DE GRUYTER OPEN
Information Technology and Management Science
doi: 10.1515/itms-2015-0020
2015 / 18

Ontology-Based Classification System Development Methodology

Peter Grabusts[1], Arkady Borisov[2], Ludmila Aleksejeva[3]
[1]*Rezekne Higher Educational Institution,* [2, 3]*Riga Technical University*

Abstract **– The aim of the article is to analyse and develop an ontology-based classification system methodology that uses decision tree learning with statement propositionalized attributes. Classical decision tree learning algorithms, as well as decision tree learning with taxonomy and propositionalized attributes have been observed. Thus, domain ontology can be extracted from the data sets and can be used for data classification with the help of a decision tree. The use of ontology methods in decision tree-based classification systems has been researched. Using such methodologies, the classification accuracy in some cases can be improved.**

Keywords **– classification, decision tree, ontology, propositionalization, taxonomy.**

I. Introduction

The rapid development of information technologies, in general, and the progress in methods of collection, storage and processing of data, in particular, allow many organisations to collect big amounts of data that need to be analysed. The volume of data is so huge that the capacity of experts is not sufficient, creating demand for automatic data analysis methods, which increases every year.

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.

In recent years, the development of ontologies – explicit formal description of the terms and relations among them – has transferred from the world of laboratories through the artificial intelligence to subject experts' desktops [5]. In the World Wide Web, the ontologies have become commonplace. Ontologies in the network range from large taxonomies, categorising websites to categorising goods sold and their characteristics. Standard ontologies that can be used by subject experts to share and annotate information in their field are being developed in many disciplines.

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 [9], 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 trees-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 three 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.

• Regression: If the target variable has continuous values, the decision trees allow establishing the dependence of the target variable from independent (input) variables. For example, the problems of numerical forecasting (prediction of the values of target variable) refer to this class.

Classification trees are viewed 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 outcoming arcs. The leaves of decision tree are nodes, which have an incoming arc, but have no outcoming arcs [3].

Each internal node of decision tree divides the instance space into one or more sub-instance spaces in accordance with the input attribute value of discrete function (in the simplest case, one attribute and the instance space are divided according to the attribute values).

The leaves of an inductive decision tree display classifications and the arcs represent combination of features.

The application of a decision tree is shown in the following steps: first, a classification algorithm and a set of training data are used to create a decision tree, and then the new entry classification using a decision tree takes place. New entry classification is done, starting with the node of the tree root where the value of the corresponding entry attribute is checked, and the appropriate arc of the decision tree is selected, and then the next node of the decision tree and the value of corresponding entry attribute are recursively checked until the corresponding node of the tree is a leaf. The leaves of the decision tree represent classes and leaf-linked class is assigned to the entry.

Decision trees in classification are used when entries of data sets consist of pairs of "attribute-value", the classes are a discreet set of values. Decision trees can be used for verifying data containing errors and missing values of the data [1].

Various algorithms can be used for creation of a decision tree. The most commonly used in classification are ID3 (Iterative Dichotomiser 3) [1], [13], C4.5 (ID3 successor) [1], [2] and CART (Classification and Regression Tree) [4]. Since these algorithms are widespread, there is no need to describe them in this article.

A well-known Iris dataset [15], Matlab [14] and Weka [16] software environment were selected to perform the experiments. A decision tree for Iris dataset by using Weka is given in Fig. 1.

