Tree retraining in the decision tree learning algorithm

S A Mitrofanov and E S Semenkin
Reshetnev Siberian State University of Science and Technology
31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660037, Russian Federation

E-mail: sergeimitrofanov95@gmail.com

Abstract. Decision trees belong to the most effective classification methods. The main advantage of decision trees is a simple and user-friendly interpretation of the results obtained. But despite its well-known advantages the method has some disadvantages as well. One of them is that decision tree learning algorithm build an “almost optimal” tree. The paper considers the way to improve the efficiency of decision trees. The paper proposes a modification of decision tree learning algorithms by retraining the part of tree at every node training. The classification problems were solved to compare standard decision tree learning algorithms with the modified ones. Algorithm efficiency refers to the percentage of correctly classified test sample objects. Statistical analysis based on Student's t-test was carried out to compare the efficiency of the algorithms.

1. Introduction
Intelligent data analysis technologies include solving the classification problem. This direction is one of the most important. Many decision support methods are designed to solve classification problems. One of the best practices in this area is decision trees. Decision trees are a method based on the use of various functions to separate the original data set, in particular, simple threshold rules [1]. Decision tree results are very well interpreted for humans. However, decision trees are very difficult to optimize due to their discrete structure. This article considers a way to improve the efficiency of decision trees by changing the way of learning.

2. Decision trees
A decision tree is a binary tree in which there are two types of nodes: inner and leaf. Each inner node contains a function, and each leaf node contains a forecast [1]. In most cases, one-dimensional predicates are used that compare the value of one of the attributes with a threshold. However, there are also multidimensional predicates [2]. Multivariate predicates allow even more complex dividing surfaces to be constructed, but they are rarely used in practice, in part because they increase the tendency to retrain decision trees.

Decision tree learning algorithms have a set of settings and parameters like any other machine learning algorithms. Varying these settings and parameters provides a variety of decision tree learning algorithms. The specific method for constructing the decision tree is determined by:

1. Types of predicates at the vertices;
2. Quality functional \( Q(X, j, s) \);
3. Stop criterion;
4. Missing values processing method;
5. “Pruning” method.

Decision tree learning methods necessarily include the first three parameters, and only some algorithms include the last two. We have already mentioned the types of predicates, so we turn to the consideration of the quality functional \( Q(X, j, s) \).

To build a decision tree, you need to set the quality functional. Based on the selected quality functional, the sample is split at each node. We denote the set of objects that have fallen into some node as \( R_m \), and the objects that fall into the left and right subtrees, respectively, for a given predicate as \( R_l \) and \( R_r \). The following functional is used:

\[
Q(R_m, j, s) = H(R_m) - \frac{|R_l|}{|R_m|} H(R_l) - \frac{|R_r|}{|R_m|} H(R_r).
\]

Here \( H(R) \) is an information criterion that evaluates the quality of the target variable distribution among objects of the set \( R \). The smaller the diversity of the target variable is, the less the value of the information criterion should be and, accordingly, its value is minimized. The quality functional \( Q(R_m, j, s) \) where \( j \) is the attribute number and \( s \) is the threshold value is maximized at the same time. Later on when considering specific algorithms [3] we will indicate what specific information criteria are used for classification.

One can come up with a lot of stop criteria. We list some restrictions and criteria:

- Limiting the maximum depth of a decision tree.
- Limiting the minimum number of objects in a leaf.
- Limiting the maximum number of leaves in a decision tree.
- Stopping if all objects in the leaf belong to the same class.
- The requirement that the quality functional during splitting should be improved by at least \( s \) percent.

3. Decision tree learning algorithms

There are two main algorithms for learning decision trees - ID3 and CART. In this article, these algorithms were implemented and compared with modified ones [4]. These algorithms terminate if the observations of one class remain in the sheet or if a constraint is imposed on the depth of the decision tree. In this article, no depth limitation was imposed.

The main difference between these algorithms lies in information criteria. The ID3 algorithm uses the entropy criterion [3]:

\[
H(R) = - \sum_{k=1}^{K} p_k \log p_k
\]

where \( p_k \) is the fraction of objects of class \( k \) that have fallen into the node \( R \), \( K \) is the number of classes.

