Cost-Sensitive Classification: An Improvement to the Fitness Function of ICET

Bayram Boyraz

Bilkent University, Department of Computer Science,
06800 Ankara, Turkey
bboyraz@cs.bilkent.edu.tr
http://www.cs.bilkent.edu.tr/~bboyraz/

Abstract. This paper proposes an improvement to the fitness function of ICET (Inexpensive Classification with Expensive Tests – pronounced "iced tea") algorithm that is for cost-sensitive classification. ICET uses a genetic algorithm to evolve population of biases for a decision tree induction algorithm. The fitness function of ICET is the average cost of classification when using the decision tree. ICET can handle both test costs and misclassification error costs. In this paper only the improvement and its justification are suggested but no experimental results are presented because of time limitations.

Introduction

The motivation to bother with the test costs in machine learning algorithms is that the cost of a test can be more than the expected benefit of the test. Consider the situation depicted in Figure 1. You are at position A and you want to go to the position B as early as possible. There is a crossroad on the way and you can reach to B using any of the roads. Assume that you have been informed that the time difference between the shortest road and the longest road is only 5 minutes and there is a way to learn which road is the shortest. You need to perform a test to learn it. But you are aware that performing this test will take at least 20 minutes. Clearly, in such a case you will never do the test since its expected benefit is less than its cost.

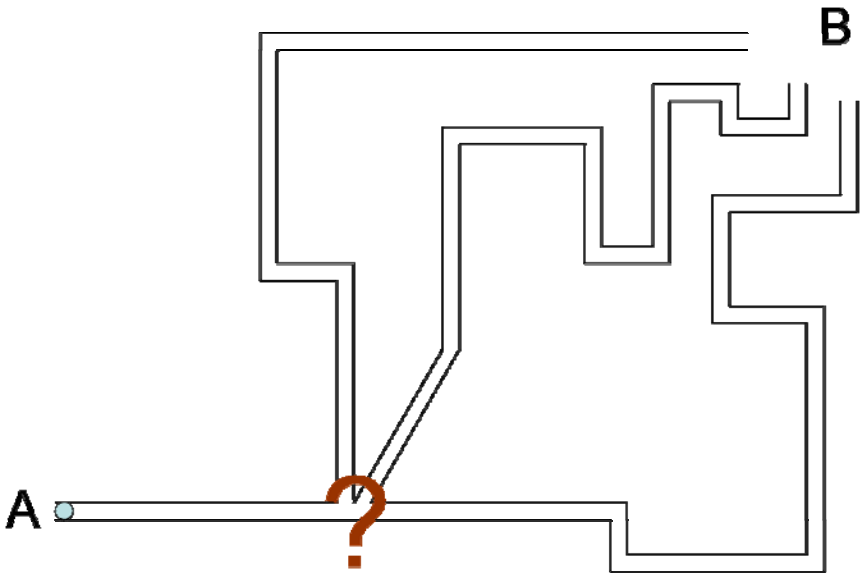


Figure 1

The same principle can be used for machine learning algorithms also. If cost of a test is higher than the maximum cost of misclassification error, then it does not worth to perform the test.

Cost Sensitive Classification

It is well explained by (Turney, 1995) that the natural form of knowledge representation for classification with expensive tests is a decision tree. Organization of the decision tree will be in the following way: Unless being a leaf node, each node will represent a test result and the leaf nodes will be the classification that is determined by the decision tree.

To specify the costs of tests we will simply list each test with its corresponding cost in a table. (See Figure 2 for an example of such a table)

Test	Cost
Alpha	$5.00
Beta	$10.00
Delta	$7.00
Epsilon	$10.00

Figure 2 – An example table that specifies cost of tests for a certain case.

Also to specify costs of misclassification errors we will use a *classification cost matrix*. A classification cost matrix is a *c*x*c* matrix where *c* is the number of classes. Each $C_{i,j}$ entry of the matrix will represent the cost of classifying the case as belonging to the class *i*, where it actually belongs to class *j*. (See Figure 3 for an example classification cost matrix where there are two classes, namely *class 0* and *class 1*)

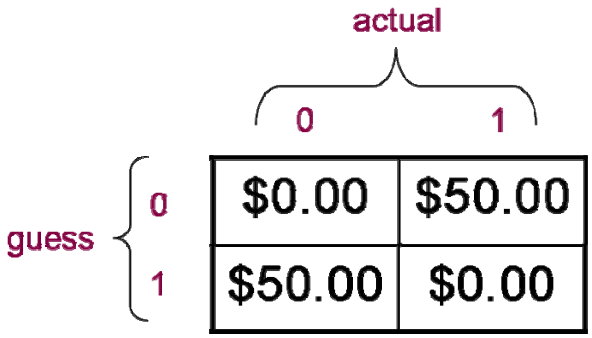


Figure 3 – An example classification cost matrix

Calculating the Average Cost of Classification

Before calculating the average cost of classification for a decision tree, we divide the dataset into training set and test set. Next, we calculate the average cost by calculating the cost for each case in the test set and dividing the total by the number of cases in the test set again.

