LEARNING FUZZY RULES FROM FUZZY DECISION TREES

Karol MATIAŠKO, Ján BOHÁČIK, Vitaly LEVASHENKO, Štefan KOVALÍK

University of Žilina, Faculty of Management Science and Informatics, Slovak Republic
e-mail: Karol.Matiasko@fri.utc.sk, Jan.Bohacik@fri.utc.sk, Vitaly.Levashenko@fri.utc.sk, Stefan.Kovalik@fri.utc.sk

Abstract

Classification rules are an important tool for discovering knowledge from databases. Integrating fuzzy logic algorithms into databases allows us to reduce uncertainty which is connected with data in databases and to increase discovered knowledge’s accuracy. In this paper, we analyze some possible variants of making classification rules from a given fuzzy decision based on cumulative information. We compare their classification accuracy with the accuracy which is reached by statistical methods and other fuzzy classification rules.

Keywords: fuzzy rules, fuzzy decision trees, classification, fuzzy logic, decision making, machine learning

1 INTRODUCTION

Now, data repositories are growing quickly and contain huge amount of data from commercial, scientific, and other domain areas. According to some estimates, the amount of data in the world is doubling every twenty months [7]. Because of it, information and database systems are widely used. The current generation of database systems is based mainly on a small number of primitives of Structured Query Language (SQL). The database systems usually use relational databases which consist of relations (tables) [11]. The tables have attributes in columns and instances in rows. Because of increasing data, these tables contain more and more covert information. The covert information can't be found out and transformed into knowledge by classical SQL queries any more. One approach how to extract knowledge from the tables is to search dependences from the data.

There are a lot of ways how to describe the dependences among data: statistical methods – Naïve Bayes Classifier, $k$-Nearest Neighbor Classifier etc., neural networks, decision tables, decision trees, classification rules.
Naïve Bayes Classifier (NBC) represents each instance as a $N$-dimensional vector of attribute values $[a_1, a_2, \ldots, a_M]$. Given that there are $m_c$ classes $c_1, \ldots, c_k, \ldots, c_{m_c}$, the classifier predicts a new instance (unknown vector $X = [a_1, a_2, \ldots, a_M]$) as belonging to the class $c_k$ having the highest posterior probability conditioned on $X$. In other words, $X$ is assigned to class $c_k$ if and only if $P(c_k/X) > P(c_j/X)$ for $1 \leq j \leq m_c$ and $j \neq k$. NBC is very simple, it requires only a single scan of the data, thereby providing high accuracy and speed for large databases. Their performance is comparable to that of decision trees and neural networks. However, there arise inaccuracies due to a) the simplified assumptions involved and b) a lack of available probability data or knowledge about the underlying probability distribution [13, 20].

$K$-Nearest Neighbor Classifier ($k$-NN) assumes all instances correspond to points in the $n$-dimensional space $R^n$. During learning, all points with known classes are remembered. When a new point is classified, the $k$-nearest points to the new point are found and are used with a weight for determining the new point’s class. For the sake of increasing classification accuracy, greater weights are given to closer points during classification [12]. For large $k$, the algorithm is computationally more expensive than a decision tree or neural network and requires efficient indexing. When there are lots of instances with known classes, a reduction of their number is needed.

A neural network consists of a system of interconnected neurons. A neuron, in a simplified way, takes positive and negative stimuli (numerical values) from other neurons and when the weighted sum of the stimuli is greater than given threshold value, it activates itself. The neuron’s output value is usually a non-linear transformation of the sum of stimuli. In more difficult models, the non-linear transformation is adapted by some continuous functions. Neural networks are good for classification. They are an alternative to decision trees or rules when understandable knowledge is not required. A disadvantage is that it is sometimes difficult to determine the optimal number of neurons [14].

A decision table is the simplest, most rudimentary way of representing knowledge. It has instances in rows and attributes in columns. The attributes describe some properties of the instances. One (or more if it’s needed) of the attributes are considered as class attribute. It’s usually the last column’s attribute. The class attribute classifies the instances into a class - an admissible value of the class attribute. A decision table is simply makeable and useable, e. g. a matrix. On the other hand, such a table may become extremely large for complex situations and it’s usually hard to decide which attributes to leave out without affecting the final decision [15].

