A Hybrid Algorithm for Medical Diagnosis

Camelia Vidrighin Bratu*, Cristina Savin* and Rodica Potolea*
* Technical University of Cluj-Napoca, Computer Science Department, Cluj-Napoca, Romania

Abstract – **Medical diagnosis and prognosis is an emblematic example for classification problems. Machine learning could provide invaluable support for automatically inferring diagnostic rules from descriptions of past cases, making the diagnosis process more objective and reliable. Since the problem involves both test and misclassification costs, we have analyzed ICET, the most prominent approach in the literature for complex cost problems. The hybrid algorithm tries to avoid the pitfalls of traditional greedy induction by performing a heuristic search in the space of possible decision trees through evolutionary mechanisms. Our implementation solves some of the problems of the initial ICET algorithm, proving it to be a viable solution for the problem considered.**

I. INTRODUCTION

Decision tree learning represents one of the simplest, yet most popular methods for inductive inference. It has been successfully applied to a wide variety of problems from medical diagnosis to air traffic control or the assessment of credit risk for loan applicants. Its popularity is justified by the fact that it has some key advantages over other inductive methods. First of all, decision trees offer a structured representation of knowledge (as disjunction of conjunctive rules). As a direct consequence, decision trees may be rewritten as a set of"if-then" rules, increasing human readability. Secondly, decision trees are robust to errors, requiring little or no data preprocessing. Other important features include the capacity of handling both nominal and numeric attributes, as well as missing values and a good time complexity even for large data sets.

An emblematic example of classification problem is that of medical diagnosis and prognosis. Our study focuses on the characteristics of this particular problem, although the results are not restricted to this domain. The need of using machine learning techniques in the medical field is rooted in the fact that the accuracy of diagnosis and prognosis is, in many cases, rather low, despite the boost in biomedical technology. The reasons for this situation are multiple. First of all, medical diagnosis is known to be subjective; it depends on the physician making the diagnosis (his experience, intuition and biases, the psycho-physiological conditions). Secondly, and most importantly, the amount of data that should be analyzed to make a good prediction is usually huge. In this context, machine learning can be used to automatically infer diagnostic rules from descriptions of past, successfully treated patients, and help specialists make the diagnostic process more objective and more reliable.

Records of previous patients are gathered into hospital archives and can be made available through machine learning techniques; the classifier derived from this data provides support for future diagnosis and treatment and can help improve the physician's speed, accuracy and reliability in establishing a new diagnosis. Also, it may offer invaluable support in the training of students and for non-specialists.

II. COST-SENSITIVE CLASSIFICATION

Typically, the task of classification is concerned with error reduction, i.e. the minimization of the number of errors. However, it has been recognized that, in real world problems, the cost of different errors is rarely the same. Recent studies suggest that more complex measures for evaluating the adequacy of classifiers should be considered. For example, non-uniform costs are inherent in fields such as medical diagnosis (classifying a healthy patient as ill is far less expensive than the reverse situation), or fraud detection.

Such problems are addressed by *cost-sensitive classification*, which is directed towards the reduction of the total cost, instead of just minimizing the number of misclassification errors.

Turney [10] provides a general taxonomy of costs involved in inductive concept learning, the most important of which being *misclassification costs* and *test costs*. The first category encompasses the costs which are conventionally considered by most cost-sensitive classifiers; however, several solutions address the second category also. A brief survey of the most important cost-sensitive classifiers, as described in the literature, will be provided in the following.

Chronologically one of the first solutions for the problem of reducing the total misclassification cost was a procedure called *stratification*. In this approach, the actual classifier is not altered in any way; instead, the distribution of examples for each class is changed. The modified training set includes proportionally more examples of the classes having high misclassification costs and may be generated either by undersampling or by oversampling. Each alternative comes at a certain price (see [1] for a detailed discussion on the subject), but the most serious limitation of the approach is that it restricts the dimension or the form of the misclassification cost matrix - the technique it is only applicable to two-class problems or to problems where the cost is independent of the predicted class. More complex techniques, which overcome these limitations, usually involve meta-learning algorithms, which typically are applicable to a range of base classifiers. In this category we include algorithms

based on various *ensemble* methods, such as *AdaBoost.M1* [3], *AdaCost* [2], or *MetaCost* [1] and those which take an *evolutionary* approach, the best-known being ICET [9].

