17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems - KES2013

Decision Rules, Trees and Tests for Tables with Many-Valued Decisions – Comparative Study

Mohammad Azad[a,*], Beata Zielosko[a,b], Mikhail Moshkov[a], Igor Chikalov[a]

[a]*Computer, Electrical and Mathematical Sciences and Engineering Division*
King Abdullah University of Science and Technology
Thuwal 23955-6900, Saudi Arabia
[b]*Institute of Computer Science, University of Silesia*
39, Będzińska St., 41-200 Sosnowiec, Poland

Abstract

In this paper, we present three approaches for construction of decision rules for decision tables with many-valued decisions. We construct decision rules directly for rows of decision table, based on paths in decision tree, and based on attributes contained in a test (super-reduct). Experimental results for the data sets taken from UCI Machine Learning Repository, contain comparison of the maximum and the average length of rules for the mentioned approaches.

© 2013 The Authors. Published by Elsevier B.V. Open access under CC BY-NC-ND license.
Selection and peer-review under responsibility of KES International

Keywords: decision table with many-valued decisions; decision rule; decision tree; test

1. Introduction

In this paper, we study decision tables with many-valued decisions. In such tables each row is labeled with a set of decisions, and for a given row, we should find a decision from the set of decisions attached to this row.

We can meet such tables when we work with experimental or statistical data. In such data sets, we often have groups of rows with equal values of conditional attributes but, probably, different values of the decision attribute. In this case, instead of a group of rows, we can consider one row given by values of conditional attributes, and we attach to this row a set of decisions: either all decisions for rows from the group, or k the most frequent decisions for rows from the group [1].

In the rough sets theory [2, 3] a generalized decision is used often to work with decision tables which have equal rows labeled with different decisions (inconsistent decision tables). In this case, we also work with the decision table with many-valued decisions. The set of decisions attached to equal rows is called the generalized decision for each of these equal rows [4, 5]. The usual way is to find for a given row its generalized decision. However, the problem of finding an arbitrary decision or one of the most frequent decisions from the group is

---

*Corresponding author.
E-mail address: mohammad.azad@kaust.edu.sa.

interesting also. Proposed approach for construction of decision rules was considered in [6], for construction of decision trees in [7, 8], and for construction of tests (super-reducts) in [9, 10].

Decision rules, decision trees and tests (super–reducts) can be considered as a way of knowledge representation, can be used for feature selection and for construction of classifiers. Based on decision trees and based on tests we can construct decision rules.

The aim of this paper is to make comparative study of the maximum and the average length of decision rules. We consider three approaches for construction of decision rules for decision tables with many-valued decisions:

- for each row of a decision table T, a greedy algorithm constructs directly a decision rule;
- for a given decision table T, a greedy algorithm constructs a decision tree, then for each row of T we find the path in a decision tree from a root to a terminal node which accepts this row, and construct a decision rule;
- for a given decision table T, a greedy algorithm constructs a test, then for each row of T, based on attributes contained in a test we construct a decision rule.

To construct decision rules, decision trees and tests we use greedy algorithms. Theoretical results were presented in [9, 6, 1, 8]. It was shown that under the assumption $NP \nsubseteq DTIME(n^{O(\log \log n)})$ greedy algorithms are close to the best (from the point of view of precision) approximate polynomial algorithms for minimization of rule length, depth of decision tree and test cardinality.

In this paper, we study binary decision tables with many-valued decisions. However, the obtained results can be extended to the decision tables filled by numbers from the set $\{0, \ldots, k - 1\}$, where $k \geq 3$. We present experimental results for data sets from UCI Machine Learning Repository [11] that have been converted to the format of decision tables with many-valued decisions after removal of some conditional attributes.

This paper consists of seven sections. Section 3, contains main notions corresponding to decision tables with many-valued decisions. Sections 4, 5 and 6 describe greedy algorithms for construction of decision rules, decision trees and tests, respectively. Section 7 contains experimental results, and Sect. 8 – conclusions.

2. Related Work

In literature, often, problems that are connected with multi-label data are considered from the point of view of classification: multi-label learning [12], multi-instance learning [13]. There are also semi-supervised learning [14] where some examples are labeled but some are not labeled. Our problem does not match with the above learning problems, but to some extent, it matches with partial learning [15], ambiguous learning [16], and multiple label learning [17]. Additionally, these papers only focus on classification results rather than optimization of data model. We consider our approach as a unique one from the point of view of knowledge representation which is based on decision tree model.