A group of classification rules is also a popular way of knowledge representation. Each rule in the group usually has the following form: IF Condition THEN Conclusion. Both Condition and Conclusion contain one or more “attribute is attribute’s value” and there are operators among them, such as OR, AND. All classification rules in a group have in their conclusions only the same class attributes. A group of rules is used for classifying instances into classes [2].
Another knowledge representation is a (crisp) decision tree. Such a decision tree consists of several nodes. There are two types of nodes: a) the root and internal nodes, b) leaves. The root and internal nodes are associated with a non-class attribute.

Leaves are associated with a class. Basically, each non-leaf node has an outgoing branch for each its attribute’s possible. To classify a new instance using a decision tree, beginning with the root, successive internal nodes are visited until a leaf is reached. At the root or each internal node, the test for the node is applied to the instance. The outcome of the test at an internal node determines the branch traversed, and the next node visited. The class for the instance is simply the class of the final leaf node [5]. Thus, the conjunction of all the conditions for the branches from the root to a leaf constitutes one of the conditions for the class associated with the leaf. It implicates a decision tree can be represented by classification rules which use AND operator.

The paper is organized as follows. Section 2 explains why it’s needed to integrate fuzzy logic into FDTs and fuzzy rules. It also offers a survey of FDTs induction’s methods. Section 3 describes how to make fuzzy classification rules from FDTs based on cumulative information and how to use them for classification. Section 4 demonstrates our study’s experimental results on several databases. Section 6 concludes this work.

2 DATA IMPERFECTION AND FUZZY DECISION TREES

To be able to work with real data it’s needed to consider cognitive uncertainties. Because of it, data aren’t often accurate and contain some uncertainty. These uncertainties are caused by various reasons, such as a) there is no reliable way how to measure something – e.g. a company’s market value, a person’s value, etc., b) it can be too expensive to measure something exactly, c) vagueness – it is associated with humans’ difficulty of making sharp boundaries – e.g. it’s strange for people to say it’s hot when it’s 30 °C and it’s not hot when it’s 29.9°C, d) ambiguity – it is associated with one-to-many relations, i.e., situations with two or more alternatives such that the choice between them is left unspecified. These problems with real data have been successfully solving thanks to fuzzy sets and fuzzy logic for several years. Fuzzy sets are a generalization of (crisp) sets. In the case of a set, an element \( x \) either is the set’s member or isn’t. In the case of a fuzzy set, an element \( x \) is the set’s member with given membership degree which is denoted by membership function \( \mu(x) \). \( \mu(x) \) has a value in the continuous interval 0 to 1 [23]. Fuzzy sets are one of main ideas of Fuzzy Logic, which is considered as an extension of Boolean Logic as well as Multi-Valued Logic [21].

Decision trees, which make use of fuzzy sets and fuzzy logic for solving the introduced uncertainties, are called Fuzzy decision trees (FDTs). It has been developed several algorithms for making FDTs so far. The algorithms usually make use of principles of ID3 algorithm which was initially proposed by Quinlan [16] and which was meant for making crisp decision trees. The algorithm’s a top-down greedy heuristic, which means that makes locally optimal decisions at each node. The
algorithm uses minimal entropy as a criterion at each node. So-called Fuzzy ID3 is a
direct generalization of ID3 to fuzzy case. Such an algorithm uses minimal fuzzy
entropy as a criterion, e. g. [19, 17]. Notice that there are several definitions of fuzzy
entropy [1]. A variant of Fuzzy ID3 is [22] which uses minimal classification
ambiguity instead of minimal fuzzy entropy. An interesting investigation to ID3-based
algorithms, which uses cumulative information estimations, was proposed in [10].
These estimations allow us to build FDTs with different properties (unordered,
ordered, stable, etc.). There are also some algorithms, which are quite different from
ID3. For example, Wang et al. [18] present optimization principles of fuzzy decision
trees based on minimizing the total number and average depth of leaves. The algorithm
minimizes fuzzy entropy and classification ambiguity at node level, and uses fuzzy
clustering so as to merge branches. Another non-Fuzzy ID3 algorithm is discussed in
[4]. The algorithm uses look-ahead method. Its goal is to evaluate so-called
classifiability of instances that are split along branches of a give node.