AdaBoost.M1, first introduced by Freund and Schapire [3], employs an ensemble method, by combining several weak classifiers through voting; the resulting composite classifier generally has a higher predictive accuracy than any of its components. Each distinct model is build through the same learning mechanism, by varying the distribution of examples in the training set. After each boosting phase, the weights of the misclassified examples are increased, while those for the correctly classified examples are decreased. It has been mathematically proved that the error rate for the composite classifier on the unweighted training examples approaches zero exponentially with an increasing number of boosting steps [3], [6]. Also, various experimental results report that the reduction in error is maintained for unseen examples.

Another solution for reducing misclassification costs is *MetaCost* [1]. The algorithm is based on the Bayes optimal prediction principle, which minimizes the conditional risk of predicting that an example belongs to class *i*, given its attributes *x*. The solution requires accurate estimates for the class probabilities of examples in the training set. This distribution is obtained through an ensemble method, by uniform voting from individual classifiers. Once the conditional probabilities are estimated, the algorithm re-labels the examples in the training set, according to their optimal predictions and generates the final classifier, using the modified training set. The main advantages of this procedure are related to its applicability to wide range of base classifiers, the fact that it generates a single, understandable model, and its efficiency under changing costs (the conditional probabilities need to be computed only once, after which they can be used to generate models for various cost matrices).

Several approaches exist also for tackling the problem of test costs. They typically involve some alteration of the information gain function, as to make it cost-sensitive. Various cost dependent functions have been proposed in the literature, such as *EG2*, *IDX* or *CS-ID3* [9].

Significantly less work has been done for aggregating several cost components. The most prominent approach in the literature is *ICET*, which combines a greedy search heuristic (decision tree) with a genetic search algorithm. Other possible solutions are explored in [4], [7] and [8].

Medical diagnosis is one field in which such an aggregated approach is of utmost importance. First of all, a doctor must always consider the potential consequences of a misdiagnosis. In this field, misclassification costs may not have a direct monetary quantification, but they represent a more general measure of the impact each particular misclassification may have on human life. These costs are non-uniform (diagnosing a sick patient as healthy carries a higher cost than diagnosing a healthy patient as sick). Another particularity of the medical diagnosis problem is that medical tests are usually costly. Moreover, collecting test results may be time-consuming; arguably time may not be a 'real' cost, but it does have some implication for the decision whether it is practical to take a certain test or not. In the real case, performing all possible tests in advance is unfeasible and only a relevant subset should be selected. The decision on performing or not a certain test should be based on the relation between its cost and potential benefits. When the cost of a test exceeds the penalty for a misclassification, further testing is no longer economically justified.

III. THE ICET ALGORITHM

Classical tree induction uses *hill climbing* search, which, as most greedy techniques, suffers from the *horizon effect*, i.e. it tends to get caught in local optima. One possible way to avoid the pitfalls of simple greedy induction is to perform a heuristic search in the space of possible decision trees through evolutionary mechanisms.

ICET (Inexpensive Classification with Expensive Costs) is such a hybrid algorithm. Introduced by Peter Turney, the technique tackles the problem of cost-sensitive classification by combining a greedy search heuristic (decision tree) with a genetic algorithm [9].

The GA evolves a population of parameters, each individual corresponding to one decision tree. Standard mutation and crossover operators are applied to the trees population and, after a fixed number of iterations, the fittest individual is returned. The genetic algorithm used in [9] is GENESIS, and the decision tree algorithm is a modified version of Quinlan's C4.5 [5], which uses ICF (Information Cost Function) as attribute selection function, same as in EG2.

For the i^{th} attribute, ICF may be defined as follows:

$$ICF_i = \frac{2^{\Delta I_i}}{(C_i + 1)^w}, \quad where \quad 0 \le w \le 1 \qquad (1)$$

An important remark is that, unlike *EG2*, *ICET* does not minimize test costs directly. Instead, it uses ICF for the codification of the individuals in the population. The n costs, C_i, are not true costs, but *bias parameters*. They provide enough variation to prevent the decision tree learner from getting trapped in a local optimum, by overrating/ underrating the cost of certain tests based on past trials' performance. However, it is possible to use true costs, when generating the initial population, which has been shown to lead to some increase in performance.

Each individual is represented as a bit string of $n + 2$ numbers, encoded in Gray. The first n numbers represent the bias parameters ('alleged' test costs in the ICF function). The last two stand for the algorithm's parameters CF and w; the first controls the level of pruning (as defined for C4.5), while w is needed by ICF.

