Evolutionary Induction of Cost-Sensitive Decision Trees

Marek Krętowski and Marek Grześ

Faculty of Computer Science, Białystok Technical University
Wiejska 45a, 15-351 Białystok, Poland
{mkret, marekg}@ii.pb.bialystok.pl

Abstract. In the paper, a new method for cost-sensitive learning of decision trees is proposed. Our approach consists in extending the existing evolutionary algorithm (EA) for global induction of decision trees. In contrast to the classical top-down methods, our system searches for the whole tree at the moment. We propose a new fitness function which allows the algorithm to minimize expected cost of classification defined as a sum of misclassification cost and cost of the tests. The remaining components of EA i.e. the representation of solutions and the specialized genetic search operators are not changed. The proposed method is experimentally validated and preliminary results show that the global approach is able to effectively induce cost-sensitive decision trees.

1 Introduction

In many data-mining applications, especially in medicine or business, the traditional minimization of classification errors is not the most adequate scenario. Particular decisions have often significantly different impact on the overall result. It is especially evident in medical diagnosis, where misclassifying an ill person as a healthy one is generally much more dangerous (and costly) than an inverse error. In such situations different misclassification costs associated with decisions are used to compensate the problem. Additionally, the cost of decision making (i.e. the cost of used features) can be also taken into account. This is usually obtained by preferring such a classifier which gives equally accurate predictions with a lower cost, calculated as a sum of costs of the performed tests. In medical domain, similar diagnostic accuracy can be sometimes obtained with very expensive tests as well as with simple and cheap examinations.

Cost-sensitive classification is the term which encompasses all types of learning where cost is considered [12]. However most of existing cost-sensitive systems consider only one cost type. There are two main approaches to making a classifier cost sensitive. In the first group classical classifiers are converted into cost-sensitive ones. In the context of decision tree learning it encompasses mainly changing the splitting criteria and/or adopting pruning techniques for incorporating misclassification cost (e.g. [1,2,4]) or cost of tests (e.g. in *EG2* [9]). Another method of misclassification cost minimization is proposed in [10], where instance-weighting is applied, but the method requires the conversion of the cost matrix into the cost vector. Among systems, which take into account both cost

F. Esposito et al. (Eds.): ISMIS 2006, LNAI 4203, pp. 121–126, 2006.

types the two approaches should be mentioned: *Inexpensive Classification with Expensive Tests - ICET* [11] and *Internal Node Strategy - INS* [7,14]. *ICET* uses the standard genetic algorithm to evolve a population of biases for modified C4.5 and *INS* is based on a *total cost* criterion for nominal attributes and two-class problems. The second group includes general methods for making an arbitrary classifier cost-sensitive. *MetaCost* [3] is based on wrapping a meta-learning stage around the error-based classifier. Another method proposed by Zadrozny *et al.* [13] uses cost-proportionate rejection sampling and ensemble aggregation. However, both mentioned approaches incorporate only misclassification costs.

In the paper, a cost-sensitive extension of our EA-based system [6] for decision tree learning is presented. In contrast to the typical top-down induction, the global method searches simultaneously for both the optimal tree structure and all tests in internal nodes. Such an approach is computationally more complex but it often allows avoiding sub-optimal solutions imposed by greedy techniques. As a result accurate and more compact classifiers are obtained. It should be underlined that necessary adaptation of our error-based system to incorporate both misclassification and test costs are limited almost only to the fitness function. Apart from slightly modified method for assigning the class labels to leaves all remaining elements of the original algorithm do not have to be changed.

The rest of the paper is organized as follows. In the next section, an evolutionary algorithm for global induction of decision trees is briefly described. In section 4 the cost-sensitive extension of our system is proposed. Preliminary experimental validation of the proposed approach is presented in section 5. The paper is concluded in the last section.

2 Global Induction of Decision Trees

In this section, main ideas of our EA-based system called *Global Decision Tree* (*GDT*) are briefly presented. For more detailed description please refer to [6].

