Decision Tree Creation Methodology Using Propositionalized Attributes

Pēteris Grabusts1, Arkādijs Borisovs2, Ludmila Aleksejeva3
1 Rezekne Academy of Technologies, 2, 3 Riga Technical University

Abstract – The aim of the article is to analyse and thoroughly research the methods of construction of the decision trees that use decision tree learning with statement propositionalized attributes. Classical decision tree learning algorithms, as well as decision tree learning with propositionalized attributes have been observed. The article provides the detailed analysis of one of the methodologies on the importance of using the decision trees in knowledge presentation. The concept of ontology use is offered to develop classification systems of decision trees. The application of the methodology would allow improving the classification accuracy.

Keywords – Decision tree, ontology, propositionalization, taxonomy.

I. INTRODUCTION

This paper is the continuation of last year’s publication “Ontology-based Classification System Development Methodology” [7], where the main goal was “to analyse ontology-based classification systems with decision trees”. Some methods were investigated – ontology-based inductive learning systems with classification rules and decision tree learning with taxonomy of proposed attributes.

To achieve the goal, theoretical research was conducted on the methods using ontologies in decision-tree classification systems. Ontologies were used in classification tasks with real and artificial data [5], [8]. Ontologies of the proposed attributes were recommended to improve the quality of classification for datasets with a small amount of unique values.

To complete this topic, it is planned to undertake the detailed study on decision tree construction techniques using propositionalized attributes.

Nowadays, the analysis and interpretation of data processing results are significant and important. Usually, there is a desire to reflect the results in the form of the rule – in the form of knowledge. Therefore, it is necessary to search for the ways and methods of such knowledge mining.

Classification is one of the main tasks of data mining – determination of the belonging of the object to predefined object groups. These predefined groups are called classes, but the process – classification. During a classification stage, the classification model or classifier is created – the model determines classes based on the rules that are derived during classification. There are a lot of classic algorithms and techniques to carry out classification, but in ontology classification and clustering they are used rarely, however, the need for them is confirmed by the author [6], who describes problems with the use of data mining in e-commerce and points out that hierarchical background knowledge is necessary. Solving such problems could be one of the possible uses of the ontology classification, so the authors’ motivation is to investigate the methods of using ontologies in decision tree-based classification systems.

II. CLASSICAL DECISION TREE LEARNING METHODS

Data classification process consists of two stages: training on the base of existing data and new data classification. First, the model training is carried out using one of the classification algorithms. In this process, a classification model or classifier is obtained. After that the use of classifier on new data is carried out, including model testing and classification evaluation.

Classification system includes a classifier, pre-treatment, post-processing, and classifier modelling.

Decision trees – a way of representing rules in a hierarchical, coherent structure, where each object corresponds to a single node, giving the decision. The rule refers to a logical structure presented in the form of “If-Then”.

The application area of decision trees is wide, but all the problems solved by this unit can be grouped into the following classes:

- Data description: Decision trees allow storing information about the data in a compact form; instead we can store a decision tree that contains an exact description of the objects.
- Classification: Decision trees primarily cope with the tasks of classification, i.e. matching the objects with one of the previously known classes. The target variable must have discrete values.

Classification trees are considered in the article. A decision tree is a classification using a recursive instance space division. Decision tree is composed of nodes and oriented arcs. The root node has no incoming arcs. All other nodes have exactly one incoming arc. Internal node has an incoming arc and one or more outgoing arcs. The leaves of decision tree are nodes, which have an incoming arc, but have no outgoing arcs.

Various algorithms can be used for creation of a decision tree. The most commonly used in classification are ID3 (Iterative Dichotomiser 3) [2],[11], C4.5 (ID3 successor) [2],[3], [10] and CART (Classification and Regression Tree) [4]. Since these algorithms are widespread, there is no need to describe them in this article.

For example, a decision tree for a well-known Iris dataset [12] is given in Fig. 1.
III. PROBLEM FORMULATION

A. Related Studies

Three approaches to ontology use for higher accuracy and shorter rule generation can be found in literature review [9]:
- Attribute value taxonomy (AVT);
- Word taxonomy (WT);
- Propositionalized attribute taxonomy (PAT).

The value of the attribute in taxonomy use for the creation of the classifier single taxonomy for each attribute is used in order to obtain classification rules of different levels. This approach automatically generates the taxonomy of attribute values for each attribute and uses special decision tree learning to create the rules. To solve classification problems in decision tree learning, which is based on attribute value taxonomy, the Naive Bayes classifier is used.

The word taxonomy is used to group words and sentences hierarchically, and then this taxonomy is used to classify the entries. This approach also uses the Naive Bayes classifier.