The *fitness function* for an individual is computed as the average cost of classification of the corresponding tree (obtained by randomly dividing the training set in two subsets, the first used for the actual tree induction and the second for error estimation). The average cost of classification is obtained by normalizing the total costs (obtained by summing the test and misclassification costs) to the training set size.

Test costs are specified as attribute - cost value pairs. The classification costs are defined by a cost matrix $(C_{ij})_{nxn}$, where C_{ij} - the cost of misclassifying an instance of class *j* as being of class *i*. If the same attribute is tested twice along the path (numeric attribute), the second time its cost is 0.

IV. ICET ENHANCEMENTS AND EXPERIMENTAL WORK

A significant problem related to the *ICET* algorithm is rooted in the fact that costs are learned indirectly, through the fitness function. Rare examples are relatively more difficult to be learned by the algorithm. This fact was also observed in [9], where, when analyzing complex cost matrices for a two-class problem, it is noted that: *it is easier to avoid false positive diagnosis [...] than it is to avoid false negative diagnosis [...]. This is unfortunate, since false negative diagnosis usually carry a heavier penalty, in real life.*

Turney, too, attributes this phenomenon to the distribution of positive and negative examples in the training set. In this context, our aim is to modify the fitness measure as to eliminate such undesirable asymmetries.

Last, but not least, previous *ICET* papers focus almost entirely on test costs and lack a comprehensive analysis of the misclassification costs component. This paper tries to fill this gap, by providing a comparative analysis with some of the classic cost-sensitive techniques, such as *MetaCost* and *AdaBoost*.

Our version of *ICET* was based on the implementation of the C4.5 algorithm, revision 8, provided by *Weka* (referred to as *J4.8*), and a general-purpose genetic algorithms library called *GGAT* (*General Genetic Algorithm Tool)*, both written in *java*.

Weka (*Waikato Environment for Knowledge Analysis*) is a data mining tool developed at the University of Waikato, New Zeeland [11]. The application is distributed under the GPN (Gnu Public License). It includes a wide variety of state-of-the-art algorithms and data processing tools and provides extensive support for the entire process of experimental data mining (input filtering, statistical evaluation of learning schemes, data visualization, preprocessing tools). Apart from an easy to use interface, Weka also comes with a command line interface, which enables the user to call the different learning schemes from the command line. This feature was particularly useful when invoking the modified decision tree learner for computing the fitness function in the genetic algorithm part of the application.

GGAT is a generic GA library, developed at the Brunel University, London. It uses a technique called *single population* for generating a new generation, which directly implements *elitism* (the best individuals of the current generation can survive unchanged in the next generation).

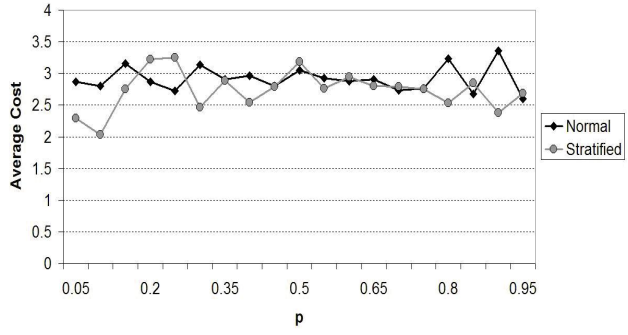

Fig. 1 - ICET average costs for the breast cancer dataset

Another prominent feature is the use of *ranking* in the fitness function estimation. The individuals in the population are ordered according to their fitness value, after which probabilities of selection are distributed evenly, according to their rank in the ordered population. Ranking can be a very effective mechanism for avoiding the premature convergence of the population, which can occur if the initial pool has some individuals which dominate, having a significantly better fitness than the others.

The information gain function of J4.8 algorithm was modified, similarly to the implementation presented in [9], to consider the bias parameter associated to each attribute, as specified by equation (1).

The modified *ICET* algorithm was implemented within the framework provided by GGAT. For each individual, the n + 2 chromosomes were defined (n being the number of attributes in the data set, while the other two correspond to parameters w and CF); each chromosome is represented as a 14 bits binary string, encoded in Gray. The population size is 50 individuals. The *roulette wheel* technique was used for parent selection; as recombination techniques, we have employed *single point random mutation* with mutation rate 0.2, and *multipoint crossover*, with 4 randomly select crossover points.