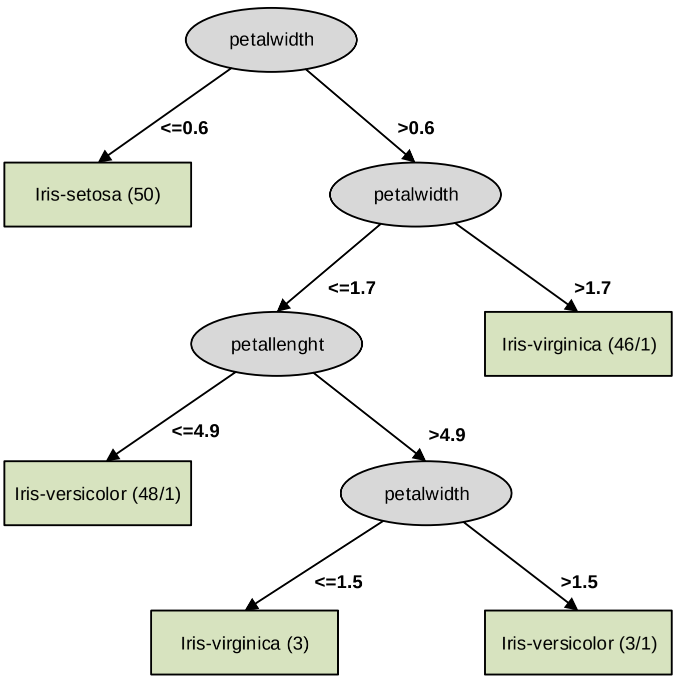

Fig. 1. Decision tree for Iris dataset.

III. Ontology Approach

Over the past twenty years, the term "ontology" has been introduced and is commonly used in the engineering science; it is used to describe the models with different levels of detail (structuring) in an understandable way, as well as to demonstrate wide and complex information through conceptual schemes.

Informally, ontology is a description of the view of the world in relation to a particular area of interest. This description consists of the terms and rules for the use of these terms, limiting their roles within a specific area.

The main components of the ontology are: classes or concepts, relationships, functions, axioms, and examples.

There are various definitions of ontology, but recently the generally recognized is the following definition: "An ontology is a formal explicit specification of a shared conceptualisation" [10]. Ontologies are often equated with taxonomic hierarchies of classes. Thus, the aim of ontology is to accumulate knowledge in general and formal way.

Ontologies can be classified in different forms. One of the most popular types of classification is proposed by Guarino, who classified types of ontologies according to their level of dependence on a particular task or point of view [8]:

- *Top-level ontologies*: describe general concepts like space, time, event, which are independent of a particular problem or domain.
- *Domain-ontologies*: describe the vocabulary related to a generic domain by specialising the concepts introduced in the top-level ontology.
- *Task ontologies*: describe the vocabulary related to a generic task or activity by specialising the top-level ontologies.
- *Application ontologies*: they are the most specific ones. Concepts often correspond to roles played by domain entities. They have a limited reusability as they depend on the particular scope and requirements of a specific application.

Ontologies are widely used in Semantic Web and document clustering, but there is very little information about the use of ontology in numerical data classification and clustering.

Thus, the ontology is an explicit representation of knowledge. It is a formal, explicit specification of shared conceptualisations, representing the concepts and their relations that are relevant for a given domain of discourse [10].

Ontology in its simplest form is a taxonomy. The taxonomy is a hierarchical tree-shape classification of the defined theme, which is centred on the "is-a" link recognition between classes. If the taxonomy is a tree-shape structure, then ontology is a wood-shape structure, as it may contain interlinked taxonomies.

All taxonomies are ontologies, but not all ontologies are taxonomies.

Taxonomic structures are often used to describe the concept of interdependence. In commercial environment, the concept "taxonomy" is often used, while in academic literature and in the environment, the concept "ontology" is used more often.

Taxonomy is limited knowledge acquisition, because it does not have the flexibility of link description. Since the ontology in practice means the representation of a certain domain, or field of knowledge data of the related concepts in the form of the net, the ontology includes and combines the related taxonomies but, additionally, may contain attributes with cardinality and value restrictions. Among the concepts in ontology, there can be any number of relationships and, therefore, ontologies are suitable for knowledge retrieval area.

In classification tasks, the domain ontology is most widely used.

Ontologies and classification have a very wide range of application in various sectors and areas, where it is necessary to facilitate human work with the system operation, however, ontologies and classification are not concepts that often appear together – only by increasing the amount of data, it has been found out that the classification only is not sufficient to form good and short rules for classification and some domain knowledge is required to improve the work of classification [9]. However, the use of ontology is not limited to the provision of background knowledge. Ontologies in classification can also be used to create a classification system itself, not only classifiers [7].

The use of ontologies in classification has great prospects in future and it is shown by the active work on the creation of automatic ontologies [6], limiting manual work, that will facilitate the use of ontologies in various classification systems and provide positive results of successful ontology implementation in classification systems [7], [11], [12].

