A New Version of the Fuzzy-ID3 Algorithm

Łukasz Bartczuk[1] and Danuta Rutkowska[1,2]

[1] Department of Computer Science
Czestochowa University of Technology, Poland
[2] Department of Knowledge Engineering and Computer Intelligence
Academy of Humanities and Economics in Lodz, Poland
{bartczuk, drutko}@kik.pcz.czest.pl

Abstract. In this paper, a new version of the Fuzzy-ID3 algorithm is presented. The new algorithm allows to construct decision trees with smaller number of nodes. This is because of the modification that many different attributes and their values can be assigned to single leaves of the tree. The performance of the algorithm was checked on three typical benchmarks data available on the Internet.

1 Introduction

Decision trees are commonly used as knowledge representation and an approach to classification. They are appreciated for their clarity and high accuracy. Many algorithms designed for buiding decision trees have been proposed. The most popular are ID3 (Interactive Dichotimizer 3) introduced by Quinlan in 1986 [8], and its modifications, e.g. C4.5 [9]. Those algorithms allow to create decision trees from symbolic data, in an easy and effective way. Numerical data, when applied, must be splitted into limited number of disjoint intervals. The data present values of the attributes, i.e. features of objects to be classified.

In some classification problems, determination of crisp values of attributes is not possible or not fully correct. The solution of that problem is the use of the theory of fuzzy sets and fuzzy logic, introduced in 1965 by Lofti Zadeh [12]. Fuzzy sets may describe uncertain or imprecise phenomenona. The Fuzzy-ID3 algorithm [6],[7], created by Janikow in 1995, combines simplicity and clarity of decision trees with fuzzy sets which can define linguistic values and allow to use fuzzy intervals.

There are two main problems related to decisions trees and fuzzy decision trees. The first is the large size of the tree (number of nodes) in high dimensional classification tasks. The second problem concerns the structure of the tree. The tree created according to the ID3 or Fuzzy-ID3 algorithm is a proper structure for the data representation. However, it is not always the best solution. The main reason of these two problems is the manner in which the attribute (represented by a node) to be the best split is chosen; see Fig.1.

In this paper, a modified Fuzzy-ID3 algorithm is presented. According to this modification, more than one attribute, and more than one linguistic value of these attributes, may be assigned to single leaves (decision nodes). This

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1060–1070, 2006.

modification results in obtaining trees with smaller number of nodes. An example of such a tree is illustrated in Fig. 4.

The paper is organized as follows. Section 2 presents the ID3 and Fuzzy-ID3 algorithms. The modified version of the Fuzzy-ID3 algorithm is proposed in Section 3. Experimantal results are illustrated in Section 4. Final conclusions are included in Section 5.

In this paper, capital letters denote sets, for instance A, C, E - sets of attributes, classes, and examples, respectively. Cardinalities of the sets are denoted as $|A|$, $|C|$, $|E|$. Specific attributes, for $k = 1, \ldots, |A|$, are denoted as A^k, and A^k is a set of values of attribute A^k, so $A^k = \{a_l^k\}$, for $l = 1, \ldots, |A^k|$, and $A^k \in A$. Specific classes are denoted as c^j, for $j = 1, \ldots, |C|$, and $c^j \in C$. Examples are denoted as e^i, for $i = 1, \ldots, |E|$ and $e^i \in E$. Thus, every example is described as $e^i = [a_i^1, \ldots, a_i^{|A|}, c_i^j]$, where $c_i^j \in C$ is the class associated with e^i.

2 ID3 and Fuzzy-ID3 Algorithms

This section presents classical and fuzzy versions of the ID3 algorithm introduced in [8] and [6], respectively. Decision trees are techniques for partitioning examples into sets corresponding to decision rules.

2.1 ID3 Algorithm

The purpose of the ID3 algorithm is to create a tree structure from an example set, E, which contains values of attributes, A^k, for $k = 1, \ldots, |A|$, that characterize objects to be classified. In addition, every example includes the class, c^j, to which the object belongs. These examples are called training examples, and E is a training set. The tree structure can further be used for classification, data analysis or knowledge representation.