The algorithm is run for 1000 fitness evaluation steps or until convergence. Due to the fact that a new generation is evolved using single population, the final result yielded by the procedure is the best individual over the entire run, which makes the decision on when to stop the evolution less critical. More than that, experiments show that usually the best individual does not change significantly after 800 steps.

A. Symmetry Through Stratification

As we have mentioned before, it is believed that the asymmetry in the evaluated costs for two-class problems, as the proportion of false positives and false negatives misclassification costs varies, is owed to the small number of negative examples in most datasets. If the assumption is true, the problem could be eliminated by altering the distribution of the training set, either by oversampling, or by undersampling. This hypothesis was tested by performing an evaluation of the ICET results on the Wisconsin breast cancer dataset.

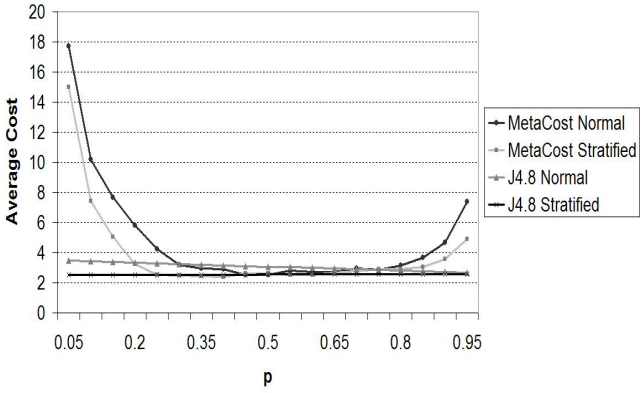

Fig. 2 – Improved average cost for the stratified Wisconsin dataset

This particular problem was selected as being one of the largest two-class datasets presented in the literature.

Since the algorithm involves a large heuristic component, the *ICET* evaluation procedure assumes averaging the costs over 10 runs. Each run uses a pair of randomly generated training-testing sets, in the proportion 70% - 30%; the same proportion is used when separating the training set into a component used for training and one for evaluating each individual (in the fitness function). Test costs are set to 1 during individual evaluation, in order to avoid overfitting or degenerate solutions. Since the misclassification costs are the one studied by the procedure, test costs are ignored during the evaluation of the final results.

For the stratified dataset, the negative class is increased to the size of the positive class, by repeating examples in the initial set, selected at random, with a uniform distribution. Oversampling is preferred, despite of an increase in computation time, due to the fact that the alternate solution involves some information loss. Undersampling could be selected in the case of extremely large databases, for practical reasons. In this situation, oversampling is no longer feasible, as the time required for the learning phase on the extended training set becomes prohibitive.

The misclassification cost matrix used for this analysis has the form:

$$C = 100 \cdot \begin{pmatrix} 0 & p \\ 1-p & 0 \end{pmatrix} \qquad (2)$$

where p is varied with a 0.05 increment.

The results of the experiment are presented in Fig. 1. We observe a small decrease in misclassification costs for the stratified case throughout the parameter space. This reduction is visible especially at the margins, when costs become more unbalanced. Particularly in the left side, we notice a significant reduction in the total cost for expensive rare examples, which was the actual goal of the procedure.

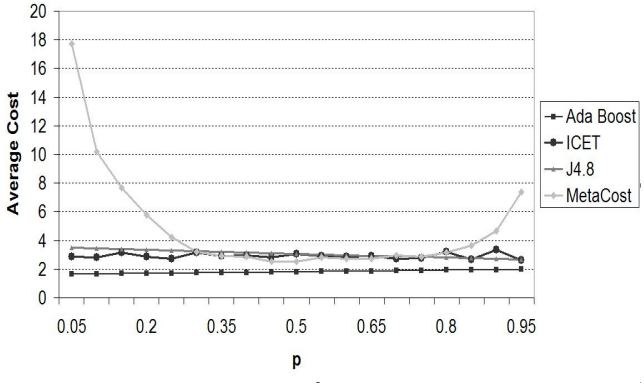

Fig. 3 – A comparison of average misclassification costs on the Wisconsin dataset

Starting from the assumption that the stratification technique may be applicable to other cost-sensitive classifiers, we have repeated the procedure on the Weka implementation of MetaCost, using J4.8 as base classifier. J4.8 was also considered in the analysis, as baseline estimate.