One way how to make fuzzy rules is a transformation of fuzzy decision trees. In a
crisp case, each leaf of the tree corresponds to one rule. The condition of the rule is a
group of “attribute is attribute’s value” which are connected with operator AND. These
attributes in the condition are attributes associated with the nodes in the path from the
root to a considered leaf. The attribute’s values are the values associated with outgoing
branch of a particular attribute’s node in the path. The conclusion of the rule is “class
attribute is class”. The conclusion is the output for the condition. In other words, it’s
classification for a known instance whose values correspond to the condition. This
notion is based on rules made from crisp decision tree [16]. In this paper in section 3,
we describe a generalization for a fuzzy case. For a fuzzy case, the class values for a
new instance (or, more exactly, membership function’s values for particular classes)
are described by several rules which belong to one or more leaves.

3 FUZZY CLASSIFICATION RULES INDUCTION.

The main goal of a made group of classification rules is to determine the value of
the class attribute from the attribute’s values of a new instance. In this section, we
explain the mechanism of making fuzzy rules from FDTs and the mechanism of their
use for classification. To explain the mechanisms, a FDT which is made on basis of
cumulative information is considered [10].

Let the FDT be made from a database whose instances are described by \( N \) attributes \( A = \{A_1, ..., A_i, ..., A_N\} \). Let an attribute \( A_i \) take \( m_i \) possible values \( a_{i,1}, a_{i,2}, \ldots, a_{i,m_i} \), which is denoted by \( A_i = \{a_{i,1}, ..., a_{i,j}, ..., a_{i,m_i}\} \). In the FDT, each non-leaf
node is associated with an attribute from \( A \). When \( A_i \) is associated with a non-leaf
node, the node has \( m_i \) outgoing branches. The \( j \)-th branch of the node is associated with
value \( a_{i,j} \in A_i \). In general, the database can contain several class attributes which
depend on non-class attributes. Because of simplicity, only one class attribute \( C \) is
considered here. The class attribute \( C \) has \( m_c \) possible values \( c_1, ..., c_k, ..., c_{m_c} \), which is
denoted by \( C = \{c_1, ..., c_k, ..., c_m\} \). Let the FDT have \( R \) leaves \( \mathcal{L} = \{l_1, ..., l_r, ..., l_R\} \). It is also considered there is a value \( F_{k}^{\prime} \) for each \( r \)-th leaf \( l_r \) and each \( k \)-th class \( c_k \). This value \( F_{k}^{\prime} \) means the certainty degree of the class \( c_k \) attached to the leaf node \( l_r \).

Now, the transformation of this FDT into fuzzy rules is described. In fuzzy cases, a new instance \( e \) may be classified into different classes with different degrees. In other words, the instance \( e \) is classified into each possible class \( c_k \) with a degree. For that reason, a separate group of classification rules can be considered for each class \( c_k \), \( 1 \leq k \leq m \). Let’s suppose the group of classification rules for \( c_k \). Then, each leaf \( l_r \in \mathcal{L} \) corresponds to one \( r \)-th classification rule. The condition of the \( r \)-th classification rule is a group of “attribute is attribute’s value” which are connected with operator AND. These attributes in the condition of \( r \)-th rule are attributes associated with the nodes in the path from the root to the \( r \)-th leaf. The attribute’s values are the values associated with the respective outgoing branches of the nodes in the path. The conclusion of the \( r \)-th rule is “\( C \) is \( c_k \)”. Let’s consider that in the path from the root to the \( r \)-th leaf there are \( q \) nodes associated with attributes \( A_{i_1}, A_{i_2}, \ldots, A_{i_q} \) and respectively their \( q \) outgoing branches associated with the values \( a_{i_1,j_1}, a_{i_2,j_2}, \ldots, a_{i_q,j_q} \). Then the \( r \)-th rule has the following form:

\[
\text{IF } A_{i_1} \text{ is } a_{i_1,j_1} \text{ AND } A_{i_2} \text{ is } a_{i_2,j_2} \text{ AND } \cdots \text{ AND } A_{i_q} \text{ is } a_{i_q,j_q} \text{ THEN } C \text{ is } c_k \text{ (with truthfulness } F_{k}^{\prime}).
\]

In the two following paragraphaps, two ways how to use these fuzzy classification rules for classification of a new instance \( e \) into a class \( c_k \), \( 1 \leq k \leq m \), are described. The first way uses only one classification rule whereas the second uses one or more rules for the classification. The classification means that the degree the instance \( e \) is classified into the class \( c_k \) with is denoted by membership function \( \mu_{c_k}(e) \). \( \mu_{c_k}(e) \) has a value in the continuous interval 0 to 1. It is also supposed that all attribute’s values’ membership functions of the new instance \( e \), \( \mu_{a_{i,j}}(e) \), \( a_{i,j} \in A_{i} \), \( A_{i} \in \mathcal{A} \), are known.