This algorithm employs the entrophy for determining the discriminatory power of each attribute. This is applied in order to determine the attribute that should be chosen to split the node associated with this attribute. The ID3 algorithm is based on the following assumptions [8]:

(1) The root node of the decision tree contains all training examples. Each node is reqursively split by partitioning its examples.
(2) Every training example belongs to class c^j with probability (the relative frequency):

$$p_j = \frac{\left|E_j^N\right|}{|E^N|} \tag{1}$$

where E^N - set of examples in node N, and E_j^N - set of examples that belong to class c^j in node N; $E_j^N \subset E^N \subset E$.
(3) For the data set in current node, N, we compute the information content:

$$I^N = -\sum_{j=1}^{|C|} p_j \log_2 p_j \tag{2}$$

(4) If an attribute, A^k, is chosen as a node, N, of the decision tree, the information to be supplied to the subtree corresponding to the node's branch, i.e. the path from parent (root) node N,$A^k = a_l^k$, to a child node, is denoted as $I^{N|a_l^k}$. The expected information required for the subtree with the attribute A^k in node N is determined as follows:

$$I^{N|A^k} = \sum_{l=1}^{|A^k|} \frac{\left|E_{a_l^k}^N\right|}{|E^N|} I^{N|a_l^k} \tag{3}$$

where $I^{N|A^k}$ is called the weighted entrophy, $E_{a_l^k}^N$ denotes the set of examples whose attribute value a_l^k corresponds to the node's branch.

(5) The information gained by branching on the attribute A^k at node N is:

$$G = I^N - I^{N|A^k} \tag{4}$$

The node is split using the most discriminatory attribute, whose information gain, determined using (4), is maximal.

The process of splitting tree nodes starts from the root node (as node N), then repeats, and the algorithm ends when all attributes appear on the path from the root node to the current node or when all examples in the node come from a unique class. The fulfillment of the second criterion can leads to overlearning effect. The threshold $\tau \in [0, 1]$ can be used to prevent that situation. If the ratio of the number of examples with the same class to all examples in node N is equal or greater than this threshold, the node became a leaf. The ID3 algorithm is presented, in many publications, e.g in [5],[6],[8],[9].

2.2 Fuzzy-ID3 Algorithm

In classical decision trees, created by the ID3 algorithm, atrributes can have only symbolic or discrete numerical values. In case of fuzzy decision trees attributes can also have linguistic values (eg. small, warm, low) represented by fuzzy sets. Fuzzy decision trees have been obtained as a generalisation of classical decision trees through application of fuzzy sets and fuzzy logic. The Fuzzy-ID3 algorithm is an extension of ID3 algorithm. The difference between these two algorithms is in the method of computing the example count in node N. In the Fuzzy-ID3 algorithm the total examples count, P^N, in node N are expressed as [6]:

$$P^N = \sum_{j=1}^{|D_C|} P_j^N \tag{5}$$

where: D_C - set of linguistic values for the decision attribute, $\mathbf{x}_i$ and y_i are input vector and output value, which correspond to attributes and class, respectively, and the examples count, P_j^N, for decision j (class c_j) is determined as follows:

$$P_j^N = \sum_{i=1}^{|E^N|} \mathsf{f}\left(\mu_s(\mathbf{x}_i), \mu_j(c_i)\right) \tag{6}$$

where: f - function employed to compute the value of fuzzy relation (e.g. min, prod) [2],[10],[11],[13], μ_s - membership function of Cartesian product of fuzzy sets that appear on the path from the root node to node N, and μ_j - membership function of fuzzy set that determines class c^j, for $j = 1 \ldots, |C|$.

Equations (2) and (3) in the Fuzzy-ID3 algorithm takes the forms:

$$I^N = -\sum_{j=1}^{|D_C|} \frac{P_j^N}{P^N} \log_2 \frac{P_j^N}{P^N} \tag{7}$$

$$I^{N|A^k} = \frac{\sum_{l=1}^{|A^k|} P^{N|a_l^k} I^{N|a_l^k}}{\sum_{l=1}^{|A^k|} P^{N|a_l^k}} \tag{8}$$

where: $P^{N|a_l^k}$ - total examples count in node N containing value a_l^k, assuming that attribute A^k is used to split the node N. Stopping criteria are the same as in the ID3 algorithm.

2.3 Illustration of Fuzzy Decision Trees

Suppose we want to build a tree to solve a binary classification task with two attributes (*attr1* and *attr2*), and three fuzzy sets (*low*, *medium*, *high*) defined for each attribute. We know that attribute *attr2* is relevant for the solution of this problem because all examples with value *low* for this atrribute belong to class *0* and with value *hi* belong to class *1*; see Fig. 1b.