IV. Problem Formulation

A. Related Work

Three approaches to ontology use for higher accuracy and shorter rule generation can be found in literature review [12]:

- 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 get 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 use this taxonomy to classify the entries. This approach also uses the Naive Bayes classifier.

Next, the use of propositionalized attribute taxonomy methodology in classification tasks is viewed and analysed. Kang [12] 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 relational data set clearly and explicitly is transformed into propositional data set. In this process, the input data is in the form of relational database table and the output data is 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.

For further presentation, it is necessary to introduce some definitions from [12].

Let $\mathbb{A} = \{A_1, A_2, \ldots, A_{|\mathbb{A}|}\}$ be a set of nominal attributes. Let $V_i = \{v_i^1, v_i^2, \ldots, v_i^{|V_i|}\}$ be a finite domain of mutually exclusive values associated with attribute A_i where v_i^j is the j th attribute value of A_i and $|V_i|$ is the number of attribute values of A_i. Let $\mathbb{C} = \{C_1, C_2, \ldots, C_{|\mathbb{C}|}\}$ be a set of mutually disjoint class labels. Instance I is a fixed tuple of attribute values such that $I \in V_1 \times V_2 \times \ldots \times V_{|\mathbb{A}|}$. Data set D is a collection of instances and their associated class label such that $D \subseteq V_1 \times V_2 \times \ldots \times V_{|\mathbb{A}|} \times \mathbb{C}$. Then propositionalization is a function $f: V_i \to \widetilde{\mathbb{A}}$ that, for each value $v_i^j \in V_i$ associated with $A_i \in \mathbb{A}$, constructs a new Boolean attribute $\tilde{A}_l \epsilon \widetilde{\mathbb{A}}$.

Propositionalized attribute value $\tilde{V}_l$ associated with $\tilde{A}_l$ is a Boolean value $\epsilon\{True, False\}$, and a propositionalized data set $\tilde{D}$ is defined as $\tilde{D} \subseteq \tilde{V}_1 \times \tilde{V}_2 \times \ldots \tilde{V}_{|\widetilde{\mathbb{A}}|} \times \mathbb{C}$. Attribute $\tilde{A}_l$ of a propositionalized instance $\tilde{I}$ has a value $True$ if and only if the original instance I has the corresponding attribute value v_i^j. Let $\tilde{T}$ denote PAT defined over the Boolean attributes $\epsilon \widetilde{\mathbb{A}}$ propositionalized from $\mathbb{A}$, and let $Root(\tilde{T})$ denote the root of $\tilde{T}$. We represent a set of leaves of $\tilde{T}$ as $Leaves(\tilde{T}) \subseteq \widetilde{\mathbb{A}}$. The internal nodes of the tree correspond to abstract values of $\widetilde{\mathbb{A}}$.

Cut γ is a subset of nodes in taxonomy $\tilde{T}$ satisfying the following two properties:

(1) For any leaf $x \in Leaves(\tilde{T})$, either $x \in \gamma$ or x is a descendant of a node in $\tilde{T}$.

(2) For any two nodes $x, y \in \gamma$ is neither a descendant nor an ancestor of y.

Cut γ of $\tilde{T}$ induces a partition of propositionalized attributes $\widetilde{\mathbb{A}}$.

B. Creating Taxonomy with PAT-Learner

Hierarchical clustering is used (from the bottom to up direction, HAC – Hierarchical Agglomerative Clustering) for Boolean attributes based on the class value division together with pseudo-attributes when they have “is” value. Taxonomy $\tilde{T}$ is created by first adding $\widetilde{\mathbb{A}}$ attributes as the leaves to the taxonomy, then recursively one by one adding nodes to the taxonomy by combining two existing nodes.

To create the propositionalized attribute taxonomy, the following algorithm is used (see Table I) [12].