Representation and Initialization. Decision trees are represented in their actual form as univariate trees where any test in an internal node concerns only one attribute. In case of a nominal attribute at least one value is associated with each branch (inner disjunction). For a continuous-valued feature typical inequality tests with boundary thresholds as potential splits are considered. A boundary threshold for the given attribute is defined as a midpoint between the successive pair of examples from different classes in the sequence sorted by the increasing value of the attribute. All boundary thresholds are calculated before starting the evolutionary induction.

Individuals in the initial population are generated using the classical top-down algorithm, but tests are chosen in a dipolar-like way [5].

Genetic Operators. There are two specialized genetic operators corresponding to classical mutation and crossover. When crossover is applied the randomly chosen parts (i.e. sub-trees or only tests) of two individuals are swapped. There are a few variants of this exchange.

Mutation-like operator is a more complex operator and is applied to a given tree node. Possible modifications depend on the node type (i.e. whether it is a leaf node or an internal node). For a non-terminal node a few possibilities exist: *i)* a completely new test can be drawn, *ii)* existing test can be altered by shifting the splitting threshold (continuous-valued feature) or re-grouping feature values (nominal feature), *iii)* the test can be replaced by another test from the tree and finally *iv)* the node can be transformed into a leaf. Modifying a leaf node makes sense only if it contains feature vectors form different classes. The leaf is transformed into an internal node and a new test is randomly chosen. The search for effective tests can be recursively repeated for all descendants.

The application of any genetic operator can result in a necessity of relocation of the input vectors between parts of the tree rooted in the modified node. Additionally local maximization of the fitness function is performed by pruning lower parts of the subtree on condition it improves the value of the fitness.

Error-Based Fitness Function. The goal of any classification system is correct prediction of class labels of new objects, however such a target function cannot be directly defined. Instead the accuracy on the training data is often used, but their direct optimization leads to the over-fitting problem. In classical systems this problem is usually mitigated by post-pruning techniques. In our approach a complexity term is introduced into the fitness function preventing the over-specialization. The fitness function, which is maximized, has the following form:

$$Fitness(T) = Q_{Reclass}(T) - \alpha \cdot (S(T) - 1), \tag{1}$$

where $Q_{Reclass}(T)$ is the re-classification quality, $S(T)$ is the size of the tree T expressed as the number of nodes and α is a relative importance of the complexity term (default value is 0.005) and a user supplied parameter.

3 Cost-Sensitive Extension

There are only two modifications (new fitness function and class label assignment) necessary for incorporating misclassification and test costs into *GDT*.

Preliminaries. Let learning set $E = \{e_1, e_2, \ldots, e_M\}$ consist of M examples. Each example $e \in E$ is described by N attributes (features) $A_1, A_2, \ldots, A_N$ and the corresponding feature costs are denoted by $C_1, C_2, \ldots, C_N$ respectively. Additionally, a decision (class label) associated with an object e is represented by $d(e) \in D$. The set of all examples with the same decision $d_k \in D$ is denoted by $D_k = \{e \in E : d(e) = d_k\}$ and the class assigned (predicted) by the tree T to the example e is denoted by $T(e)$. Let $Cost(d_i, d_j) \geq 0$ be the cost of misclassifying an object from the class d_j as belonging to the class d_i. We assume that the cost of the correct decision is equal zero i.e., $Cost(d_i, d_i) = 0$ for all d_i.

Cost-Sensitive Fitness Function. Performance of an error-based tree is judged by the classification accuracy whereas a cost-sensitive tree is assessed by the average cost, which is a sum of the average misclassification cost and

the average test costs. This suggests replacing in the fitness function the reclassifcation accuracy with the expected cost of classification. The expected cost in general is not limited, but for a given dataset, cost matrix and attribute costs, the maximum misclassification and test costs can be calculated. As a result the normalized cost can be easily obtained and fits into $[0, 1]$ range.

First, the misclassification cost $MCost(T)$ of the tree T is estimated on E:

$$MCost(T) = \frac{1}{M} \cdot \sum_{e \in E} Cost(T(e), d(e)). \tag{2}$$

The maximal misclassification cost for a given dataset and a cost matrix is equal:

$$MaxMC = \frac{1}{M} \cdot \sum_{d_k \in D} |D_k| \cdot \max_{i \neq k} Cost(d_i, d_k). \tag{3}$$

By dividing $MCost(T)$ by $MaxMC$ one can obtain the *normalized misclassification cost*, which is equal 0 for the perfect classification and 1 in the worst possible case.