Suppose also that the proportion of examples with these values in the data set is small. When we compute examples count for each class and total examples count, for each fuzzy set (attribute value) that can be associated with the child node of the *Root* node, we get the values shown in Table 1, where *lv* stands for "linguistic value" that is the attribute value.

If we compute the weighted entrophy (8) and information gain (4), for each attribute, we see that according to the algorithm the best attribute to split is *attr1*. The result is a correct representation of the data set, but this is not the

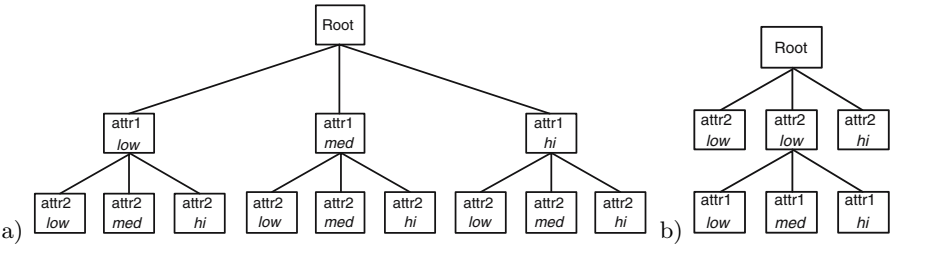

Fig. 1. Two possible trees: a) created by Fuzzy-ID3 algorithm, b) the better tree that can be created for the same problem

Table 1. The examples count for each class and total examples count in each fuzzy set that can be associated with the child node of the *Root* node

lv	$P_0^{Root\|lv}$	$P_1^{Root\|lv}$	$P^{Root\|lv}$
a_{low}^{attr1}	14.55	4.20	18.75
a_{med}^{attr1}	19.27	3.98	23.25
a_{hi}^{attr1}	4.58	49.95	54.53
a_{low}^{attr2}	13.23	0	13.23
a_{med}^{attr2}	23.39	49.85	73.24
a_{hi}^{attr2}	0	9.24	9.24

best solution that can be achieved for this problem. This is because the algorithm chooses the best split on average.

The tree created according to the Fuzzy-ID3 algorithm contains thirteen nodes and is depicted in Fig. 1a. The better tree that can be created for the same problem contains only seven nodes and is shown in Fig. 1b.

The solution of the problem mentioned above for the crisp ID3 algorithm has been presented by Friedman et. al.[4]. This algorithm, which is called the Lazy Decision Tree is very interesting for symbolic or numerical values of attributes, but it requires a process of creating of a new decision tree for every new example. In case of fuzzy values, many branches of the tree can be activated. This causes that number of computations that have to be performed may be too big to build a new tree for every new example. Therefore, this solution can be inefficient for fuzzy decision trees.

3 Fuzzy Decision Trees with Multi-Attribute Leaves

In this section, a new version of Fuzzy ID3 algorithm is proposed. The classical algorithms, described in Section 2 have been designed to create decision trees with nodes that represent only one attribute value. This algorithm allows to use more than one attribute value in leaves, so the decision trees contain less number of the nodes. This algorithm can be called MAL Fuzzy ID3, or MAL FID3, for short, where MAL stands for Multi-Attribute Leaves.

3.1 MAL Fuzzy ID3 Algorithm

The proposed algorithm introduces some modifications to the tree structure and to the procedure of creating the tree. We assume that there can be more than one linguistic value in the leaves of the tree, and also that there can be values of different attributes. This modification allows the use of all values of attributes that give unambiguous classification as a child of the current node. The membership of the example, in such a node, can be computed as the maximum values of the membership functions describing fuzzy sets in this node. We can also use other

s-norms [10],[11],[2],[13], than the maximum. However, we will achieve better results when we apply an arithmetic mean value of membership functions.

Let F^N denotes a set of values of attributes which can be used to split node N, and E^N - set of examples that have nonzero membership in node N.

The proposed algorithm is shown below:

Step 1: In the root of the tree, assume: $F^N = A$ and $E^N = E$

Step 2: From set F^N choose these linguistic values that give unambiguous classification, i.e. for $a_l^k \in F^N$; let us define

$$\Theta_j^N = \left\{ a_l^k : \frac{P_j^{a_l^k}}{\sum\limits_{m=1,..,|C|} P_m^{a_l^k}} > \tau \right\} \quad \text{for} \quad j = 1, \ldots, |C| \qquad (9)$$

where $P_j^{a_l^k}$ - total example count for class c^j, and attribute value a_l^k. For each nonempty set Θ_j^N create a new node. Linguistic values from sets Θ^N will not be taken into consideration for further spliting of the nodes.