Next, the use of propositionalized attribute taxonomy methodology in classification tasks is considered and analysed. Kang [9] offers a new automatic way how domain ontology can be obtained from the data sets to be used to classify the database entries in different sections with the help of a decision tree.

The method introduces an attribute or propositionalized attribute taxonomy transformed into statements, PAT, to conduct the learning algorithm of a decision tree, PAT-DTL, which extends the C4.5 learning algorithm of decision tree to be used in the created PAT taxonomy. PAT-DTL is used in both top-down and bottom-up directed search methods in the PAT taxonomy to find the necessary abstraction for a classification task.

Propositionalization is a process where a relational data set is clearly and explicitly transformed into a propositional data set. In this process, the input data are in the form of a relational database table and the output data are an attribute-value representation in the form of one table, where each example applies to one entry and is described with the values of a specific attribute set. The aim of propositionalization is to make the pre-treatment of relational data in order to analyse them later using machine learning tools, which use attribute-value input data.

The operation of these algorithms is presented in detail in [9].

IV. DECISION TREE AND KNOWLEDGE PRESENTATION

Decision tree method makes it possible to predict the belonging of objects to one or another class depending on the respective values of attributes characterising these objects. Decision trees provide the construction of automatic rules "If-Then" on available statistics and on the basis of that further decision on affiliation of observation or object to a particular class is made.

Let there be n objects represented by a set \( T = \{ t_1, ..., t_n \} \), where each element of this set is described by the same set of attributes named \( C_i, i = 1, ..., m \). Each propositionalized or pseudo-attribute can take \( k_i \) values- \( x_{ip}, p = 1, ..., k_i \), measured in a random scale.

If we look at the statistics, such as, for example, the bank clients [1], then the clients themselves are the set \( T \). Each client is characterised by a set of characteristics: gender, age, crediting purpose, total income, etc. These attributes are \( C_2, C_3, C_4 \) etc. Attribute \( C_1 \) (gender) can take two values: \( M \) and \( F \), i.e., \( x_{11} = M, x_{12} = F \) and so on.

Let there be a set of classes \( K_j \). Herewith, each object of set \( T \) (each bank customer) has been assigned to a certain class of objects, and this is shown in the statistics. For example, in the case of bank customers there can be two classes: \( K_1 \) (the borrower repays the loan on time) and \( K_2 \) (the borrower fails to satisfactorily repay the loan). It is required to construct the classifying rules to identify the regularities between the values of the attributes of each object from set \( T \) and class \( K_j \), which the object belongs to.

Classifying rule is the following: if the attributes of the object \( t_i (i = 1, ..., n) \) takes the values

\[ C_1 = x_{1p} \text{ and } C_2 = x_{2p} \cdots C_m = x_{mp}, \]

then \( t_i \) belongs to class \( K_j \).

To construct the classifying rule, it is necessary to construct a decision tree first, the top (root) of which is a check of the first attribute value of the presented object for conformity with the class, branches – intermediate checks, and the leaves – classes of objects. Such a tree can have the form shown in Fig. 2. Then, by each branch from top to bottom the renewal of the classifying rule takes place.

Constructing the tree in Fig. 2 presents no problems when for each set (or group of sets) of object attribute values to one correspondence it is possible to put a definite class of objects. However, in practical problems such compliance often has probabilistic nature. In other words, one of the selected object classes corresponds to each of the ordered sequence of object attribute values only with a certain probability. In this case, the classification is performed in a probabilistic uncertainty.
characterised by a certain probability.  

of attribute chain and a certain class, is a random event and is 
event, which establishes a correspondence between the values 
probabilistic uncertainty of classification. In other words, an 
of attribute values of different objects, which results in the 

values:

In Table I, different classes may correspond to the same sets 
of attribute values of the selective set of object the appropriate 

For example, if there is a single attribute \(C_1\) with two possible 
values: \(x_{11}\) and \(x_{12}\) then the table of statistics for 10 objects (set 
\(T\)) might look as follows (Table II).

In Table I, different classes may correspond to the same sets 
of attribute values of different objects, which results in the 
probabilistic uncertainty of classification. In other words, an 
event, which establishes a correspondence between the values 
of attribute chain and a certain class, is a random event and is 
characterised by a certain probability.

For example, if there is a single attribute \(C_1\) with two possible 
values: \(x_{11}\) and \(x_{12}\) then the table of statistics for 10 objects (set 
\(T\)) might look as follows (Table II).

