, 11, 12, 13, 14, and 15 respectively.

	Benign	Benign
Benign	8684	3361
Malignant	17	355

Table 11. Confusion matrix with Cost=10, effmin2=10

	Benign	Malignant
Benign	8187	3858
Malignant	14	358

Table 12. Confusion matrix with Cost=10, effmin2=20

	Benign	Malignant
Benign	7466	4579
Malignant	8	364

Table 13. Confusion matrix with Cost=100, effmin2=20

	Benign	Malignant
Benign	7274	4771
Malignant	4	368

Table 14. Confusion matrix with Cost=1000, effmin2=20

	Benign	Malignant
Benign	6568	5477
Malignant	3	369

Table 15. Confusion matrix with Cost=1000, effmin2=50

	Benign	Malignant
Benign	5320	6725
Malignant	0	372

Table 16. Confusion matrix with Cost=1000, effmin2=60

The obtained results show clearly the interest of the proposed method. So, we can conclude from that, that increasing the misclassification cost of the malignant circles and *effmin2* involves systematically a reduction in the misclassification rate of the malignant circles. Note that the objective of the application was to reduce the misclassification cost of malignant items what is done successfully (See Table 16).

7. CONCLUSION AND FUTURE WORK

The cost sensitive learning is a very significant problem in the machine learning community. Several work was devoted to this subject. In this work, we were interested in the misclassification and used the decision trees like learning method. We proposed a method able to take into account the misclassification cost in the various steps of the decision tree construction process while keeping a good quality of the tree. The major contribution of this work is, certainly, the intervention on the various levels of the learning process (decision tree construction). The performed tests show very interesting results.

As future work, we plan to find an automated manner for initializing the *effmin2* parameter. Also, we plan to extend the method to multi-classes datasets.

References

[1] J. P. Bradford, C. Kunz, R. Kohavi, C. Brunk, and C. E. Brodley. Pruning decision trees with misclassification costs. In ECML '98: Proceedings of the 10th European Conference on Machine Learning, pages 131–136, London, UK, 1998. Springer-Verlag.

[2] L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and Regression Trees. Wadsworth, 1984.

[3] P. Domingos. Metacost: A general method for making classifiers cost-sensitive., 1999.

[4] C. Dummond and R. C. Holte. Exploiting the cost (in)sensitivity of decision tree spliting criteria. In M. Kaufmann, editor, In Machine Learning : Proceedings of the Seventeeth International Conference, pages 239–246, San Francisco, CA, 2000. Morgan Kaufmann.

[5] W. Erray. Faur : Mthode de rduction unidimensionnelle d'un tableau de contingence. In SFC05 : 12me rencontres de la Socit Francophone de Classification, Montreal, Canada, May 2005.

[6] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In International Conference on Machine Learning, pages 148–156, 1996.

[7] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting, 1998.
[8] M. Heath, K. Bowyer, D. Kopans, R. Moore, and P. Kegelmeyer. The digital database for screening mammography. In W. Medical Physics. Publishing (Madison, editor, The Proceedings of the 5th International Workshop on Digital Mammography, Toronto, Canada, June 2000.
[9] G. Kass. An exploratory technique for investigating large quantities of categorical data. j-APPL-STAT, 29(2):119–127, 1980.
[10] J. Platt. Probabilistic outputs for support vector machines and comparison to regularize likelihood methods. In B. S. D. S. A.J. Smola, P. Bartlett, editor, Advances in Large Margin Classifiers, pages 61–74, 2000.
[11] J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993.
[12] C. B. S. Hettich and C. Merz. UCI repository of machine learning databases, 1998.
[13] C. Shannon and W. Weaver. The Mathematical Theory of Communication. The University of Illinois Press, 1949.
[14] A. Tschuprow. On the mathematical expectation of moments of frequency distribution. Biometrika, pages 185–210, 1921.
[15] P. D. Turney. Types of cost in inductive concept learning. cs.LG/0212034, 2002.
[16] B. Zadrozny and C. Elkan. Learning and making decisions when costs and probabilities are both unknown. In KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pages 204–213, New York, NY, USA, 2001. ACM Press.
[17] D. A. Zighed, G. Ritschard, W. Erray, and V.-M. Scuturici. Arbogodaï, a new approach for decision trees. In PKDD, pages 495–506, 2003.

.

					CSDTM		
Cost	Dataset	C2	C4.5	BDTM	effmin2=5	effmin2=5	effmin2=5
5	Australian	1	331	298	217	192	194
	Australian	0	311	314	276	279	266
	Breast	2	62	80	87	76	76
	Breast	1	70	88	50	49	56
	Heart	1	223	114	170	158	156
	Heart	2	251	162	143	133	140
	House	REPUBLICAN	64	35	35	35	35
	House	DEMOCRAT	80	55	55	55	55
10	Australian	1	611	543	453	404	326
	Australian	0	566	579	479	380	315
	Breast	2	112	145	155	115	115
	Breast	1	130	163	55	56	66
	Heart	1	403	199	268	202	195
	Heart	2	466	307	255	227	230
	House	REPUBLICAN	114	60	60	60	60
	House	DEMOCRAT	150	105	105	105	105
15	Australian	1	891	788	631	559	470
	Australian	0	821	844	585	412	284
	Breast	2	162	210	180	180	188
	Breast	1	190	238	90	62	71
	Heart	1	583	284	358	265	248
	Heart	2	681	452	382	318	320
	House	REPUBLICAN	164	85	110	100	100
	House	DEMOCRAT	220	155	155	155	155

Figure 8. Results on some UCI Irvine datasets.

Walid Erray received his PhD degree in computer science from the University of Lyon 2. He is currently a Data Mining engineer at TOTAL Research Centre of Gonfreville. His research interests include data mining, databases and Information retrieval. More details are available at http://eric.univ-lyon2.fr/~werray/.

Hakim Hacid received his Master degree in computer science from the university of Lyon 2. He is currently a PhD student at ERIC laboratory at the university of Lyon 2. His research interests include databases, data mining, and multimedia. More details are available at http://eric.univ-lyon2.fr/~hhacid/