The results for the second set of tests are presented in Fig. 2. We observe that MetaCost yields significant costs, as the cost matrix drifts from the balanced case, a characteristic which has been described previously. Another important observation is related to the fact that the cost characteristic in the case of J4.8 is almost horizontal. This could give an explanation of the way stratification affects the general ICET behavior, by making it insensitive to the particular form of the cost matrix. Most importantly, we notice a general reduction in the average costs, especially at the margins of the domain considered. We conclude that our stratification technique could be also used for improving the cost characteristic of MetaCost. Further testing is required before formulating more general results.

B. Comparing Misclassification Costs

The procedure employed when comparing misclassification costs is similar to that described in the previous section. Again, the Wisconsin dataset was used, and misclassification costs were averaged on 10 randomly generated training/test sets. For all the tests described in this section, the test costs are not considered in the evaluation, in order to isolate the misclassification component and eliminate any bias.

As illustrated by Fig. 3, MetaCost yields the poorest results. ICET performs slightly better than J4.8, while the smallest costs are obtained for AdaBoost, using J4.8 as base classifier. The improved performance is related to the different approaches taken when searching for the solution. If ICET uses heuristic search, AdaBoost implements a procedure that is guaranteed to converge to minimum training error, while the ensemble voting reduces the risk of overfitting.

TABLE I.
MISCLASSIFICATION COST MATRIX FOR
BUPA LIVER DISORDER DATASET

Class	less than 3	more than 3
less than 3	0	5
more than 3	15	0

TABLE II.
MISCLASSIFICATION COST MATRIX FOR
THE CLEVELAND HEART DISEASE DATASET

Class	0	1	2	3	4
0	0	10	20	30	40
1	50	0	10	20	30
2	100	50	0	10	20
3	150	100	50	0	10
4	200	150	100	50	0

TABLE III.
MISCLASSIFICATION COST MATRIX FOR THE THYROID DATASET

Class	3	2	1
3	0	5	7
2	12	0	5
1	20	12	0

However, the approach cannot take into account test costs, which should make it perform worse on problems involving both types of costs.

C. Total Costs Analysis

When estimating the performance of the various algorithms presented, we have considered three problems from the UCI repository. All datasets involve medical problems: Bupa liver disorders, heart disease Cleveland and thyroid. For the first dataset, we have used the same modified set as in [9]. Also, the test costs estimates are taken from the previously mentioned study. The misclassification costs values were more difficult to estimate, due to the fact that they measure the risks of misdiagnosis, which do not have a clear monetary equivalent. These values are set empirically, assigning higher penalty for undiagnosed disease and keeping the order of magnitude as to balance the two cost components (the actual values are displayed in tables I, II and III).

As anticipated, ICET significantly outperforms all other algorithms, being the only one built for optimizing total costs (Fig. 4). Surprisingly, our implementation performs quite well on the heart disease dataset, where the initial algorithm obtained poorer results. This improvement is probably owed to the alterations made to the genetic algorithm, which increase population variability and extend the ICET heuristic search. The cost reduction is relatively small in the Thyroid dataset, compared to the others, but is quite large for the two cases, supporting the conclusion that ICET is the best algorithm for problems involving complex costs.

V. CONCLUSIONS

The study presented here makes several improvements to the initial *ICET* implementation in [9], mostly involving the evolutionary component of the algorithm. By introducing elitism, increasing the search variability factor, and extending the number of iterations, we manage to outperform other cost-sensitive algorithms, even for datasets on which the initial implementation yielded modest results.

Another important result is the reduction of misclassification costs for unbalanced cost matrices, by adjusting the training set class distribution. This cost reduction is significant, especially in the case of rare expensive cases, which is often critical in practice. More than that, the advances induced by our stratification method are shown to be maintained for other algorithms, such as *MetaCost*.

Although, when analyzing the misclassification costs, *ICET* is outperformed by *AdaBoost.M1*, it still manages to yield better results than other algorithms built exclusively for optimizing misclassification costs.