<table>
<thead>
<tr>
<th>Name of object</th>
<th>Values of object attributes</th>
<th>Class of object</th>
</tr>
</thead>
<tbody>
<tr>
<td>(t_1)</td>
<td>(x_{11}) (x_{12}) ... (x_{10})</td>
<td>(K_1)</td>
</tr>
<tr>
<td>(t_2)</td>
<td>(x_{11}) (x_{12}) ... (x_{10})</td>
<td>(K_1)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>(t_n)</td>
<td>(x_{11}) (x_{12}) ... (x_{10})</td>
<td>(K_n)</td>
</tr>
</tbody>
</table>

In Table I, different classes may correspond to the same sets 
of attribute values of different objects, which results in the 
probabilistic uncertainty of classification. In other words, an 
event, which establishes a correspondence between the values 
of attribute chain and a certain class, is a random event and is 
characterised by a certain probability.

For example, if there is a single attribute \(C_1\) with two possible 
values: \(x_{11}\) and \(x_{12}\) then the table of statistics for 10 objects (set 
\(T\)) might look as follows (Table II).

<table>
<thead>
<tr>
<th>Name (number) of object</th>
<th>Value of attribute (C_1)</th>
<th>Class of object</th>
</tr>
</thead>
<tbody>
<tr>
<td>(t_1)</td>
<td>(x_{11})</td>
<td>(K_1)</td>
</tr>
<tr>
<td>(t_2)</td>
<td>(x_{11})</td>
<td>(K_1)</td>
</tr>
<tr>
<td>(t_3)</td>
<td>(x_{12})</td>
<td>(K_1)</td>
</tr>
<tr>
<td>(t_4)</td>
<td>(x_{12})</td>
<td>(K_1)</td>
</tr>
<tr>
<td>(t_5)</td>
<td>(x_{11})</td>
<td>(K_2)</td>
</tr>
<tr>
<td>(t_6)</td>
<td>(x_{12})</td>
<td>(K_2)</td>
</tr>
<tr>
<td>(t_7)</td>
<td>(x_{11})</td>
<td>(K_2)</td>
</tr>
<tr>
<td>(t_8)</td>
<td>(x_{11})</td>
<td>(K_1)</td>
</tr>
<tr>
<td>(t_9)</td>
<td>(x_{12})</td>
<td>(K_2)</td>
</tr>
<tr>
<td>(t_{10})</td>
<td>(x_{11})</td>
<td>(K_1)</td>
</tr>
</tbody>
</table>

From Table II it is seen that in 7 cases (a subset \(T_1 (C_1)\) with 
an attribute value \(x_{11}\) 5 objects were classified to the class \(K_1\) 
and 2 objects to the class \(K_2\); in 3 cases (a subset \(T_2 (C_1)\)) with 
a value \(x_{12}\) 2 objects were classified to class \(K_2\) and 1 to class 
\(K_1\). The values of attributes and classes of objects are dependent 
on random events; therefore, the conditional probabilities of a 
random classification are calculated in the following way:

\[
P(K_1 | C_1 = x_{11}) = \frac{5}{7}; \quad P(K_2 | C_1 = x_{11}) = \frac{2}{7};
\]

\[
P(K_2 | C_1 = x_{12}) = \frac{2}{3}; \quad P(K_1 | C_1 = x_{12}) = \frac{1}{3}.
\]

Appropriate classification rules under the condition of 
probabilistic uncertainty can be written as follows:

- if \(C_1 = x_{11}\), then with a probability of 5/7 the object 
  belongs to class \(K_1\);
- if \(C_1 = x_{12}\), then with a probability of 2/3 the object 
  belongs to class \(K_2\).

It is seen that in the presence of a single attribute, the 
automatic construction of rules is not difficult. In case there are 
several attributes, it is needed to select the order of the sequence 
of analysis of their values. Ordering attributes is 
advantageously carried out according to the principle of 
maximum removal of uncertainty; the measure of uncertainty is 
the information entropy.

If the characteristic attributes were not used, then the 
probability of assigning any new object (not included in the 
Table II) to class \(K_1\) would be \(P(K_1) = 6/10\), and to class 
\(K_2 - P(K_2) = 4/10\). In this case, the value of entropy is 
calculated by the formula:

\[
H(T) = - \sum_{i=1}^{2} P(K_i) \ln P(K_i) = 0.673.
\]