3. Main Notions

In this section, we present definitions of notions corresponding to decision tables with many-valued decisions. Notions connected with decision rules, decision trees and tests are contained, respectively, in Sec. 4, Sect. 5, and Sect. 6.

A *(binary) decision table with many-valued* decisions is a rectangular table T filled by numbers from the set $\{0, 1\}$. Columns of this table are labeled with attributes $f_1, \ldots, f_n$. Rows of the table are pairwise different, and each row is labeled with a nonempty finite set of natural numbers (set of decisions). By $N(T)$ we denote the number of rows in the table T. Note, that each (binary) decision table with one-valued decisions can be interpreted also as a decision table with many-valued decisions. In such table, each row is labeled with a set of decisions which has one element.

We will say that T is a *degenerate* table if either T has no rows, or the intersection of sets of decisions attached to rows of T is nonempty.

A decision which belongs to the maximum number of sets of decisions attached to rows in T is called the *most common decision for T*. If we have more than one such decisions we choose the minimum one. If T is empty then 1 is the most common decision for T.

A table obtained from T by removal of some rows is called a *subtable* of T. Let $f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\}$ and $\delta_1, \ldots, \delta_m \in \{0, 1\}$. We denote by

$$T(f_{i_1}, \delta_1) \ldots (f_{i_m}, \delta_m)$$

the subtable of the table T which consists of all rows that at the intersection with columns $f_{i_1}, \ldots, f_{i_m}$ have numbers $\delta_1, \ldots, \delta_m$ respectively.

A subtable T' of T is called a *boundary* subtable if T' is not degenerate but each proper subtable of T' is degenerate. We denote by $B(T)$ the number of boundary subtables of the table T. It is clear that T is a degenerate table if and only if $B(T) = 0$.

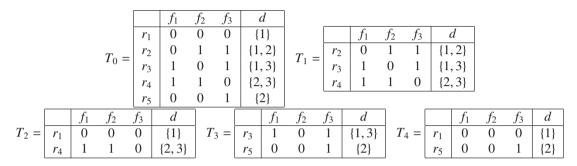All boundary subtables of a decision table T_0 with many-valued decisions can be found in Fig. 1.

Fig. 1. All boundary subtables T_1, T_2, T_3, T_4 of the decision table T_0

We will say that an attribute f_i *divides* a boundary subtable if this attribute is not constant on the rows of this subtable (for example, for a binary decision table, at the intersection with the column f_i we can find some rows which contain 1 and some rows which contain 0).

We denote by $Tab(t)$, where t is a natural number, the set of decision tables with many-valued decisions such that each row in the table has at most t decisions.

Theorem 1. *[8] Each boundary subtable of a table $T \in Tab(t)$ has at most $t + 1$ rows.*

Therefore, for tables from $Tab(t)$, there exists a polynomial algorithm for the computation of the parameter $B(T)$. For example, for any decision table T with one-valued decision the equality $B(T) = P(T)$ holds, where $P(T)$ is the number of unordered pairs of rows from T with different decisions.

4. Decision Rules

A *decision rule over T* is an expression of the kind

$$f_{i_1} = b_1 \wedge \ldots \wedge f_{i_m} = b_m \rightarrow d \tag{1}$$

where $f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\}$, $d \in \mathbb{N}$. It is possible that $m = 0$.

Let $r = (b_1, \ldots, b_n)$ be a row of T labeled with the set of decisions $D(r)$ and $d \in D(r)$. By $U(T, r, d)$ we denote the set of rows r' from T for which $d \notin D(r')$. We will say that an attribute f_i separates a row $r' \in U(T, r, d)$ from the row r if the rows r and r' have different values at the intersection with the column f_i.

A decision rule (1) is called a *decision rule for the pair (T, r) and decision $d \in D(r)$* if attributes $f_{i_1}, \ldots, f_{i_m}$ separate from r all rows r' from $U(T, r, d)$. If $m = 0$ then we require $U(T, r, d) = 0$. By $l(rule)$ we denote the *length* of the rule (1). It is the number m of descriptors (pairs *attribute = value*) on the left-hand side of the rule.