The CART algorithm uses the Gini criterion [3]:

\[
H(R) = \sum_{k=1}^{K} p_k(1 - p_k).
\]

To optimize the information criteria, standard algorithms use a full search over the initial data set. Since it is necessary to calculate the values of the information criterion for all attribute values for all observations of the training set, this process requires a significant amount of time. The decision tree learning process can be represented as a diagram in figure 1.
4. Decision tree learning approach

In a standard decision tree learning algorithm, all nodes are trained sequentially. With sequential training, we get a "greedy" algorithm. To weaken the "greed" of the algorithm, it is proposed to retrain the decision tree when training each node, that is, to optimize not only the threshold value at the node, but to optimize the threshold values in some part of the tree.

If we retrain the whole tree for each node, then we will worsen the classification for the nodes that are trained first. For example, if we first train the left node, optimizing it together with the parent node, then when training the right node, optimization together with the parent node will spoil the classification in the left node. Therefore, we introduce the concept of "main" and "additional" tree nodes.

At the stage of dividing the sample into left and right nodes, we will also define the "main" and "additional" nodes. The determination is made on the basis of information criteria. Since the greater the information criteria, the more diverse the sample in a node, then a node with a large variety should be given more attention during training. Thus, the node with the highest value of the information criteria will be the "main" one. Accordingly, the node with the lower value of the information criteria will be "additional".

For retraining, the thresholds will be selected as follows:

1) The threshold value of the considered node is added to the vector for training.
2) If the node in question is the "main" one, then step 3, otherwise the end of the vector formation for training.
3) The parent node becomes considered and go to step 1.

In figure 2, the red area marks the nodes whose threshold values will be combined for optimization.
However, if the optimization is performed only for the quality functional of the lower node, then changing the previous threshold values will fit the sample in the lower node so that it can be ideally divided. Therefore, we will minimize the sum of criteria for the information content of all nodes that are subject to optimization.

The selection of the attribute for the threshold value will be carried out using the Separation Measure method [5]. Optimization of the vector of threshold values will be performed by the method of differential evolution [6, 7]. Differential evolution has a number of settings, the choice of which determines the efficiency of its work. Categorical settings are self-adjusting according to the Population-Level Dynamic Probabilities algorithm [8-10]. The adaptation of numerical parameters is carried out by the Success History Adaptation method [11].

Figure 3 shows a diagram of the modified learning algorithm for the decision tree.

![Diagram of modified decision tree learning algorithm](image-url)
5. Solution of classification problems

To compare standard learning decision tree algorithms with modified algorithms, we took eight problems from the repository. These tasks are often used to analyze the efficiency of classification algorithms [12]:

1) Car type recognition.
2) Recognition of the type of urban landscape.
3) Determination of the variety of iris.
4) Diagnosis of Parkinson's disease.
5) Image recognition by segment.
6) Diagnosis of heart disease.
7) Determining the type of soil from satellite images.
8) Determination of biodegradable chemicals.

Comparison the efficiency of standard and modified algorithms is presented in the form of diagrams in figure 4–5. Efficiency refers to the percentage of correctly classified test sample objects. It should be noted that the results averaged over 100 starts are presented for a modified learning algorithm with the differential evolution method, which is predetermined by the stochastic nature of the algorithm. The horizontal axis in the diagrams shows the numbers of tasks.

Figure 4. Comparison of ID3 algorithms classification efficiency.

Figure 5. Comparison of CART algorithms classification efficiency.
6. Statistical analysis
Statistical analysis for a statistically reliable comparison of the efficiency of the standard and modified algorithms [13] was carried out in the present paper.

The hypothesis of the equality of mathematical expectations was put forward, an alternative hypothesis assumes inequality of mathematical expectations, the critical area is two-way. Cross-validation of each data set was performed, the algorithms were trained and tested several times on different parts of the samples in order to test the hypothesis. Student's t-test was used for comparison. According to Student’s distribution table, \( t_{cr} = 2.101 \) was determined with a significance level of \( \alpha = 0.05 \) [14, 15]. Table 1 shows the observed values of Student’s t-test for the considered tasks. Each cell corresponds to \( t_{obs} \) when comparing standard and modified algorithms.