The use of a single attribute for classification from Table II 
reduces the extent of uncertainty. The corresponding value of 
entropy in this case is given by:

\[
H(T_1, T_2) = \frac{|T_1|}{|T|} \cdot H(T_1) + \frac{|T_2|}{|T|} \cdot H(T_2) =
\]

\[
= \left[ \frac{7}{10} \cdot \left( - \sum_{i=1}^{2} P \left( \frac{K_i}{C_1} = x_{11} \right) \ln P \left( \frac{K_i}{C_1} = x_{11} \right) \right) \right] +
\]

\[
+ \left[ \frac{3}{10} \cdot \left( - \sum_{i=1}^{2} P \left( \frac{K_i}{C_1} = x_{12} \right) \ln P \left( \frac{K_i}{C_1} = x_{12} \right) \right) \right] = 0.6085.
\]

When using several attributes as the first attribute for 
analysis, it is necessary to select the one that provides the 
maximum reduction of classification uncertainty with respect to 
the original set. As an example, let us include one more attribute 
\(C_2\) with values \(x_{21}, x_{22}\) in Table II and we will get Table III.
The analysis of classifying ability of the second attribute $C_2$ indicates that the capacity of sub-multitude $T_1$ ($C_2$) is equal to 7, and probability $P(K_1|C_2 = x_{21}) = 6/7$; the capacity of subset $T_2(C_2)$ is equal to 3, and the probability $P(K_2|C_2 = x_{22}) = 1$. We calculate the value of entropy for the second attribute:

$$H(T_1, T_2) = \frac{[T_1]}{[T]} \cdot H(T_1) + \frac{[T_2]}{[T]} \cdot H(T_2) =$$

$$= \left(\frac{7}{10}\right) \cdot \left(-\sum_{i=1}^{2} P(K_i|C_2 = x_{21}) \ln P(K_i|C_2 = x_{21})\right) +$$

$$+ \left(\frac{3}{10}\right) \cdot \left(-\sum_{i=1}^{2} P(K_i|C_2 = x_{22}) \ln P(K_i|C_2 = x_{22})\right) = 0.175.$$

The second attribute eliminates the uncertainty to a much greater extent; therefore, in this example the chain of checks in the decision tree begins with the second attribute. Let there be a set of objects $T$, where each element of the set is described with $m$ attributes $C_1, ..., C_m$. Also, a set of classes $[K_j], j = 1, ..., J$ is given and it is known to which class each object of set $T$ belongs. The process of constructing the tree takes place from the top down – the root of the tree is created first, then the descendants of the root, etc.

We use the following algorithm to construct a decision tree [1]:

Step 1. There is an empty tree (there is only a root), and the initial set $T$ (associated with the root). In the root of the tree, the statistical probabilities of belonging of each new object to a particular class $K_j, j = 1, ..., J$ are counted. Apparently, $P(K_j) = n_j/n$, where $n$ – the number of objects of set $T$ belonging to class $K_j$.

Step 2. It is required to divide the initial set $T$ into subsets. It can be done by selecting one of the attributes $C_i$ and sorting through all the possible values of this attribute. At the same time, each value of $C_i$ is associated with a particular subset of set $T$ consisting of elements from which attribute $C_i$ has taken this value. Then, as a result of division, there are $k_1$ subsets of $T_1, ..., T_{k_1}$, and respectively $k_i$ descendants of the root are created, each of which is assigned to its own subset resulting in division of $T$. For each of the descendants (respectively, for each subset of $T$) probabilities of belonging of each new object to one or another class $K_j, j = 1, ..., J$ are calculated. Unlike the prior probabilities $P(K_j)$, these probabilities are already conditional, as they depend on a value the attribute $C_i$ takes.

Calculation of the probabilities is $P \left( \frac{K_j}{C_i} = x_{ip} \right), j = 1, ..., J, p = 1, ..., k_i$ carried out according to the Bayes' formula:

$$P \left( \frac{K_j}{C_i} = x_{ip} \right) =$$

$$= \frac{P(K_j) P(C_i = x_{ip} | K_j)}{P(K_j)} = \frac{x_{ip} K_j}{\sum_{j=1}^{J} x_{ip} K_j}.$$

Step 3. All the actions of Step 2 are repeated, but there is already a division into subsets, the subsets of $T_{pl}$ themselves by the following selected attribute. Each set $T_{pl}$ can be divided into subsets on different attributes. To calculate conditional probabilities, the Bayes’ formula will change, as the event is that two attributes have adopted certain values (moreover, independently of each other):

$$P \left( \frac{K_j}{C_i} = x_{ip}, C_t = x_{tr} \right) =$$

$$= \frac{P(K_j) P(C_i = x_{ip}, C_t = x_{tr})}{P(K_j)} = \frac{x_{ip} K_j}{\sum_{j=1}^{J} x_{ip} K_j}.$$

Step 4. All the actions of Step 3 are repeated with further division into subsets (creation of descendants) and the correction of Bayes’ formula based on appearance of another attribute, etc. The criterion for choosing the attribute on which the division of corresponding subset of $T_p$ should take place is the minimum entropy:

$$\min_{C_i} (H(C_i | T_p)).$$

The process of branching in a certain direction takes place until we get the top, in which the posterior probability of belonging the object to a certain class is equal to 1.