Now, we present a greedy algorithm for decision rule construction (see Algorithm 1). Let T be a decision table with many-valued decisions containing n columns labeled with attributes $f_1, \ldots, f_n$, and r be a row of T with a set of decisions $D(r)$. Greedy algorithm works for each decision $d \in D(r)$. At each iteration it chooses an attribute with the minimum index which separates from r the maximum number of unseparated rows from $U(T, r, d)$. It

---

Algorithm 1 Greedy algorithm for decision rule construction

Require: Binary decision table T with conditional attributes $f_1, \ldots, f_n$, row $r = (b_1, \ldots, b_n)$ of T labeled by the set of decisions $D(r)$, decision $d \in D(r)$.

Ensure: decision rule for (T, r) and d.

  $Q \leftarrow \emptyset$;

  while attributes from Q separate from r less than $|U(T, r, d)|$ rows from the set $U(T, r, d)$ **do**

    select $f_i \in \{f_1, \ldots, f_n\}$ with the minimum index i such that f_i separates from r the maximum number of rows from $U(T, r, d)$ not separated by attributes from Q;

    $Q \leftarrow Q \cup \{f_i\}$;

  end while

  $\bigwedge_{f_i \in Q}(f_i = b_i) \rightarrow d$.

---

stops when attributes contained in the decision rule separate from r all rows from the set $U(T, r, d)$. After that, among all decision rules constructed for a given row r and each decision $d \in D(r)$, we choose a rule with the minimum length.

We apply this algorithm sequentially to the table T and each row r of T. As a result, for each row of the decision table T, we obtain one decision rule. Such rules form a vector of rules $vec_{rule} = (rule_1, \ldots, rule_{N(T)})$. By $l_{max}(vec_{rule})$ we denote the maximum length of a rule from vec_{rule}:

$$l_{max}(vec_{rule}) = \max\{l(rule_i) : i = 1, \ldots, N(T)\}.$$

By $l_{avg}(vec_{rule})$ we denote the average length of rules from vec_{rule}:

$$l_{avg}(vec_{rule}) = \frac{\sum_{i=1}^{N(T)} l(rule_i)}{N(T)}.$$

For decision table T_0, depicted in Fig. 1, the vector of constructed decision rules is the following: $vec_{rule} = (f_1 = 0 \wedge f_3 = 0 \rightarrow 1, f_2 = 1 \rightarrow 2, f_1 = 1 \rightarrow 3, f_2 = 1 \rightarrow 2, f_1 = 0 \wedge f_3 = 1 \rightarrow 2)$, $l_{max}(vec_{rule}) = 2$, $l_{avg}(vec_{rule}) = 1.4$.

5. Decision Trees

A *decision tree over T* is a finite tree with a root in which each terminal node is labeled with a decision (a natural number), and each nonterminal node is labeled with an attribute from the set $\{f_1, \ldots, f_n\}$. Two edges start in each nonterminal node. These edges are labeled with 0 and 1 respectively.

Let Γ be a decision tree over T and v be a node of Γ. We correspond to the node v a subtable $T(v)$ of the table T. If v is the root of Γ then $T(v) = T$. Otherwise, let nodes and edges in the path from the root to v be labeled with attributes $f_{i_1}, \ldots, f_{i_m}$ and numbers $\delta_1, \ldots, \delta_m$ respectively. Then $T(v)$ is the subtable $T(f_{i_1}, \delta_1) \ldots (f_{i_m}, \delta_m)$ of the table T.

It is clear that for any row r of T there exists exactly one terminal node v in Γ such that r belongs to $T(v)$. The decision attached to v will be considered as the result of Γ work on the row r. We will say that Γ is a *decision tree for T* if for any row r of T, the decision as the result of the work of Γ on the row r, belongs to the set of decisions attached to the row r.

The *depth* of the decision tree Γ is the maximum length of a path from the root to a terminal node.