Step 3: From set E^N, choose those examples, $e^i \in E^N$ for which the arithmetic mean value of membership of fuzzy sets describing linguistic values from Θ_j^N is smaller than threshold $\sigma \in [0, 1]$, that is

$$\Psi^N = \left\{ e^i : \frac{\sum\limits_{a_l^k \in \Theta_j^N} \mu_{a_l^k}(e^i)}{|\Theta_j^N|} < \sigma \right\} \quad \text{for} \quad j = 1, \ldots, |C| \qquad (10)$$

Step 4: For examples from set Ψ^N, compute information content according to (7).

Step 5: Compute weighted entrophy (8) for all attributes from F^N, and theirs values which are not included in Θ_j^N; $\quad j = 1, \ldots, |C|$.

Step 6: Select the attribute maximizing the information gain, G, and split the node N, using this attribute.

Step 7: For the nodes created in step 6, set $F^{N+1} = F^N \setminus \bigcup\limits_{j=1,\ldots,|C|} \Theta_j^N$ and $E^{N+1} = \Psi^N$

Step 8: Repeat steps from 2 to 8 until the stopping criteria are not fulfilled.

3.2 Illustration of MAL FID3 Algorithm on IRIS Data

The application of this algorithm will be presented on the Iris classification problem [1],[3]. We have to split the iris flowers into three classes representing iris spieces. Each example is described by four attributes (width and length of petal and width and length of sepal).

The data set E consists of 150 examples spliting into three classes: Setosa, Versicolour and Virginica (50 examples from each class). Distribution of the

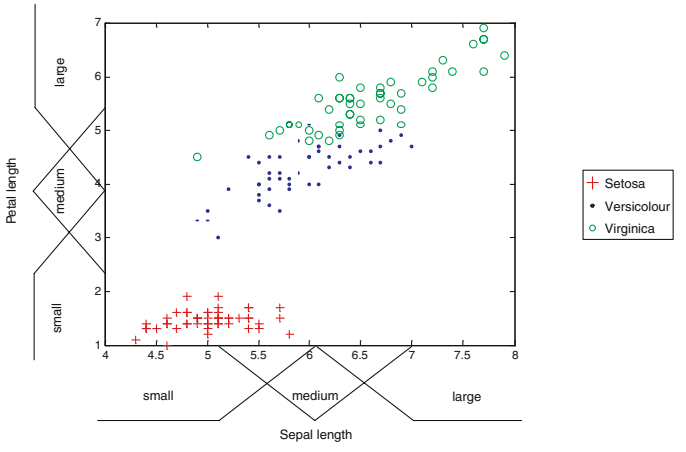

Fig. 2. Distribution of examples for Iris classification problem; for attributes: Sepal length and Petal length

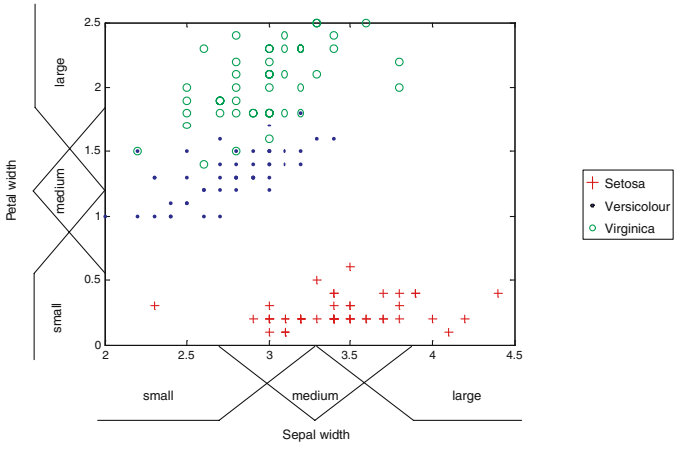

Fig. 3. Distribution of examples for Iris classification problem; for attributes: Sepal width and Petal width

examples in the attribute space is presented in Figs. 2 and 3. For each attribute, three fuzzy sets (representing values: small, medium, large) are defined.