V. CONCEPT OF ONTOLOGY-BASED CLASSIFICATION SYSTEM

Decision tree learning, using pseudo-attribute taxonomy, consists of the following stages (see Fig. 3): pre-treatment of the data sets; pseudo-attribute data set creation from the data set attributes; pseudo-attribute taxonomy creation; the decision tree learning and test performance. A further statement is based on [9] and [7] of described algorithms.

Stage 1. It is possible to use a wide range of data sets with some limitations: the data set must be full or the missing values should make a small percentage of all values. If the data set contains continuous attributes, they must be converted into discrete intervals.
Stage 2. Pseudo-attribute set is created as follows: first, for each attribute this unique attribute value domain is found. Then, based on this unique value pairs pseudo-attributes are designed that consist of attribute-value pairs and pseudo-attribute value set is \{0,1\} or \{True, False\}. After new pseudo-attribute creation, a set of data is transformed, converting each entry into a new pseudo-attribute set.

Stage 3. Pseudo-attribute taxonomy can be created from the new data set using similarity measures. Taxonomy is created agglomeratively as a starting point choosing the pseudo-attribute set, then the class distribution to pseudo-attribute value “1” is calculated for each pseudo-attribute. Taking into account the obtained values, the similarity measure J-divergence is calculated for each pair of pseudo-attributes.

A pair of attributes with the lowest J-divergence value is found and pseudo-attribute pair \((x \text{ and } y)\) with the lowest J-divergence value is incorporated into a new z value by combining the attribute values with logical OR. Then the class distribution to the combined attribute z is calculated and attribute z is added to the taxonomy as a parent according to the x and y terms.

After that, the data set is changed – the combined value of z is added to the data set and pseudo-attributes x and y are removed from the data set. Then it is checked whether the current cut size = 1. If the current cut size is 1, then the aim has been achieved and taxonomy is withdrawn. If the current cut size is not one cut, then J-divergence values are calculated again to determine which attributes are next to be combined.

Stage 4. The decision tree creation and testing include the fulfillment of multiple C4.5 algorithms, which are based on data sets that are formed in accordance with the previously created taxonomy. After the decision tree creation, the cross-validation is done and testing accuracy is obtained. Then the parent set of cut elements is made and each element of pseudo-attribute data set is replaced with its parent.

The work is finished when all taxonomies are passed through or in parent data set there are no more “valid” parents.

VI. CONCLUSION

There are a lot of methodologies with propositionalized attributes on decision tree construction techniques. The methodology developed in the article is one of many methodologies of such type used in decision tree learning and ontology application, and the authors’ task has been to explore suchlike methodology. One of the methodologies on decision tree construction with assistance of propositionalized attributes for knowledge presentation has been examined in the article. The concept of ontology use in developing classification systems of decision trees has been proposed. The application of the methodologies would allow improving the classification accuracy. The use of ontology in classification tasks, decision tree learning and analysis has great prospects. In the future research, the opportunities of this methodology will be evaluated comparing it to other similar methodologies.

REFERENCES


Pēteris Grabusts received his Dr. sc. ing. degree in Information Technology from Riga Technical university in 2006. Since 1996 he has been working at Rezekne Academy of Technologies. Since 2014 he has been a Professor at the Department of Computer Science. His research interests include data mining technologies, neural networks and clustering methods. His current research interests include ontologies. E-mail: peter@ru.lv

Arkādijs Borisovs received his Doctoral degree in Technical Cybernetics from Riga Polytechnic Institute in 1970 and Dr. habil. sc. comp. degree in Technical Cybernetics from Taganrog State Radio Engineering University in 1986. He was a Professor of Computer Science at the Faculty of Computer Science and Information Technology, Riga Technical University (Latvia). His research interests included fuzzy sets, fuzzy logic and computational intelligence. He wrote more than 235 publications in the field.

Ludmila Aleksejeva received her Dr. sc. ing. degree from Riga Technical University in 1998. She is an Associate Professor at the Department of Modelling and Simulation, Riga Technical University. Her research interests include decision making techniques and decision support system design principles, as well as data mining methods and tasks, and especially collaboration and cooperation of the mentioned techniques. E-mail: ludmila.aleksejeva_1@ru.lv