1. Introduction

Classification decision tree is one of the ways how to interpret first order rules in sequential and hierarchical structure, where each object from a training set has only one node - leaf, which provides a solution. Application of the classification tree foresees that all possible classes are already defined, and this is the main reason why classification trees are used mainly for discrete value determination. Theoretically it is possible to use a classification tree in continuous value forecasting task, by splitting the range of the target value into intervals and calculating an average and a square root error for each interval. But in most cases one more coefficient is needed to show how much we can trust a forecast – a confidence level of the forecast. All the above topics are discussed in this paper. By using the found solutions, classification decision trees, built with the CART algorithm, are modified to suit the continuous value forecasting task. The methods used are described in Section 2, and practical examples are provided in Section 3.

2. Using classification tree for continuous value forecasts

Classification decision tree is one of the ways how to interpret “If … Then …” rules in sequential and hierarchical structure, where each object from a training set has only one node – a leaf which gives a solution.

One of the tree-based forecasting methods is a regression tree. In CART, regression trees have a constant numerical value in the leaves and use the variance as a measure of impurity, and searching for the best split uses such methods as “Least Absolute Deviation” and “Least Square Root” [2]. In this paper it is shown how to apply a similar approach to continuous value forecasting by using a classification tree building algorithm.

The target value of the forecast must be discretised so that the classification tree building algorithm can be used. Such methods as “Equal Frequency Intervals” (EFI) and “Equal Width Intervals” (EWI) can be applied at this stage. Each of the intervals should be treated as a single class, represented by upper and lower bounds, average value and square root error. As a next step a classification tree is built, using a training data set \( D \), defined by \( n \) records and \( m \) attributes, where \( m-1 \) attributes store descriptive information about the discretised target value. In this paper, the CART algorithm [8] is used to build a classification tree. Such criteria as Gini, Entropy and Twoing are used as splitting criteria [3, 6].

After a classification tree is built, the action-assignment or “Then”- part of the rules should be changed to match the continuous value forecasting task. From a classification tree a tree structure is taken and records from the training dataset with continuous target value are spread across the tree leaves. Then the average value and the square root error (1) are calculated for each leaf.
\[ \forall L_i, i = 1, \ldots, h, \quad \begin{cases} 
    n = 1 \rightarrow s_i = 0 \\
    n > 1 \rightarrow s_i = \sqrt{\frac{1}{n-1} \sum_{j=1}^{n} (Y_j - \bar{Y})^2}, 
\end{cases} \]  

where \( L_i \) is the current leaf of the tree, \( h \) - total number of leaves and \( n \) - number of records on the leaf \( L_i \).

By updating a classification tree in that way, it is possible to make a continuous value forecast. But one of the things that should be taken into account is the process of choosing the best tree. As it will be shown in the example (Section 3), by increasing the number of the intervals for the target value, which would be better for continuous value forecasting, because of the decreasing deviation, the classification error also increases. An explanation for that could be that the attribute values of the records in the continuous intervals become similar as the width of the intervals decreases. Due to that, it becomes harder for a classifier to define correctly the class value (interval) for each record in a training set, and the classification error increases. So if a classification error is considered as a criterion for choosing the best tree for a continuous value forecast, then a tree, built using the training data set with minimum intervals – two- is chosen as the best. The smaller the number of intervals is, the higher the deviation is.

To choose the best tree for making a continuous value forecasts, the weighted average square root error (2) should be used as a criterion.

\[ \bar{s} = \frac{\sum_{i=1}^{h} n_i \cdot s_i}{\sum_{i=1}^{h} n_i}, \]  

where \( h \) is the total number of leafs, \( n_i \) holds the number of records at the leaf \( L_i \) and \( s_i \) is the square root error for the leaf \( L_i \). The tree with the lower \( \bar{s} \) will be the best one, \( T^* = \arg \min (\bar{s}) \). The size of the tree should also be taken into account. An algorithm for this approach is shown in Figure 1.