In the begining the set of attribute values contains the following elements:

$$F^{Root} = \left\{ \begin{array}{lll} a^{PL}_{small}; & a^{PL}_{medium}; & a^{PL}_{large}; \\ a^{PW}_{small}; & a^{PW}_{medium}; & a^{PW}_{large}; \\ a^{SL}_{small}; & a^{SL}_{medium}; & a^{SL}_{large}; \\ a^{SW}_{small}; & a^{SW}_{medium}; & a^{SW}_{large}; \end{array} \right\}$$

where PL, PW, SL, SW stands for petal length, petal width, sepal lenght, sepal width, respectively.

Set E^{Root} includes all 150 examples from set E. According to equations (5) and (6), we compute the ratio of examples count for each class to total examples count, for fuzzy sets describing liguistic values from F^{Root}. The results are presented in Table 2.

Table 2. The ratio of examples count from each class to total examples count for all linguistic values

Sepal length (SL)				Sepal width (SW)			
Value	*Setosa*	*Versicolour*	*Virginica*	*Value*	*Setosa*	*Versicolour*	*Virginica*
small	0.704	0.2432	0.0523	small	0.0766	0.5457	0.3775
medium	0.070	0.5053	0.4239	medium	0.3695	0.2705	0.3599
large	0	0.2405	0.7594	large	0.8231	0.0204	0.1564
Petal length (PL)				**Petal width (PW)**			
Value	*Setosa*	*Versicolour*	*Virginica*	*Value*	*Setosa*	*Versicolour*	*Virginica*
small	**0.948**	0.0520	0	***small***	**0.9036**	0.0964	0
medium	0	0.8616	0.1384	***medium***	0	**0.8849**	0.1151
large	0	0.2293	0.7707	large	0	0.1424	0.8576

Assuming the threshold $\tau = 0.87$ (values for which the ratio exceed this threshold are marked as bold in Table 2.), we can create the following sets:

$$\Theta^{Root}_{Setosa} = \left\{a^{PL}_{small}; a^{PW}_{small}\right\}, \quad \Theta^{Root}_{Versicolour} = \left\{a^{PW}_{medium}\right\}$$

For $\sigma = 0.5$, set Ψ^N contains 102 examples. By executing steps 4 and 5, the algorithm chooses attribute *petal width* for splitting the root node. Repeating the algorithm for each new nonleaf node, we obtain the tree shown in Fig. 4.

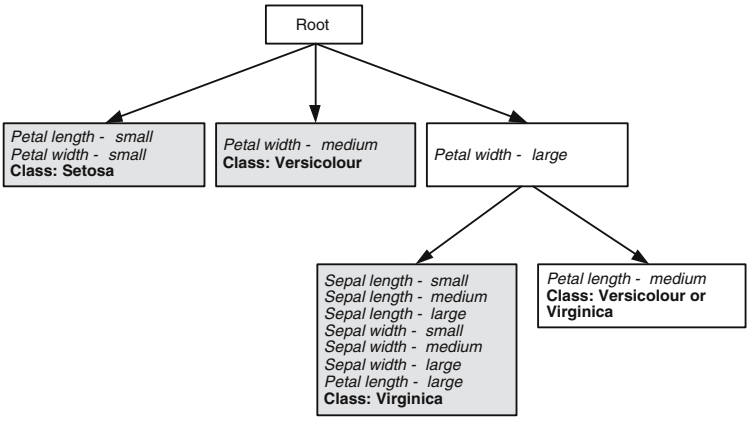

Fig. 4. Decision tree created by MAL FID3 algorithm, for Iris classification problem

It contains 6 nodes and gives 96% correct classifications. The tree created by the classical Fuzzy-ID3 algorithm contains 26 nodes and reaches 93% correct classifications.

4 Other Experimental Results

Three popular data sets, which are available at the UCI Machine Learning Repository [1], were used in the experiments. Every experiment was repeated twenty times to get the error ratio, shown in Tables 3-5. These tables present results for the classical Fuzzy-ID3 algorithm and the MAL FID3. Table 3 concerns the wine classification problem, with 13 attributes, 3 classes, 128 examples in the training set and 32 examples in the testing set. Table 4 includes results for the glass classification problem with 9 attributes, 6 classes, 174 examples in the training set and 40 examples in the testing set. Table 5 presents results for the heart desease (medical diagnosis problem), with 10 attributes, 2 classes, 221 examples in the training set and 40 examples in the testing set. The threshold value, τ, and the average number of nodes obtained from 20 experiments are included in the tables.