### Table 1. Experimental values of Student’s t-test (average values).

| Task number | ID3   | CART |
|-------------|-------|------|
| Task 1      | 0.287 | 0.123|
| Task 2      | 0.273 | 0.297|
| Task 3      | 0.018 | 0.332|
| Task 4      | 1.258 | 1.123|
| Task 5      | 1.014 | 1.085|
| Task 6      | 1.255 | 1.315|
| Task 7      | 0.134 | 0.662|
| Task 8      | 0.312 | 0.83 |

All observed values of Student’s t-test from table 1 did not fall into the critical region, i.e. \( t_{obs} < t_{cr} \), therefore the hypothesis of mathematical expectations equality is accepted. Table 2 shows the observed values of Student’s t-test when comparing trees obtained by the standard algorithm with the best trees obtained by the modified algorithm.

### Table 2. Experimental values of Student’s t-test (best values).

| Task number | ID3   | CART |
|-------------|-------|------|
| Task 1      | 3.454 | 1.985|
| Task 2      | 2.351 | 3.171|
| Task 3      | 0.566 | 0    |
| Task 4      | 2.299 | 2.351|
| Task 5      | 2.411 | 2.341|
| Task 6      | 2.908 | 2.661|
| Task 7      | 1.203 | 1.64 |
| Task 8      | 1.927 | 3.054|

In table 2 bold indicates values that exceed the critical indicator. Most of the observed values of the Student's test are greater than the critical indicator, therefore, almost all the best trees found by the modified algorithm have statistically significant differences from the trees obtained by the standard algorithm.

7. Conclusion
In accordance with the statistical analysis the following conclusion can be drawn: when using the proposed modification of the decision tree learning algorithm, the classification efficiency is on average the same. It is also worth noting that while the algorithms perform the same on average, the modified algorithms sometimes allow finding decision trees that perform better on the task. In the future, to
improve the efficiency of this method, it is planned to optimize not only the threshold value, but also the attribute.

Acknowledgments
The research was carried out with the financial support of the project № FEFE-2020-0013.

References
[1] Breiman L, Friedman J H, Olshen R A and Stone C T 1984 *Classification and Regression Trees* (Belmont: Wadsworth) p 368
[2] Hastie T, Tibshirani R and Friedman J 2009 *The Elements of Statistical Learning* (New York: Springer) p 189
[3] Quinlan J R 1993 *C4.5: Programs for Machine learning* (San Francisco: Morgan Kaufmann Publishers) p 302
[4] Quinlan J R 1986 Induction of decision trees *Machine learning* **1**(1) 81-106
[5] Davies D L and Bouldin D W 1979 A Cluster Separation Measure *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2** 224-7
[6] Storn R 2009 On the usage of differential evolution for function optimization *Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS)* 519-23
[7] Goldberg D E 1989 *Genetic Algorithms in Search, Optimization and Machine Learning* (Boston: Addison-Wesley) p 432
[8] Qin A K and Suganthan P N 2005 Self-adaptive differential evolution algorithm for numerical optimization *Proceedings of the IEEE congress on evolutionary computation (CEC)* 1785-91
[9] Semenkin E S and Semenkina M E 2012 Self-configuring Genetic Algorithm with Modified Uniform Crossover Operator *Lecture Notes in Computer Science* 7331 414-21
[10] Semenkin E and Semenkina M 2012 Spacecrafts' control systems effective variants choice with self-configuring genetic algorithm *ICINCO 2012 – Proceedings of the 9th International Conference on Informatics in Control, Automation and Robotics* 84-93
[11] Tanabe R and Fukunaga A 2013 Success-history based parameter adaptation for Differential Evolution *IEEE Congress on Evolutionary Computation Cancun* 71-8
[12] Machine Learning Repository https://archive.ics.uci.edu/ml/index.php (accessed 19.08.2020)
[13] Gmurman V Ye 2003 *Probability Theory and Mathematical Statistics* (Moscow: Vyshaya shkola) p 479
[14] Ayvazyan S A, Enyukov I S and Meshalkin L D 1983 *Applied statistics: Fundamentals of modeling and primary data processing* (Moscow: Finance and Statistics) p 471
[15] Rumshishkiy L Z 1971 *The mathematical processing of the experimental results* (Moscow: Nauka Publ) p 192