**Figure 1. Algorithm for choosing the best tree**

One of the problems that can also occur while making a forecast for a new record is that some values on the leaf to which the new record is linked, might be blocked [1]. The blocked values are such values for which it is known a priori that they cannot appear in the forecast for the observed new record. An example of such case is provided in Section 3. The problem is that the majority of the records on observed leaf \( L_i \) could be blocked and the forecast should be made by not using all available information. Figure 2 displays possible value range on the leaf with blocked intervals.
To solve this problem, it is necessary to measure the amount (or part) of the information that was used for the forecast. This could be measured as a part of the records in the leaf $L_i$ that are not blocked and are used for making a forecast. This value should be calculated as follows:

$$p_i = \frac{n_i}{n_L},$$

where $n_i$ is the number of records in the none blocked value interval $i$ and $n_L$ - total number of records in the leaf $L$. If there are more than one none blocked value interval, then the average value, square root error and a $p_i$ must be calculated for each of the intervals and the decision maker should choose the forecast to use by taking all three parameters into account.

3. Practical examples

In this section, examples for the aforementioned methods and cases are presented. The dataset used in the examples contains data on eBay auctions, supplied by DATA MINING CUP year 2006 international contest [7]. The size of the supplied dataset is 8000 records and 26 attributes of different type. The data were pre-processed [4, 5], after that the size decreased to 7882 records and 18 attributes. As a target value, the end price of the auction was chosen.

3.1. Building classification trees

To build classification trees a product of the Salford Systems Company – data mining tool CART 5 was used. Figure 3 demonstrates how a classification error changes as the number of the intervals for the target value increases.

![Figure 3. Changes in the classification error. Chart a) for EFI and chart b) for EWI](image)

The target value in the training data set was discretised using different methods (EFI and EWI) and different numbers of intervals. Totally 10 training datasets were obtained and used to build 30 classification trees by using different splitting criteria, as was mentioned in
Section 2. Table 1 contains data on classification errors that were used to build Figure 3. For testing classification trees a 10-fold cross-validation was used.

Table 1

<table>
<thead>
<tr>
<th>Intervals</th>
<th>GINI EFI</th>
<th>GINI EWI</th>
<th>ENTROPY EFI</th>
<th>ENTROPY EWI</th>
<th>TWOING EFI</th>
<th>TWOING EWI</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.185993</td>
<td>0.185993</td>
<td>0.184978</td>
<td>0.128267</td>
<td>0.185993</td>
<td>0.185993</td>
</tr>
<tr>
<td>4</td>
<td>0.381756</td>
<td>0.369703</td>
<td>0.379853</td>
<td>0.47082</td>
<td>0.380741</td>
<td>0.371226</td>
</tr>
<tr>
<td>6</td>
<td>0.483126</td>
<td>0.290916</td>
<td>0.48338</td>
<td>0.339127</td>
<td>0.484395</td>
<td>0.444938</td>
</tr>
<tr>
<td>8</td>
<td>0.556711</td>
<td>0.426288</td>
<td>0.55595</td>
<td>0.418422</td>
<td>0.55595</td>
<td>0.389368</td>
</tr>
<tr>
<td>10</td>
<td>0.615707</td>
<td>0.467394</td>
<td>0.618878</td>
<td>0.373002</td>
<td>0.610886</td>
<td>0.485537</td>
</tr>
</tbody>
</table>

The data obtained show that a classification tree, built with minimal number of intervals (two), provides better results for the classification. But it may not perform better than the other trees while making a forecast for a continuous value. So a different performance measure should be used to choose the best tree.

3.2. Choosing the best tree

To demonstrate the best tree choosing method, described in Section 2, six classification trees were built by using datasets with 2, 4 and 6 EF and EW intervals for the target value. GINI index was used as a splitting criterion. Trees were modified using the Microsoft Excel. The results of this experiment are displayed in Figure 4.