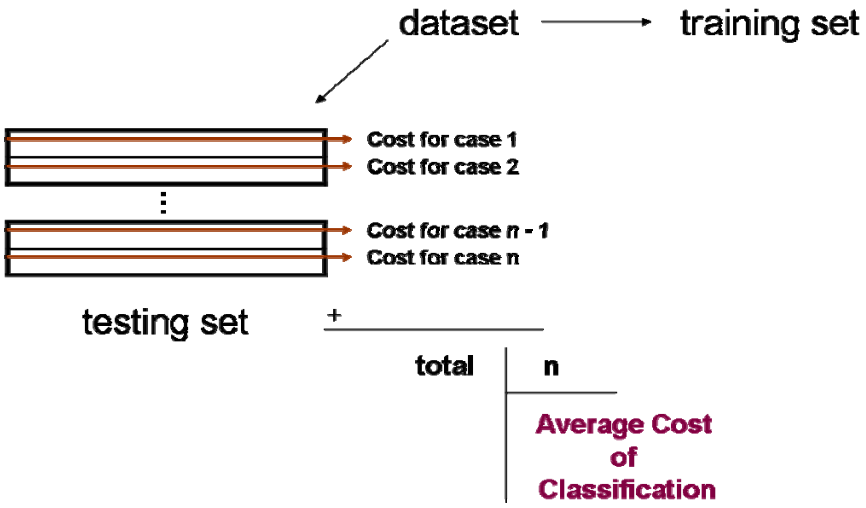


Figure 4 – The process of calculating the average cost for a decision tree

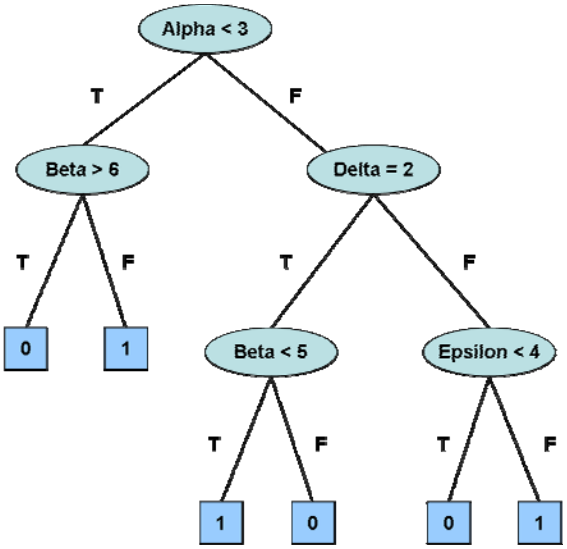
To calculate the cost for a single case we follow its path down in the decision tree and we add up the cost of each test on the way. For example, consider the case in Figure 5a and the tree in Figure 5b. Using the test results in Figure 5a, clearly we perform Alpha, Delta, and Beta tests. Using the cost table in Figure 2 the total cost of tests will be $22. And then the decision tree classifies the case to the *class 1*, whereas the actual class is *class 0*. So we add the misclassification error cost using the classification error matrix in Figure 3. Finally we will have computed the total cost for this particular case as $72.

This is the core of the method to calculate average cost of classification when using a decision tree. There are two additional elements to the method, for handling *conditional test costs* and *delayed test results*.

Some tests may share a common cost. For example, if we want to find the amount of a certain protein in one's blood, and also amount of red blood cells of the same person, we should not calculate the cost of collecting blood twice, we can use the blood that we have already.

Sometimes we cannot learn the result of a test immediately. These tests are called "*delayed tests*". If we have a delayed test, then we cannot know which test to perform. In this case, to calculate the average cost of classification Turney suggests adding costs of all tests in the subtree that is rooted at that delayed test.

	result
Alpha	7
Beta	3
Delta	2
Epsilon	6
Class	0



(a) (b)

Figure 5 – An example case in test data and an example of decision tree

Now we add two fields to our cost list table that is shown in Figure 2, one for grouping the tests having common cost and one for specifying whether a test is delayed or not. A sample cost list table is present in Figure 6.

Test	Group	Cost	Delayed
Alpha		$5.00	No
Beta		$10.00	No
Delta	A	$7.00 if first in group $5.00 otherwise	Yes
Epsilon	A	$10.00 if first in group $8.00 otherwise	Yes

Figure 6

If we recalculate the cost using the sample case and decision tree in Figure 5, the cost will increase since Delta is a delayed test, and so we need to pay for all tests in the subtree rooted at Delta. With this, new cost for that particular case will be \$72 + "cost of Epsilon". Notice that Delta and Epsilon share a common cost of \$2. So we will pay \$8 for the Epsilon test, since we have already done a test (Delta) that belongs to the same group with Epsilon.