Now, we present a greedy algorithm for construction of a decision tree for a given decision table (see Algorithm 2). Let T be a binary decision table with many-valued decisions containing n columns labeled with attributes $f_1, \ldots, f_n$. During the construction of a tree Γ the greedy algorithm at each iteration chooses, for a subtable T', an attribute f_i with the minimum index i, for which the value $Q(f_i) = \max\{B(T'(f_i, 0)), B(T'(f_i, 1))\}$ is the minimum. It stops when all subtables corresponding to terminal nodes are degenerate. Fig. 2 presents a decision tree constructed by the greedy algorithm for the decision table T_0 depicted in Fig. 1.

Algorithm 2 Greedy algorithm for decision tree construction

Require: Binary decision table T with conditional attributes $f_1, \ldots, f_n$.
Ensure: Decision tree Γ for T.
  Construct a tree G consisting of a single node labeled with the table T;
  while (true) **do**
    if no one node of the tree G is labeled with a table **then**
      Denote the tree G by Γ;
    else
      Choose a node v in the tree G which is labeled with a subtable T' of the table T;
      if subtable T' is degenerate **then**
        Instead of T' mark the node v with the most common decision for T';
      else
        for $i = 1, \ldots, n$, compute the value
          $Q(f_i) = \max\{B(T'(f_i, 0)), B(T'(f_i, 1))\}$;
        Instead of T' mark the node v with the attribute f_{i_0}, where i_0 is the minimum i for which $Q(f_i)$ has the minimum value;
        For each $\delta \in \{0, 1\}$, add to the tree G the node $v(\delta)$ and mark this node with the subtable $T'(f_{i_0}, \delta)$;
        Draw the edge from v to $v(\delta)$ and mark this edge with δ;
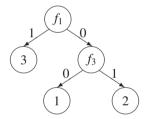   **end if**
    end if
  end while

Fig. 2. Decision tree constructed by the greedy algorithm for T_0

Let T be a decision table with rows $r_1, \ldots, r_{N(T)}$ and Γ be a decision tree for T constructed by the considered greedy algorithm. For $i = 1, \ldots, N(T)$, we correspond to the row r_i a rule $rule_i$ extracted from Γ. Let v be a terminal node of Γ such that r_i belongs to $T(v)$ and v be labeled with a decision d. If v is the root of Γ then $rule_i$ is equal to $\rightarrow d$. Let v be not the root, nodes in the path from the root to v be labeled with attributes $f_{i_1}, \ldots, f_{i_m}$, and edges in this path be labeled with numbers $\delta_1, \ldots, \delta_m$. Then $rule_i$ is equal to $f_{i_1} = \delta_1 \wedge \ldots \wedge f_{i_m} = \delta_m \rightarrow d$. One can show that $rule_i$ is a decision rule for (T, r_i) and d (it is clear that $d \in D(r_i)$).

We denote $vec_{tree} = (rule_1, \ldots, rule_{N(T)})$. By $l_{max}(vec_{tree})$ we denote the maximum length of a rule from vec_{tree}:

$$l_{max}(vec_{tree}) = \max\{l(rule_i) : i = 1, \ldots, N(T)\}.$$

This value coincides with the depth of Γ. By $l_{avg}(vec_{tree})$ we denote the average length of rules from vec_{tree}:

$$l_{avg}(vec_{tree}) = \frac{\sum_{i=1}^{N(T)} l(rule_i)}{N(T)}.$$

For the decision table T_0, depicted in Fig. 1, and decision tree depicted in Fig. 2, the vector of decision rules is the following: $vec_{tree} = (f_1 = 0 \wedge f_3 = 0 \rightarrow 1, f_1 = 0 \wedge f_3 = 1 \rightarrow 2, f_1 = 1 \rightarrow 3, f_1 = 1 \rightarrow 3, f_1 = 0 \wedge f_3 = 1 \rightarrow 2)$, $l_{max}(vec_{tree}) = 2$, $l_{avg}(vec_{tree}) = 1.6$.

6. Tests

A test for the table T is a subset of attributes $\{f_{i_1}, \ldots, f_{i_m}\}$ such that these attributes divide all boundary subtables of a decision table T.

Now, we present a greedy algorithm for test construction (see Algorithm 3). Let T be a table with many-valued decisions containing n columns labeled with attributes $f_1, \ldots, f_n$, and $B(T)$ be the number of boundary subtables of the table T. Greedy algorithm at each iteration chooses an attribute which divides the maximum number of not divided boundary subtables. This algorithm stops if attributes from the test divide $B(T)$ boundary subtables.