Figure 4. Experimental results: a) – $\bar{s}$ using EWI and EFI; b) – size of the tree using EWI and EFI

As it was expected, the results show that a forecast error is decreasing as the number of intervals increases. As can be seen from the results, the EFI target value discretisation method gives better results, comparing to EWI. But the size of the tree should also be taken into account. Sizes of trees are shown in Figure 4.b. Comparing $\bar{s}$ for trees, built using 4 EF and EW intervals, the results are 27.81 and 34.04 accordingly – less than 7 EUR, but the size of the tree is 115 for EFI and 33 for EWI.

3.3. Managing blocked intervals

To show how the case with blocked intervals could be managed, a classification decision tree was built, using a dataset with target value discretised in 4 EW intervals and Entropy as a splitting criterion. The tree was pruned; it is shown in Figure 5.
By following the steps, described in Section 2, classification decision tree was modified to match the continuous value forecasting task. The results are summarised in Table 2.

Table 2

<table>
<thead>
<tr>
<th>Node</th>
<th>Records</th>
<th>min</th>
<th>max</th>
<th>Average</th>
<th>Square Root Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>3279</td>
<td>5.5</td>
<td>399</td>
<td>171.1257</td>
<td>29.28866</td>
</tr>
<tr>
<td>L2</td>
<td>1931</td>
<td>1</td>
<td>412</td>
<td>215.7492</td>
<td>47.36066</td>
</tr>
<tr>
<td>L3</td>
<td>1415</td>
<td>88.88</td>
<td>601.51</td>
<td>212.1435</td>
<td>48.07525</td>
</tr>
<tr>
<td>L4</td>
<td>399</td>
<td>101</td>
<td>414</td>
<td>221.5555</td>
<td>48.61258</td>
</tr>
<tr>
<td>L5</td>
<td>858</td>
<td>1</td>
<td>426</td>
<td>217.0134</td>
<td>49.86823</td>
</tr>
</tbody>
</table>

Assume that a new record is classified as one of the leaf’s L1 records and blocks values in range [100; 200] as it is shown in Figure 6.

As one of the cases mentioned in Section 2, the blocked interval covers the largest part of the records on leaf L1. The results on calculation of parameters for each of two available value intervals are summarised in Table 3.

Table 3

<table>
<thead>
<tr>
<th>Interval</th>
<th>Records</th>
<th>Average</th>
<th>Square Root Error</th>
<th>$p_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>4</td>
<td>38.875</td>
<td>36.36247</td>
<td>0.00122</td>
</tr>
<tr>
<td>I2</td>
<td>446</td>
<td>221.8534</td>
<td>27.55675</td>
<td>0.136017</td>
</tr>
</tbody>
</table>

Interval I2 returns the largest $p_i$ value, so it can be employed as a forecast. But it uses only 13.6% of records on the leaf L1 and now it is up to the decision maker either to accept this forecast or not.

4. Conclusions

Theory and practical examples in this paper show that if a decision tree is chosen as a tool for continuous value forecasting, then it is not necessary to restrict your choice to using decision tree building methods, which use variance as a splitting criterion (for example, regression trees). Any decision tree building method, suitable for the used data, can be applied
to build a tree structure that can be modified to match a continuous value forecasting task. One of the possible approaches to classification tree modification is described in this paper. The paper also considers problems connected with continuous value forecasting, such as the best tree choosing and managing cases with blocked value intervals. Section 2 describes possible solutions to these problems but Section 3 – examples of described problem solving approaches.

References


Serge Parshutin, M.Sc., Institute of Information Technology, Riga Technical University, 1 Kalku Street, Riga LV-1658, Latvia. E-mail: serge.parshutin@cs.rtu.lv.

Arkady Borisov, Prof., Dr.habil.sc.comp., Institute of Information Technology, Riga Technical University, 1 Kalku Street, Riga LV-1658, Latvia. E-mail: aborisov@cs.rtu.lv.

Annotation

This paper is focused on such topics as possibility of using classification decision trees for forecasting continuous values and related problems: classification tree modification, choosing the best tree and managing the blocked value intervals. Section 2 presents theoretical solutions to the mentioned problems and Section 3 shows how methods, considered in Section 2, are realised in practise.