The first way is based on [16] and was initially meant for classification in crisp cases. In crisp case, an attribute’s possible value either is the attribute’s value or isn’t. In other words, either \( \mu_{a_{i,j}}(e)=1 \) or \( \mu_{a_{i,j}}(e)=0 \) for \( a_{i,j} \in A_{i} \), \( A_{i} \in \mathcal{A} \). Also, just one \( \mu_{a_{i,j}}(e) \) is equal 1 for \( a_{i,j} \in A_{i} \), all the others are equal 0. In fuzzy case, there can be more than one \( \mu_{a_{i,j}}(e)>0 \) for \( a_{i,j} \in A_{i} \). To be able to use this first way, \( \mu_{a_{i,j}}(e) \) with the maximal value among all \( a_{i,j} \in A_{i} \) can be set equal 1 and all the others equal 0. The instance with such membership function’s values is called rounded one. Then one rule is chosen from the made group of rules. It is the \( r \)-th rule whose attributes’ values \( a_{i_1,j_1}, a_{i_2,j_2}, \ldots, a_{i_q,j_q} \) respectively match with \( a_{i,j} \) with \( \mu_{a_{i,j}}(e)=1 \) of the rounded instance. After that, \( \mu_{c_k}(e) \) is set equal \( F_{k}^{\prime} \).
The second way uses one or more classification rules from \( c_k \)'s group for classification of \( e \) into a class \( c_k \). The reason why it is like this is that there may be several \( \mu_{a_{i,j}}(e) > 0 \) for a node in the FDT. That’s why, there may be several paths whose all outgoing node’s branches are associated with \( a_{i,j} \) where \( \mu_{a_{i,j}}(e) > 0 \). And each this path corresponds to one classification rule. But because of the \( \mu_{a_{i,j}}(e) \) are not all equal 1, it is clear that such each rule should be included in the final value of \( \mu_{c_k}(e) \) with a certain weight. The weight is for instance \( e \) and the \( r \)-th rule in \( c_k \)'s group given by the following form:

\[
W_r(e) = \prod_{a \in \text{path}_r} \mu_a(e),
\]

where \( \mu_a(e) \) is the value of the membership function of a attribute’s value \( a \) and \( \text{path}_r \) is a set of all attribute’s values in the condition of \( r \)-th classification rule. The weight \( W_r(e) \) is equal 0 if there is a attribute’s value \( a \) whose \( \mu_a(e) = 0 \) in the condition of the \( r \)-th rule. The membership function’s value for the new instance \( e \) and for class \( c_k \) is:

\[
\mu_{c_k}(e) = \sum_{r=1}^{R} F_k^r \ast W_r(e),
\]

where \( F_k^r \) is the truthfulness of the \( r \)-th rule, or in other words, the certainty degree of the class \( c_k \) attached to the leaf node \( l_r \).

If, in the first or the second way, classification only into one class is needed, instance \( e \) is classified into the class \( c_k \) whose \( \mu_{c_k}(e) \) \( k=1, 2, \ldots, c_m \) is maximal.

Let us describe the process of transformation of the FDT into fuzzy rules and the two mentioned ways of their use for classification with the following examples.