We have paid for all tests in the subtree that are rooted at the delayed test, because we did not know the result of the test. Instead of paying the exact costs of all tests, we suggest the following: Assume that we have a delayed test. We can consider each child of this test in the decision tree as the possible results of the test. By making use of the results of previous tests we can assign a probability for each possible result of the delayed test. Later we will sum costs of all tests that are rooted at each possible result of the delayed tests. Finally we will add all sums after multiplying them with the corresponding probabilities.

For example, assume that the result of "Delta = 2" will be **F**(alse) with 0.2 probability, and it will be **T**(rue) with 0.8 probability if we consider that result of "Alpha < 3" is **F**. So we will multiply cost of Epsilon, which is \$8, with 0.2 and cost of Beta, which is \$10 with 0.8 and pay for the sum, which is:

$$\$ (0.2*8 + 0.8*10) = \$9.6$$

So the cost for the sample case will be \$71.6 instead of \$80. Consider the example in Figure 7. Assume that the total cost of all the tests in the subtree rooted at Y is \$500 and the total cost of all the tests in the subtree rooted at Z is \$100, and X is a delayed test that we need to do. By considering the results of tests before X (which are not shown in the figure), assume that the probability that $X = 2$ is 0.15, whereas the probability of $X \neq 2$ is 0.85. While calculating the cost of the tree adding the exact total cost of all the tests in the subtree rooted at Y will distort the truth, because the probability that to perform the test Y is so small compared to the probability to perform Z. So, when calculating the average cost of a decision tree it seems wise to multiply each cost of subtree with the probabilistic value that we will use that subtree, if we do not know the result of the current test.

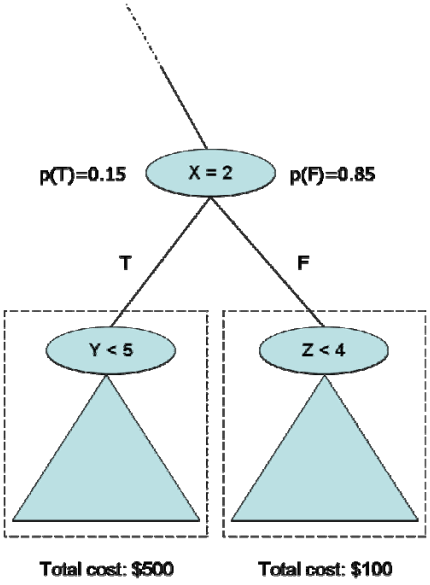


Figure 7

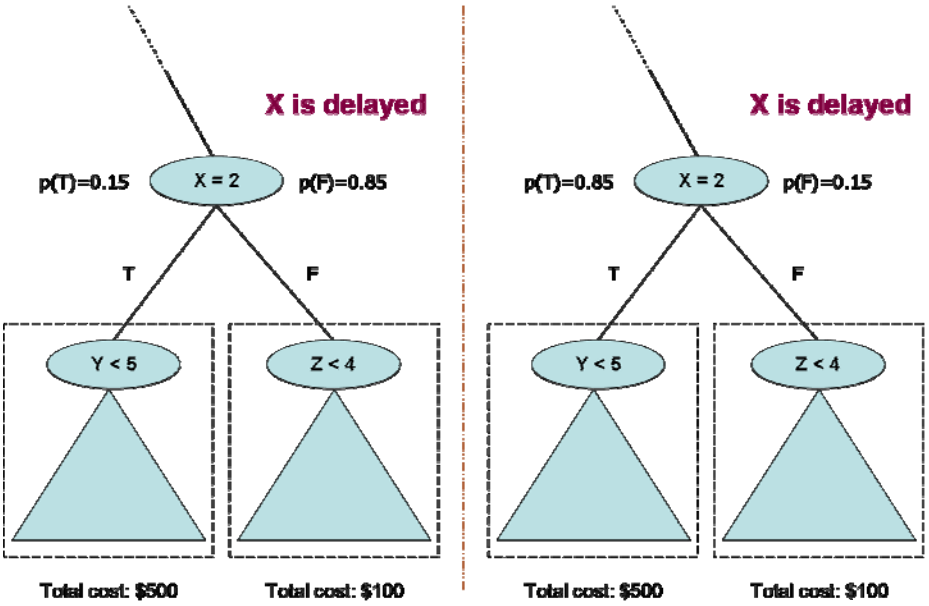


Figure 8 – The procedure proposed by Turney, treats both of the subtrees above in the same way.

Future Work

For the future work, I am planning to show the correctness of my improvement to the ICET fitness function experimentally.

One way to apply the improvement suggested in this paper is to consider each result of a delayed test as a class into which we are trying to put a case using the previous results of the tests. We plan to use different classification algorithms and see the probabilities that they assign for each result and compare their accuracy.

References:

1. P. Turney, "Types of Cost in Inductive Concept Learning," *Proceedings of the Cost-Sensitive Learning Workshop*, pp. 15-21, 2000.

2. P. Turney, "Cost-Sensitive Classification: Empirical Evaluation of a Hybrid Genetic Decision Tree Induction Algorithm," *Journal of Artificial Intelligence Research*, vol. 2, pp. 369-409, 1995.