As can be noticed for all presented problems, the trees obtained by the MAL FID3 algorithm, proposed in this paper, are smaller (less number of nodes) than those obtained by the classical Fuzzy-ID3 algorithm. The error ratios are comparable for both kinds of the trees.

Table 3. Results for the wine classification problem

	Fuzzy-ID3		MAL FID3	
τ	Average number of nodes	Error[%]	Average number of nodes	Error[%]
0.65	13.1	22.81	9	11.40
0.7	44.85	17.65	13.4	9.68
0.75	111.45	12.50	13.8	8.75
0.8	251.15	10.78	17.2	9.37
0.85	502.55	8.59	33.65	7.03
0.9	974	8.75	80.9	5.31

Table 4. Results for the glass classification problem

	Fuzzy-ID3		MAL FID3	
τ	Average number of nodes	Error[%]	Average number of nodes	Error[%]
0.65	280.45	41	98.95	42.62
0.7	342.95	40.87	192.55	41.25
0.75	386.35	40.87	203.35	40.87
0.8	443.9	41.25	291.6	40.62
0.85	515.6	41.12	313.65	40.12
0.9	591.25	40.62	354.35	41.50

Table 5. Results for the heart classification problem

	Fuzzy-ID3		MAL FID3	
τ	Average number of nodes	Error[%]	Average number of nodes	Error[%]
0.65	4.57	25	3	17.3
0.7	64.92	23.05	3	18.75
0.75	122.85	22.67	6.78	18.75
0.8	240.14	22.67	39.28	20.70
0.85	374.78	20.52	84	21.77
0.9	477	20.70	167.28	23.92

5 Conclusions

In this paper, a new version of the Fuzzy-ID3 algorithm, called MAL FID3, is presented. The modification, introduced to the classical Fuzzy-ID3, makes possible the use of many values of different attributes in the leaves of a tree. The trees build according to the proposed algorithm are smaller (less number of nodes) than those created by the classical Fuzzy-ID3 method. For some problems these trees can produce better classification results.

The purpose of the future works is a further reduction of the size of fuzzy decision trees and elimination of those fuzzy sets from the leaves that have no influence on the classification process.

References

1. Blake C., Keogh E., Merz C.: UCI repository of machine learning databases, [http://www.ics.uci.edu/∼mlearn/MLRepository.html], Irvine, CA: University of California, Dept. of Computer Science (1998).
2. Dubois D., Prade H.: Fuzzy Sets and Systems: Theory and Applications, Academic Press, San Diego (1980).
3. Fisher R.A. The Use of Multiple Measurements in Taxonomic Problems, Annals of Eugenics (1936) Part II 179-188
4. Friedman J.H., Kohavi R., Yun Y.: Lazy Decision Trees, in Proc. of the Thirteenth National Conference on Artificial Intelligence, (1996), 717-724.
5. Ichihashi H., Shirai T., Nagasaka K., Miyoshi T., Neuro-fuzzy ID3: a method of inducing fuzzy decision trees with linear programming for maximizing entropy and an algebraic method for incremental learning, Fuzzy Sets and Systems, **81**, Issue 1, (1996), 157-167.
6. Janikow C.Z.: Fuzzy Decision Trees: Issues and Methods, IEEE Transactions on Systems, Man, and Cybernetics, **28**, Issue 3, (1998), 1-14.
7. Janikow C.Z.: Exemplar Learning in Fuzzy Decision Trees, in Proc. of IEEE International Conference on Fuzzy Systems, Piscataway, NJ, (1996), 1500-1505.
8. Quinlan J.R.: Induction of decision trees, Machine Learning, **1**, (1986), 81-106.
9. Quinlan J.R.: C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, Inc., Los Altos, California, (1993).

10. Rutkowska D.: Neuro-Fuzzy Architectures and Hybrid Learning, Physica-Verlag, Springer-Verlag Company, Heidelberg, New York, (2002)
11. Rutkowski L.: Artificial intelligence methods and techniques, PWN, in Polish, (2005)
12. Zadeh L.A.: Fuzzy Sets, Information and Control, **8**, (1965), 338-353.
13. Zimmermann H.-J.: Fuzzy Set Theory, Kluwer Academic Publishers, Boston, Dordrecht, London (1994).