Algorithm 3 Greedy algorithm for test construction

Require: Binary decision table T with conditional attributes $f_1, \ldots, f_n$.
Ensure: test for T.
  $Q \leftarrow \emptyset$;
  while attributes from Q divide less than $B(T)$ boundary subtables **do**
    select $f_i \in \{f_1, \ldots, f_n\}$ with the minimum index such that f_i divides the maximum number of boundary subtables not divided by attributes from Q;
    $Q \leftarrow Q \cup \{f_i\}$;
  end while

Let us consider the decision table T_0 and its boundary subtables T_1, T_2, T_3, T_4 presented in Fig. 1. One can see that the attribute f_1 divides T_1, T_2, T_3, $f_2 - T_1, T_2$, and $f_3 - T_1, T_4$. The greedy algorithm in the first iteration chooses the attribute f_1 because it divides the maximum number of boundary subtables, in the second iteration the greedy algorithm chooses the attribute f_3. So, $\{f_1, f_3\}$ is a test for T_0 constructed by the greedy algorithm.

Let T be a decision table with n columns labeled with attributes $f_1, \ldots, f_n$, and with $N(T)$ rows $r_1, \ldots, r_{N(T)}$. Let $\{f_{i_1}, \ldots, f_{i_m}\}$ be a test for T. Now, for each $j \in \{1, \ldots, N(T)\}$, we describe a rule $rule_j$. Let $r_j = (b_1, \ldots, b_n)$. It is clear that the table $T' = T(f_{i_1}, b_{i_1}) \ldots (f_{i_m}, b_{i_m})$ is degenerate. Let d be the most common decision for T'. It is clear also that $d \in D(r_j)$. Then $rule_j$ is equal to $f_{i_1} = b_{i_1} \wedge \ldots \wedge f_{i_m} = b_{i_m} \rightarrow d$. One can show that $rule_j$ is a decision rule for (T, r_j) and d.

We denote $vec_{test} = (rule_1, \ldots, rule_{N(T)})$ and by $l_{max}(vec_{test})$ we denote the maximum length of a rule from vec_{test}. It is clear that

$$l_{max}(vec_{test}) = m.$$

By $l_{avg}(vec_{test})$ we denote the average length of rules from vec_{test}. It is clear that

$$l_{avg}(vec_{test}) = \frac{\sum_{i=1}^{N(T)} l(rule_i)}{N(T)} = m.$$

For decision table T_0 depicted in Fig. 1 and test $\{f_1, f_3\}$, the vector of decision rules is the following: $vec_{test} = (f_1 = 0 \wedge f_3 = 0 \rightarrow 1, f_1 = 0 \wedge f_3 = 1 \rightarrow 2, f_1 = 1 \wedge f_3 = 1 \rightarrow 1, f_1 = 1 \wedge f_3 = 0 \rightarrow 2, f_1 = 0 \wedge f_3 = 1 \rightarrow 2)$, $l_{max}(vec_{test}) = 2$, $l_{avg}(vec_{test}) = 2$.

7. Experimental Results

In this section, we present experimental results. First, we show how we constructed decision tables with many–valued decisions based on data sets from UCI Machine Learning Repository [11]. We consider a number of decision tables from UCI Machine Learning Repository. In some tables there were missing values. Each such value was replaced with the most common value of the corresponding attribute. Some decision tables contain conditional attributes that take unique value for each row. Such attributes were removed. We removed from these tables some conditional attributes. As a result we obtained inconsistent decision tables contained equal rows with different decisions. Each group of identical rows was replaced with a single row from the group which is labeled with the set of decisions attached to rows from the group.