TABLE I

AN OUTLINE OF PAT-LEARNER ALGORITHM PSEUDO-CODE

(1) **Input**: Propositionalized data set $\widetilde{D}$.
(2) For each attribute $\tilde{A}_i \in \tilde{\mathbb{A}}$:
(3) For each class $C_k \epsilon C$:
(4) Estimate the probability distribution $p\left(c_k \middle| \tilde{A}_i\right)$.
(5) Let $P\left(C \middle| \tilde{A}_i\right) = \left\{p\left(c_1 \middle| \tilde{A}_i\right), \ldots, p\left(c_k \middle| \tilde{A}_i\right)\right\}$ be the class distribution associated with the attribute $\tilde{A}_i$.
(6) $\gamma \leftarrow \tilde{\mathbb{A}}$
(7) Initialize $\tilde{T}$ with nodes in γ.
(8) Iterate until $|\gamma| = 1$:
(9) Find $(x, y) = argmin\{DM(P(C|x)||P(C|y))\}$, where $x, y \in \gamma$ and $x \neq y$
(10) Merge x and y to create a new value z.
(11) Calculate probability distribution $P(C|z)$.
(12) $\hat{\gamma} \leftarrow \gamma \cup \{z\}\{x, y\}$.
(13) Update T_{Σ_p} by adding nodes z as a parent of x and y
(14) $\gamma \leftarrow \hat{\gamma}$.
(15) **Output**: T_{Σ_p}
end.

First, the probability distribution for classes on the attribute value "1" is calculated for the attribute and it is considered as the class distribution in relation to attribute $\tilde{A}_i$ (1).

After that, from the full pseudo-attribute data set cut γ is formed and taxonomy $\tilde{T}$ from this cut is initialized (6–7).

Then the iterations are done by reducing the cut until the size of cut = 1 (8):

- Pseudo-attribute couple with the smallest divergence measure is found and these pseudo-attributes are combined into a new value (9–10);
- To combine the nodes, they must be similar. To determine the similarity, the pair comparison measuring with the deviation between the attribute positive value evaluation to the class is performed. When two attributes are equal, comparing their distributions to the class values, they will give a statistically similar informative example in classification;
- The probability distribution for classes to the new attribute value is calculated (11);
- A new cut, which includes z attribute, is created but combined attributes are removed (12);
- The attribute z as a parent to combined attributes is added to taxonomy $\tilde{T}$ (13);
- The main cut is replaced by the created cut (14).

Jeffreys-Kullback-Leibler divergence measure – dissimilarity between two probability distributions – is used in the shown algorithm:

$$J(P||Q) = K(P||Q) + K(Q||P) = \sum (p_i - q_i) \log(\frac{p_i}{q_i}), \quad (1)$$

where P and Q are probability distributions. Kulback-Leibler divergence (relative entropy or relative information) is represented with $K(P||Q)$ and it is calculated with the following formula:

$$K(P||Q) = \sum p_i \log\left(\frac{p_i}{q_i}\right). \quad (2)$$

J-divergence is appropriate information measure in this situation, because unlike Kulback-Leibler divergence, this measure is symmetrical, which means that $J(P||Q) = J(Q||P)$.

C. Decision Tree Creation with PAT-DTL Algorithm

Decision tree creation with PAT-DTL algorithm [12] basically converts the acquired pseudo-attribute data sets according to the acquired possible cuts of pseudo-attribute taxonomy and evaluates the acquired data set ability to create the rules with the help of C4.5 algorithm. Conceptual working model of the algorithm is shown in Table II.

TABLE II

AN OUTLINE OF PAT-DTL ALGORITHM PSEUDO-CODE

begin
(1) **Input**: Propositionalized data set $\widetilde{D}$ and propositionalized attribute taxonomy $\tilde{T}$
(2) Initialize cut γ to the leaves of $\tilde{T}$, $Leaves(\tilde{T})$
(3) Estimate class conditional frequency counts and generate the hypothesis h_γ
(4) Repeat until no change in cut γ or $|\gamma| \leq 1$
(5) $\bar{\gamma} \leftarrow \gamma$
(6) $p_\gamma \leftarrow a$ set of parents of all elements in γ
(7) For each node $v \epsilon p_\gamma$:
(8) Generate an abstraction γ^v of γ by substituting v for the children $\epsilon\gamma$ of γ
(9) Construct corresponding hypothesis h_{γ^v} using C4.5 algorithm.
(10) If $CV5(h_{\gamma^v}) > CV5(h_{\bar{\gamma}})$, then replace $\bar{\gamma}$ with γ^v .
(11) If $\gamma \neq \bar{\gamma}$ then $\gamma \leftarrow \bar{\gamma}$
(12) **Output**: h_γ
end.