**Example 1** An instance is described by \( N=4 \) attributes \( A = \{A_1, A_2, A_3, A_4\} = \{\text{Outlook}, \text{Temperature}, \text{Humidity}, \text{Wind}\} \) and the instances are classified with one class attribute \( C \). Each attribute \( A_i = \{a_{i,1}, \ldots, a_{i,j}, \ldots, a_{i,m_i}\} \) is defined as follows: \( A_1 = \{a_{1,1}, a_{1,2}, a_{1,3}\} = \{\text{sunny}, \text{cloudy}, \text{rain}\}, A_2 = \{a_{2,1}, a_{2,2}, a_{2,3}\} = \{\text{hot}, \text{mild}, \text{cool}\}, A_3 = \{a_{3,1}, a_{3,2}\} = \{\text{humid}, \text{normal}\}, A_4 = \{a_{4,1}, a_{4,2}\} = \{\text{windy}, \text{not windy}\} \). The class attribute \( C = \{c_1, \ldots, c_k, \ldots, c_{m_c}\} \) is defined as follows: \( C = \text{Plan} = \{c_1, c_2, c_3\} = \{\text{volleyball, swimming, weight lifting}\} \). There is a FDT in Fig.1. The FDT was made from a database by using cumulative information. The database and the mechanism how to build it are described in [9]. The FDT has \( R=9 \) leaves. In the leaves there are written values of \( F_k^r \) for respective \( c_k \), \( k=1, 2, 3 \). Our goal is to determine \( \mu_{c_k}(e), k=1, 2, 3 \) for a new instance \( e \) which is described in the Tab.1 on basis of the FDT.
Figure 1 An example of a FDT adopted from [9].

Table 1 A new instance $e$ with unknown $\mu_{c_k}(e)=?$, $k=1, 2, 3$.

<table>
<thead>
<tr>
<th>Attributes and their values</th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$A_3$</th>
<th>$A_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu_{a_{i,j}}(e)$</td>
<td>0.9</td>
<td>0.1</td>
<td>0.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

The classification rules for class $c_1$ ($c_1$; group of rules) made from the FDT in Fig.1. have the following form:

$r=1$: IF Temp is hot AND Outlook is sunny AND Wind is windy THEN Plan is volleyball (truthfulness $F_1^1 = 0.142$)

$r=2$: IF Temp is hot AND Outlook is sunny AND Wind is not windy THEN Plan is volleyball (truthfulness $F_1^2 = 0.332$)
r=3: IF Temp is hot AND Outlook is cloudy THEN Plan is volleyball (truthfulness $F_1^3 = 0.376$)

...  

r=9: IF Temp is cool THEN Plan is volleyball (truthfulness $F_1^9 = 0.163$)

Similarly, $c_2$’ group of 9 rules and $c_3$’ group of 9 rules can be determined.

The following Examples 2, 3 show how to use the fuzzy classification rules.

**Example 2 (classification with one fuzzy rule).** First, instance $e$ in Tab.1 is transformed into rounded instance $e$ in Tab.2.

**Table 2** The rounded instance $e$.

<table>
<thead>
<tr>
<th>Attributes and their values</th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$A_3$</th>
<th>$A_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$a_{1,2}$</td>
<td>$a_{1,3}$</td>
<td>$a_{2,1}$</td>
<td>$a_{2,2}$</td>
</tr>
<tr>
<td>$\mu_{a_{ij}}(e)$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

It means that, for each $A_i$ ($i=1, 2, 3, 4$), $\mu_{a_{ij}}(e)$ ($j=1, ..., m_i$) with the maximal value among all $a_{ij} \in A_i$ is set equal 1 and all the others equal 0. When $a_{ij}$ with $\mu_{a_{ij}}(e)=1$ are chosen, the following set $\{a_{1,1}, a_{2,1}, a_{3,1}, a_{4,2}\}$ = \{sunny, hot, humid, not windy\} is obtained. The elements of the set matches with attributes’ values of fuzzy classification rule $r=2$ in Example 1. Therefore, $\mu_{c_1}(e)$ for class $c_1$ is computed by $\mu_{c_1}(e) = F_1^2 = 0.332$. Similarly, $\mu_{c_2}(e) = F_2^2 = 0.623$ on bases of $c_2$’ group of rules, $\mu_{c_3}(e) = F_3^2 = 0.045$ on bases of $c_3$’ group of rules. The maximum value has $\mu_{c_2}(e)$. Because of it, if classification only into one class is needed, instance $e$ is classified into class $c_2$.

**Example 3 (classification with several fuzzy rules).** Paths, whose all outgoing node’s branches which are associated with $a_{ij}$ whose $\mu_{a_{ij}}(e)>0$, are shown in several –bold branches in Fig.1. In this situation, it’s needed to use rules which correspond to leaves $l_1$, $l_2$, $l_3$. The respective classification rules $r = 1, 2, 3$ for class $c_1$ are mentioned in Example 1. For the rule $r=1$, path$ _r = \{a_{2,1}, a_{1,1}, a_{4,1}\}$. And so, for instance $e$ described in Tab.1., $W_1(e) = 0.9*1.0*0.4 = 0.36$. Similarly, $W_2(e) = 0.9*1.0*0.6= 0.54$, $W_3(e) = 0.1*1.0 = 0.1$. All the other $W_r(e)$ are equal 0, $r=4, 5, ..., 9$.