Table 1. Characteristics of decision tables with many-valued decisions

Decision Table	Rows	Attr	Spectrum						Removed Attributes
			#1	#2	#3	#4	#5	#6	
balance-scale-1	125	3	45	50	30				left-weight
breast-cancer-1	193	8	169	24	0				tumor-size
breast-cancer-5	98	4	58	40					inv-nodes,node-caps,deg-malig, breast-quad,irradiat
cars-1	432	5	258	161	13				buying
flags-5	171	21	159	12					zone,language,religion,circles,sunstars
hayes-roth-data-1	39	3	22	13	4				marital_status
kr-vs-kp-5	1987	31	1564	423					katri,mulch,rimmx,skrxp,wknck
kr-vs-kp-4	2061	32	1652	409					katri,mulch,rimmx,wknck
lymphography-5	122	13	113	9					lymphatics,changes_in_node,changes_in_stru, special_forms,no_of_nodes_in
mushroom-5	4078	17	4048	30					odor,gill-size,stalk-root, stalk-surface-below-ring,habitat
nursery-4	240	4	97	96	47				parents,housing,finance,social
nursery-1	4320	7	2858	1460	2				parents
spect-test-1	164	21	161	3					F3
teeth-1	22	7	12	10					top incisors
teeth-5	14	3	6	3	0	5	0	2	bottom incisors,top canines,bottom canines, top premolars,bottom molars
tic-tac-toe-4	231	5	102	129					top-right-square,middle-middle-square, bottom-left-square,bottom-right-square
tic-tac-toe-3	449	6	300	149					middle-middle-square,bottom-left-square, bottom-right-square
zoo-data-5	42	11	36	6					feathers,backbone,breathes,legs,tail

The information about obtained decision tables with many-valued decisions can be found in Table 1. This table contains the name of initial table from [11] with an index equal to the number of removed conditional attributes, number of rows (column "Rows"), number of attributes (column "Attr"), spectrum of this table (column "Spectrum"), and names of removed attributes (column "Removed Attributes"). Spectrum of a decision table with many-valued decisions is a sequence #1, #2,..., where #i, $i = 1, 2, ...$, is the number of rows labeled with sets of decision with the cardinality equal to i.

Table T_{max} depicted in Fig. 3 presents, for a given decision table, the maximum length of decision rules constructed by Algorithm 1 (column "Rules"), the maximum length of decision rules extracted from the decision tree constructed by Algorithm 2 (column "Trees"), and the maximum length of decision rules extracted from the test constructed by Algorithm 3 (column "Tests").

Table T_{avg} depicted in Fig. 3 presents, for a given decision table, the average length of decision rules constructed by Algorithm 1 (column "Rules"), the average length of decision rules extracted from the decision tree constructed by Algorithm 2 (column "Trees"), and the average length of decision rules extracted from the test constructed by Algorithm 3 (column "Tests").

Presented results show that the maximum length of rules constructed directly for rows of T is often smaller than the maximum length of rules extracted from the decision tree and extracted from the test. Only for data set "zoo-data-5", the maximum length of decision rules constructed directly for rows of T is greater than the maximum length of rules extracted from the decision tree. For six data sets ("balance-scale-1", "hayes-roth-data-1", "nursery-4", "teeth-5", "tic-tac-toe-3") the values of the maximum length of rules are the same for each approach. For the average length of rules, presented in T_{avg}, the differences are more noticeable. For data sets "kr-vs-kp-5", "kr-vs-kp-4", and "spect-test-1", the average length of rules extracted from the test is more than six times greater than the average length of rules constructed directly for rows of T.

$T_{max} =$	Decision Table	Rules	Trees	Tests
	balance-scale-1	2	2	2
	breast-cancer-3	5	6	8
	breast-cancer-5	3	3	4
	cars-1	4	5	5
	flags-5	5	6	13
	hayes-roth-data-1	2	2	2
	kr-vs-kp-5	11	13	26
	kr-vs-kp-4	11	12	27
	lymphography-5	5	6	11
	mushroom-5	5	7	8
	nursery-4	2	2	2
	nursery-1	5	7	7
	spect-test-1	5	7	10
	teeth-1	3	4	5
	teeth-5	3	3	3
	tic-tac-toe-4	4	5	5
	tic-tac-toe-3	6	6	6
	zoo-data-5	5	4	9