Similar approach can be applied to calculate *normalized average tests cost* of tree T induced from training data E. Let $A(e)$ denote the set of all attributes used in tests (on the path from the root node to the terminal leaf reached by e) necessary to classify an object e. Hence, tests cost $TC(e)$ of classifying an object e by tree T is equal:

$$TC(e) = \sum_{A_i \in A(e)} C_i. \tag{4}$$

It should be noted that renewed use of any feature in the following tests does not increase $TC(e)$. The average test cost $TCost(T)$ can be defined as follows:

$$TCost(T) = \frac{1}{M} \cdot \sum_{e \in E} TC(e). \tag{5}$$

The maximal value of this cost for the given learning set is equal:

$$MaxTC = \frac{1}{M} \cdot \sum_{i=1}^{N} C_i, \tag{6}$$

where all features $A_i \in A$ are necessary for making prediction for any object. Normalized average test cost is calculated by dividing $TCost(T)$ by $MaxTC$. It is equal zero for the tree composed only of one leaf and is equal 1 when in every path of the tree all features are used in tests.

Finally, the fitness function, which is minimized, is defined as follows:

$$Fitness(T) = \frac{MCost(T) + TCost(T)}{MaxMC + MaxTC} + \alpha \cdot (S(T) - 1). \tag{7}$$

Class Labels for Leaves. In standard error-based decision trees the class labels are assigned to leaves by using the majority rule based on training objects which reached a leaf-node. In cost-sensitive case the class labels for leaves are chosen to minimize the misclassification cost in each leaf.

4 Experimental Results

In this section *GDT* is compared with nonnative implementations of *ICET*, *INS* and *EG2* on four datasets that were investigated in [11]. Costs of attributes that are provided for these datasets were used and cost matrices were prepared in the following way. All values of the misclassification cost are at least equal to the sum of features cost. In the subsequent experiments presented in Table 1, penalty for misclassifying less frequent class had been increasing. Ratios of such costs were: 1.5, 2, 3 and F_{LF}/F_{MF} where F_{LF} and F_{MF} are frequencies of less and more frequent class respectively.

It can be observed that *GDT* is significantly better on *heart* data both in terms of the tree size and the average cost on the test data. This dataset is a good example of the situation when the top-down heuristic is trapped into a local minimum and the method working in a global manner is able to find more optimal structure of the tree. Another dataset in which top-down algorithms found also relatively big trees is the *pima* dataset. In this case *GDT* finds smaller trees but without improvement in terms of cost, which means that the system was able to eliminate repeated tests on previously used features. As for *bupa* and *hepatitis* datasets the results of all the algorithms except *ICET* are comparable.

Evolutionary algorithms are considered to be as good as its fitness function. These statement is in some way observed in *bupa* dataset. In this case *GDT* found very small trees which are identical in each run (standard deviation is zero). It shows that the algorithm is able to find small and relatively good result that is very stable. But on the other hand this result is not optimal. Comparisons with other algorithms show that there is still place for improvement (e.g. by tuning