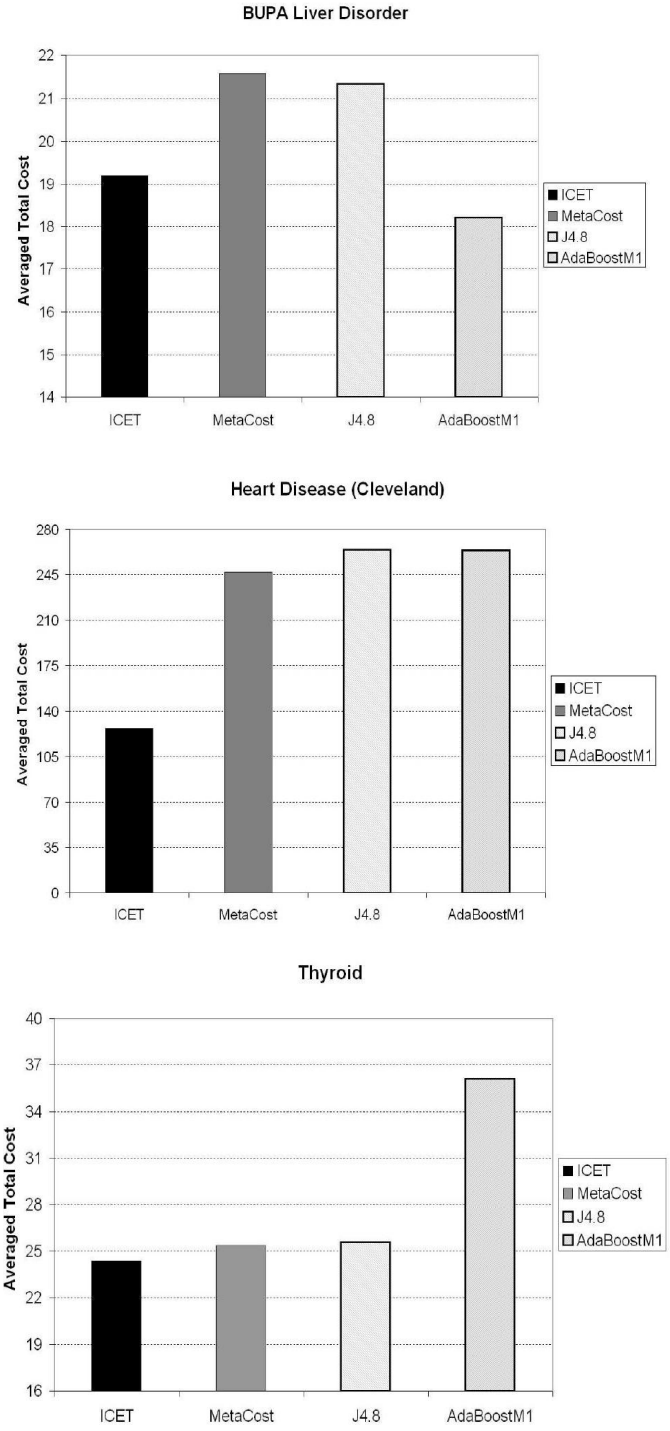

Fig. 4 – A comparison of average total costs on various datasets

The situation changes for complex costs problems, where *ICET* performs significantly better than all the other implementations considered. In conclusion, *ICET* appears to be the best solution for the medical diagnosis and prognosis problem, from the various alternatives analyzed.

Several other adjustments to the basic *ICET* algorithm could be examined in the future. Firstly, the procedure employed during the classification of an instance could consider both the class distribution in the corresponding leaf and the associated costs. On a different line of study, there appears to be a need for a more comprehensive analysis on the impact of varying the GA parameters, as we have seen that such changes can lead to significant performance improvements.

REFERENCES

[1] P. Domingos. Metacost: A general method for making classifiers cost- sensitive. *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining*, 1991.

[2] W. Fan, S. Stolfo, J. Zhang, and P. Chan. AdaCost: Misclassification cost- sensitive boosting. *Proceedings of the 16th International Conference on Machine Learning*, pages 97–105, 2000.

[3] Y. Freund and R. Schapire. A decision-theoretic generalization of on- line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[4] J. Li, X. Li, and X. Yao. Cost- sensitive classification with genetic programming. *Proceedings of the 2005 Congress on Evolutionary Computation*, 3:2114–2121, 2005.

[5] J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[6] J. Quinlan. Boosting first- order learning. *Proceedings of the 7th International Workshop on Algorithmic Learning Theory*, 1160:143–155, 1996.

[7] S. Sheng and C. Ling. Hybrid cost- sensitive decision tree. *PKDD*, pages 274–284, 2005.

[8] S. Sheng, C. Ling, and Q. Yang. Simple test strategies for cost-sensitive decision trees. *ECML*, pages 365–376, 2005.

[9] P. Turney. Cost sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research*, (2):369–409, 1995.

[10] P. Turney. Types of cost in inductive concept learning. *Proceedings of the Workshop on Cost-Sensitive Learning, 7th International Conference on Machine Learning*, 2000.

[11] I. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques, 2nd ed.* Morgan Kaufmann, 2005.