Then, $\mu_{c_1}(e) = 0.142*0.36 + 0.332*0.54 + 0.376*0.1 + 0.11*0 + ... + 0.163*0 = 0.268$.

When it’s done for $c_1$, $c_2$, $\mu_{c_2}(e) = 0.631$ and $\mu_{c_3}(e) = 0.101$ will be obtained. The maximum value has $\mu_{c_2}(e)$. And so, if classification only into one class is needed, instance $e$ is classified into class $c_2$. 


4 EXPERIMENTAL RESULTS

The main purpose of our experimental study is to compare the two mentioned ways of using fuzzy classification rules made from FDT based on cumulative information for classification. Their classification accuracy is also compared with other methods. They all have been coded in Java and the experimental results are obtained on AMD Turion64 1.6 GHz with 1024 MB of RAM.

The experiments have carried out on selected Machine Learning databases [6]. First, if it was needed, the databases were fuzzyfied with an algorithm introduced in [8]. Then we separated each database into 2 parts. We used the first part (70% from the database) for building classification models. The second part (30% from initial database) was used for verification these models. This process of separation and verification was repeated 100 times to obtain the model’s error rate for respective databases. The error rate is calculated as the ratio of the number of misclassification combinations to the total number of combinations.

The results of our experiments are in Tab.3, where CA-MM denotes FDT-based fuzzy classification rules algorithm proposed in [22], CA-SP denotes a CA-MM’s modification which is introduced in [3], CI-RM-M denotes algorithm [10] for making FDT with cumulative information and which uses the second way of fuzzy classification rules induction from previous section 3, CI-RM-0 uses the first way of fuzzy classification rules induction from previous section 3, NBC denotes Naïve Bayes Classifier, k-NN denotes k-Nearest Neighbour Classifier. The numbers in round brackets denote the order of the method for respective database, number 1 denotes the least and also the best error rate. The last row contains average error rate for all databases and respective methods.

Table 3 The error rates’ comparison of respective methods.

<table>
<thead>
<tr>
<th>Database</th>
<th>Error rate for given database and classification method.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CA-MM</td>
</tr>
<tr>
<td>BUPA</td>
<td>0.4253(4)</td>
</tr>
<tr>
<td>Ecoli</td>
<td>0.3112(6)</td>
</tr>
<tr>
<td>Glass</td>
<td>0.4430(4)</td>
</tr>
<tr>
<td>Haberman</td>
<td>0.2615(3)</td>
</tr>
<tr>
<td>Iris</td>
<td>0.0400(2)</td>
</tr>
<tr>
<td>Pima</td>
<td>0.2509(4)</td>
</tr>
<tr>
<td>Wine</td>
<td>0.08170(6)</td>
</tr>
<tr>
<td>Average</td>
<td>0.2951(6)</td>
</tr>
</tbody>
</table>
In Figure 2, there is a normalized average error rate for respective methods. It is computed as: the respective average error rate / the highest average error rate * 100. It can be seen CI-RM-M which uses several classification rules for classification is almost 10% better than CI-RM-O and it’s the best of all methods compared here.

5 CONCLUSION

This paper investigated the mechanism of making fuzzy rules from FDTs based on cumulative information and two ways of their use for classification of a new instance into a class. One way is a direct fuzzy analogy of crisp case when one classification rule is used for the new instance’s classification. The other way is a generalization of the former. This uses several classification rules and each rule is included in the classification’s result with a certain weight. Both the ways were compared with each other. The generalized way has almost 10% better results than the other one on selected databases and also is the best of all compared methods.

ACKNOWLEDGEMENTS

This research was supported by the grants of Scientific Grant Agency of the Ministry of Education of Slovak Republic and the Slovak Academy of Science (grant ZU/05 VV_MVTS13). We would like to say thank you Dr. Irina Davidovskaja (from Belarusian State Economic University) for constructive comments.
REFERENCES