First, the cut at pseudo-attribute set leaves is initialized and the probability distribution for classes on the attribute value "1" is calculated for all the attributes the same as in taxonomy creation.

The problem of learning classifiers from PAT and propositionalized data is an extension of the problem of learning classifiers from the data. In this context, a classifier is a hypothesis. Then, the task of learning classifiers from the data is to induce a hypothesis that satisfies the given criteria [12]. The hypothesis is generated for the cut using C4.5 algorithm. Then the iterations are done, reducing the cut until the size of cut = 1 or the cut is not changed (4):

- A new cut $\bar{\gamma}$ (basic cut) is created from the cut γ.
- Set of parents of the cut is introduced.
- For each element in the set of parents:
  - a new cut is generated that contains all elements of the basic cut and this parent element that replaces its children in the basic cut;
  - the hypothesis for the basic cut is generated using C4.5 algorithm;
  - the accuracy of the basic cut and the new cut are compared; if the new cut is more appropriate for classification, then the basic cut is replaced with the new cut;

- If in the beginning the cut and the basic cut do not match, then in the beginning the cut is replaced with the basic cut.

Completing the algorithm work, eliminate the best hypothesis.

V. Application of Methodology

The general idea of the developed methodology is shown in Fig. 2.

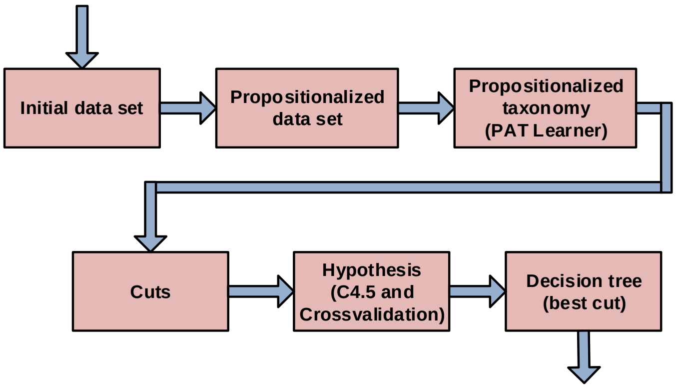

Fig. 2. PAT-DTL flowchart.

Decision tree learning, using pseudo-attribute taxonomy, consists of the following stages:

1. Pre-treatment of the data sets.
2. Pseudo-attribute data set creation from the data set attributes.
3. Pseudo-attribute taxonomy creation.
4. The decision tree learning and test performance.

Stage 1. In practice, it is possible to use a wide range of data sets; however, the following limitations are determined:

• the data set must be intended for carrying out the classification tasks (it must contain attribute columns and class column);

• the number of data set attributes and entries should be mutually proportionate, meaning that the data set cannot have more attributes than entries;

• the data set must be full or the missing values should make a small percentage of all values. If the data set contains the missing values, then they should be replaced. If the data set contains continuous attributes, they must be converted into discrete intervals.

For example, Iris data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals. One of the classes contains Iris setosa, while the other class contains both Iris virginica and Iris versicolor and is not separable without the species information.

Stage 2. Pseudo-attribute set is created as follows: first, for each $\tilde{A}_i$ attribute this unique attribute value domain is found. Then, based on this unique value pseudo-attributes are designed that consist of attribute-value pairs and pseudo-attributes 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 $\tilde{T}$ 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 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 fulfilment 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 is no more "valid" parents.

The use of such methodology with Iris data set gives the following results (see Table III).

TABLE III

Results for Iris Data Set

	Accuracy	Tree size
Initial data set	96.00 %	11
Propositionalized data set	89.33 %	15
J-divergence	95.33 %	15

The use of PAT-DTL does not give improvements in test accuracy for Iris data set, the highest accuracy is for starting data set (96 %). It can also be seen that the use of PAT-DTL with J-divergence information measure in taxonomy creation does not give improvements for the tree size. The results for this data set can be affected by the method chosen for attribute transformation from continuous attributes into discrete attributes.