Table 1. The results with different cost matrices

Dataset		*GDT*		"*ICET*"		"*EG2*"		"*INC*"	
		Size	Cost	Size	Cost	Size	Cost	Size	Cost
bupa	40/55	3.0	18.96±0.0	52.7	47.37±2.2	4	18.73	4	18.74
	40/60	3.0	19.13±0.0	52.6	48.42±2.3	5	19.56	5	19.56
	40/80	3.0	19.83±0.0	53.2	51.50±2.5	3	19.82	3	19.83
	40/120	3.0	21.22±0.0	52.1	61.86±3.3	1	22.96	1	22.96
heart	600/704	7.9	142.61±7.6	48.9	306.11±19.4	16	151.94	17	165.12
	600/900	5.8	163.25±5.5	44.8	371.58±49.8	20	187.79	20	196.82
	600/1200	5.6	199.27±27.1	46.4	382.10±40.7	24	199.56	20	227.13
	600/1800	4.1	257.9±8.1	48.3	478.74±41.9	20	239.67	20	239.67
hepatitis	50/262	6.8	30.83±4.1	5.8	34.02±4.3	6	32.11	6	30.9
	50/75	7.5	13.15±3.1	5.8	18.53±0.8	5	10.05	5	10.06
	50/100	7.3	15.34±3.2	5.7	21.98±3.1	7	7.77	5	12.84
	50/150	7.1	20.42±3.7	5.8	25.09±3.1	6	27.51	5	18.39
pima	50/92	4.3	19.92±0.6	83.9	22.86±0.7	38	18.21	24	20.05
	50/75	3.4	17.58±0.2	84.5	20.63±0.9	24	16.61	28	17.94
	50/100	4.6	21.23±0.4	83.9	24.4±1.0	31	19.18	26	22.5
	50/150	5.0	23.77±0.5	84.9	31.59±1.1	30	23.88	24	24.41

user specified α parameter). Further experiments showed that it is possible to obtain better results (e.g. for $\alpha = 0.001$ on *bupa* 40/40 we gain the best result 17.03). It means that fitness function can be further investigated and it might improve the overall performance.

5 Conclusion

In this paper, our global method for decision tree induction is extended to handle two types of cost: misclassification cost and cost of tests. The necessary modifications of the evolutionary algorithm encompass mainly the new fitness function. Even preliminary results of experimental validation show that our system is able to generate competitive cost-sensitive classifiers.

Acknowledgments. This work was supported by the grant W/WI/5/05 from Białystok Technical University. The authors thank M. Czołombitko for providing us with implementation of other cost-sensitive classifiers.

References

1. Bradford, J., Kunz, C., Kohavi, R., Brunk, C., Brodley, C.E.: Pruning decision trees with misclassification costs. In *Proc. of ECML'98.* Springer (1998) 131–136.
2. Breiman, L., Friedman, J., Olshen, R., Stone C.: *Classification and Regression Trees.* Wadsworth Int. Group (1984).
3. Domingos, P.: MetaCost: A general method for making classifiers cost-sensitive. In *Proc. of KDD'99.*, ACM Press (1999) 155–164.
4. Knoll U., Nakhaeizadeh G., Tausend B.: Cost-sensitive pruning of decision trees. In *Proc. of ECML'94.*, Springer LNCS 784 (1994) 383–386.
5. Krętowski, M.: An evolutionary algorithm for oblique decision tree induction, In: *Proc. of ICAISC'04.*, Springer LNCS 3070, (2004) 432–437.
6. Krętowski, M., Grześ, M.: Global learning of decision trees by an evolutionary algorithm, In: *Information Processing and Security Sys.*, Springer, (2005) 401–410.
7. Ling, C., Yang, Q., Wang, J., Zhang, S.: Decision trees with minimal costs, In: *Proc. of ICML'04.*, ACM Press (2004), Article No. 69.
8. Margineantu, D., Dietterich, T.: Bootstrap methods for the cost-sensitive evaluation of classifiers, In *Proc. of ICML'2000.*, Morgan Kaufmann (2000) 583–590.
9. Nunez, M.: The use of background knowledge in decision tree induction, *Machine Learning* **6**, (1991) 231–250.
10. Ting, K.: An instance-weighting method to induce cost-sensitive trees. *IEEE TKDE* **14**(3), (2002) 659–665.
11. Turney, P.: Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *J. of Artif. Intel. Res.* 2, (1995) 369–409.
12. Turney, P.: Types of cost in inductive concept learning. In *Proc. of ICML'2000 Workshop on Cost-Sensitive Learning.* Stanford, CA (2000).
13. Zadrozny, B., Langford, J., Abe. N.: Cost-sensitive learning by cost-proportionate example weighting concept learning. In *Proc. of ICDM'03.* IEEE Press (2003).
14. Zhang S., Qin, Z., Ling, C. Sheng, S.: Missing is usefull: Missing values in cost-sensitive decision trees. *IEEE TKDE* **17**(12), (2005) 1689–1693.