$T_{avg} =$	Decision Table	Rules	Trees	Tests
	balance-scale-1	2.0	2.0	2.0
	breast-cancer-3	2.9	3.7	8.0
	breast-cancer-5	1.7	1.8	4.0
	cars-1	1.4	2.9	5.0
	flags-5	2.4	3.8	13.0
	hayes-roth-data-1	1.6	1.7	2.0
	kr-vs-kp-5	4.1	8.2	26.0
	kr-vs-kp-4	4.1	8.1	27.0
	lymphography-5	2.7	3.8	11.0
	mushroom-5	1.5	2.8	8.0
	nursery-4	1.3	1.3	2.0
	nursery-1	2.1	2.8	7.0
	spect-test-1	1.3	3.3	10.0
	teeth-1	2.3	2.8	5.0
	teeth-5	1.9	2.2	3.0
	tic-tac-toe-4	2.2	3.0	5.0
	tic-tac-toe-3	3.3	4.3	6.0
	zoo-data-5	2.2	3.2	9.0

Fig. 3. Table T_{max} presents the maximum length of decision rules. Table T_{avg} presents the average length of decision rules

8. Conclusions

In the paper, we presented three approaches for construction of decision rules for decision tables with many-valued decisions. They are based on greedy algorithms for construction of decision rules, decision trees and tests. The values of the maximum length of rules constructed based on the decision tree and based on attributes contained in the test are often greater than the values of the maximum length of rules constructed directly for rows of the decision table. In the case of the average length of rules, the differences are more noticeable.

References

[1] M. Moshkov, B. Zielosko, Combinatorial Machine Learning – A Rough Set Approach, Springer, Heidelberg, 2011.
[2] Z. Pawlak, Rough Sets – Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Dordrecht, 1991.
[3] A. Skowron, C. Rauszer, The discernibility matrices and functions in information systems, in: Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory, Kluwer Academic Publishers, Dordrecht, 1992, pp. 331–362.
[4] Z. Pawlak, A. Skowron, Rough sets and boolean reasoning, Inf. Sci. 177 (1) (2007) 41–73.
[5] Z. Pawlak, A. Skowron, Rudiments of rough sets, Inf. Sci. 177 (1) (2007) 3–27.
[6] I. Chikalov, B. Zielosko, Decision rules for decision tables with many-valued decisions, in: J. Yao, S. Ramanna, G. Wang, Z. Suraj (Eds.), RSKT 2011, Vol. 6954 of LNCS, Springer, 2011, pp. 763–768.
[7] M. Azad, I. Chikalov, M. Moshkov, B. Zielosko, Greedy algorithm for construction of decision trees for tables with many-valued decisions, in: 21th International Workshop Concurrency, Specification and Programming CS&P 2012, September 26–28, Berlin, Germany, 2012, pp. 13–24.
[8] M. Moshkov, B. Zielosko, Construction of α-decision trees for tables with many-valued decisions, in: J. Yao, S. Ramanna, G. Wang, Z. Suraj (Eds.), RSKT 2011, Vol. 6954 of LNCS, Springer, 2011, pp. 486–494.
[9] M. Azad, I. Chikalov, M. Moshkov, B. Zielosko, Greedy algorithms for construction of approximate tests for decision tables with many-valued decisions, Fundam. Inform. 120 (3-4) (2012) 231–242.
[10] M. Moshkov, B. Zielosko, Construction of tests for tables with many-valued decisions, in: 20th International Workshop Concurrency, Specification and Programming CS&P 2011, September 28–30, Pułtusk, Poland, Białystok University of Technology, 2011, pp. 376–384.
[11] A. Asuncion, D. J. Newman, UCI Machine Learning Repository (2007).
URL http://www.ics.uci.edu/~mlearn/
[12] G. Tsoumakas, I. Katakis, Multi-label classification: An overview, IJDWM 3 (3) (2007) 1–13.
[13] Z.-H. Zhou, M.-L. Zhang, S.-J. Huang, Y.-F. Li, Multi-instance multi-label learning, Artif. Intell. 176 (1) (2012) 2291–2320.
[14] X. Zhu, A. B. Goldberg, Introduction to Semi-Supervised Learning, Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, 2009.
[15] T. Cour, B. Sapp, C. Jordan, B. Taskar, Learning from ambiguously labeled images, in: CVPR, 2009, pp. 919–926.
[16] E. Hllermeier, J. Beringer, Learning from ambiguously labeled examples, in: Advances in Intelligent Data Analysis VI, Vol. 3646 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2005, pp. 168–179.
[17] R. Jin, Z. Ghahramani, Learning with multiple labels, in: S. Becker, S. Thrun, K. Obermayer (Eds.), NIPS, 2002, pp. 897–904.