VI. Conclusion

The methodology developed in the article is one of many methodologies of such type used in decision tree learning and in ontology application, and the authors' task has been to explore suchlike methodology. In many cases, it is possible to increase classification accuracy (obtaining shorter rules and smaller size of decision tree) with the help of ontology in the developed classification systems. The selected data sets for the initial treatment and successful selection of similarity measure has also an important role. But in any case, the use of ontology in classification tasks, decision tree learning and analysis have great prospects. In the future research, the possibilities of this methodology will be evaluated comparing it to other similar methodologies.

Acknowledgement

We would like to offer our special thanks to Irena Tuleiko (Master's Degree in Information Technology, 2012–2014) for useful software implementation.

References

[1] T.M. Mitchell, *Machine learning*. McGraw-Hill, 1997, 414 p.

[2] J.R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[3] L. Rokach and O. Maimon, *Data mining with decision trees: theory and applications*. World Scientific Pub Co Inc., 2008.

[4] L. Breiman, J.H. Friedman, R. Olshen and C.J. Stone, *Classification and regression trees*. Belmont, CA: Wadsworth, 1984.

[5] D. Gašević, D. Djurić and V. Devedžić, Model driven architecture and ontology development. Springer-Verlag, 2006.

[6] F. Kharbat and H. El-Ghalayini, "Building Ontology from Knowledge Base Systems," in *Proc. of Int. Arab Conf. on Information Technology*, 2011.

[7] A. Suyama and T. Yamaguchi, "Specifying and learning inductive learning systems using ontologies," *Working Notes from the 1998 AAAI Workshop on the Methodology of Applying Machine Learning: Problem Definition, Task Decomposition and Technique Selection*, pp. 29–36, 1998.

[8] N. Guarino, "Formal Ontology in Information Systems," in *1st Int. Conf. on Formal Ontology in Information Systems*, FOIS, Trento, Italy, IOS Press, 1998, pp. 3–15.

[9] R. Kohavi and F. Provost, "Applications of data mining to electronic commerce," in *Applications of data mining to electronic commerce*, US: Springer, Data Mining Knowledge Discovery 5/1-2, 2001, pp. 5–10. http://dx.doi.org/10.1023/A:1009840925866

[10] T.R. Gruber, "A translation approach to portable ontologies," *Knowledge Acquisition*, vol. 5(2), pp. 199–220, 1993. http://dx.doi.org/10.1006/knac.1993.1008

[11] D. Kang and M. Kim, "Propositionalized attribute taxonomies from data for data-driven construction of concise classifiers," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12739–12746, 2011. http://dx.doi.org/10.1016/j.eswa.2011.04.062

[12] D. Kang and K. Sohn, "Learning decision trees with taxonomy of propositionalized attributes," *Pattern Recognition*, vol. 42, no. 1, pp. 84–92, 2009. http://dx.doi.org/10.1016/j.patcog.2008.07.009

[13] J.R. Quinlan, "Induction of Decision Trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986. http://dx.doi.org/10.1007/BF00116251

[14] "Matlab: The language of technical computing," [Online]. Available: http://se.mathworks.com/products/matlab/ [Accessed: September 30, 2015].

[15] "UCI Machine learning repository: Iris data set," [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Iris [Accessed: September 30, 2015].

[16] "Weka: Data mining software," [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/ [Accessed: September 30, 2015].

Peter Grabusts received his *Dr. sc. ing.* degree in Information Technology from Riga Technical university in 2006. Since 1996 he has been working at Rezekne Higher Education Institution. 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 and techniques for clustering.

E-mail: peter@ru.lv

Arkady Borisov 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 is a Professor of Computer Science at the Faculty of Computer Science and Information Technology, Riga Technical University (Latvia). His research interests include fuzzy sets, fuzzy logic and computational intelligence. He has 235 publications in the field. He has supervised a number of national research grants and participated in the European research project ECLIPS. He is a member of IFSA European Fuzzy System Working Group, Russian Fuzzy System and Soft Computing Association, honorary member of the Scientific Board, member of the Scientific Advisory Board of the Fuzzy Initiative Nordrhein-Westfalen (Dortmund, Germany).

E-mail: arkadijs.borisovs@cs.rtu.lv

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.

Address: 1 Kalku Str., LV-1010, Riga, Latvia.

E-mail: ludmila.aleksejeva_1@rtu.lv