STUDIA INFORMATICA 2013
Volume 34 Number 2A (111)

Beata ZIELOSKO
KAUST, CEMSE Division
University of Silesia, Institute of Computer Science

COVERAGE OF EXACT DECISION RULES

Summary. In the paper, author proposes a heuristics based on dynamic programming algorithm for optimization of exact decision rules relative to coverage. There are two aims for the proposed algorithm: (i) study of coverage of rules and comparison with coverage of rules constructed by the dynamic programming algorithm, (ii) study of size of directed acyclic graph (the number of nodes and edges) and comparison with size of the graph constructed by the dynamic programming algorithm.

Keywords: decision rules, coverage, dynamic programming algorithm

POKRYCIE DOKŁADNYCH REGUŁ DECYZYJNYCH

Streszczenie. W artykule zaproponowano heurystykę na podstawie algorytmu dynamicznego programowania dla optymalizacji dokładnych reguł decyzyjnych odnośnie do pokrycia. Celem przeprowadzonych badań jest: (i) zbadanie pokrycia reguł konstruowanych za pomocą proponowanego algorytmu oraz porównanie z pokryciem reguł konstruowanych za pomocą algorytmu dynamicznego programowania, (ii) zbadanie rozmiaru grafu (liczba węzłów i krawędzi w skierowanym grafie acyklicznym) skonstruowanego za pomocą proponowanego algorytmu oraz porównanie go z rozmiarem grafu skonstruowanego za pomocą algorytmu dynamicznego programowania.

Słowa kluczowe: reguły decyzyjne, pokrycie, algorytm dynamicznego programowania

1. Introduction

Decision rules are used in many areas connected with data mining and knowledge representation. There are different approaches for construction of decision rules, for example, Boolean reasoning [8, 9, 11], different kinds of greedy algorithms [8, 12], separate and conquer approach [6, 7], dynamic programming approach [2, 3]. Also, there are different rule quality measures that are used for induction or classification processes [4, 10].

In the paper, author presents a heuristics that is based on the dynamic programming algorithm for exact decision rules optimization relative to coverage [3]. The rule coverage is a measure that allows to discover major patterns in the data.

For a given decision table T the directed acyclic graph $\Delta(T)$ is constructed. Nodes of this graph are subtables described by descriptors (pairs attribute=value). In comparison with classical algorithm presented in [3], subtables of the graph $\Delta(T)$ are constructed only for the most frequent value of each attribute (value of an attribute attached to the maximum number of rows). So, the size of the graph $\Delta(T)$ is lesser. This fact is important from the point of view of number of rows and attributes in a decision table, and from the point of view of scalability. Based on the graph $\Delta(T)$ we can describe sets of decision rules for rows of table T. Then, based on a procedure of optimization relative to coverage we can find decision rules for table T and row r with the maximum coverage.

There are two aims for a proposed heuristics: (i) study of a coverage of rules and comparison with a coverage of exact decision rules constructed by the dynamic programming algorithm, (ii) study of a size of directed acyclic graph (the number of nodes and edges) and comparison with a size of the directed acyclic graph constructed by the dynamic programming algorithm.

The paper contains also experimental results with decision tables from UCI Machine Learning Repository [5].

The paper consists of six sections. In Section 2, main notions are presented. In Section 3, proposed algorithm for construction of directed acyclic graph is considered. Section 4 contains a description of a procedure of optimization relative to coverage. Section 5 contains experimental results, Section 6 – conclusions.

2. Main Notions

A *decision table T* is a rectangular table with n columns labeled with conditional attributes $f_1, \ldots, f_n$. Rows of this table are filled with nonnegative integers that are interpreted as values of conditional attributes. Rows of T are pairwise different and each row is labeled with a nonnegative integer that is interpreted as a value of the decision attribute.

We denote by $N(T)$ the number of rows in table T. The table T is called *degenerate* if T is empty (in this case $N(T) = 0$) or all rows of T are labeled with the same decision.

A minimum decision value that is attached to the maximum number of rows is called *the most common decision*. We will say that an attribute $f_{i,\, i \in 1,\ldots,n}$ is *not constant on T* if it has at least two different values. For each attribute that is not constant on T we denote *the most frequent value*. It is an attribute value attached to the maximum number of rows. If there are two or more such values then we choose the most frequent value of an attribute for which exists the most common decision.

A table obtained from T by the removal of some rows is called a *subtable* of the table T. Let T be nonempty, $f_{i1},...,f_{im} \in \{f_1,...,f_n\}$ and $a_1,...,a_m$ be nonnegative integers. We denote by $T(f_{i1},a_1)...(f_{im},a_m)$ the subtable of the table T that contains only rows that have numbers $a_1,...,a_m$ at the intersection with columns $f_{i1},...,f_{im}$. Such nonempty subtables (including the table T) are called *separable subtables* of T.

We denote by $E(T)$ a set of attributes from $\{f_1,...,f_n\}$ with the most frequent value. For any $f_i \in E(T)$, we denote by $E(T,f_i)$ the most frequent value of the attribute f_i in T.

The expression

$$f_{i1} = a_1 \wedge \ldots \wedge f_{im} = a_m \rightarrow d \qquad (1)$$

is called a *decision rule over T* if $f_{i1},\ldots,f_{im} \in \{f_1,\ldots,f_n\}$, and $a_1,\ldots,a_m$, d are nonnegative integers. It's possible that $m = 0$. In this case (1) is equal to the rule

$$\rightarrow d. \qquad (2)$$

Let $r = (b_1,\ldots,b_n)$ be a row of T. We will say that the rule (1) is *realizable for r*, if $a_1 = b_{i1},\ldots,a_m = b_{im}$. If $m = 0$ then the rule (2) is realizable for any row from T. We will say that the rule (1) is *true for T* if each row of T for which the rule (1) is realizable has the decision d attached to it. Note that (1) is true for T if and only if the table $T' = T(f_{i1},a_1)\ldots(f_{im},a_m)$ is degenerate and each row of T' is labeled with the decision d. If $m = 0$ then the rule (2) is true for T if and only if T is degenerate and each row of T is labeled with the decision d. If the rule (1) is true for T and realizable for r, we will say that (1) is a *decision*

rule for T and r.

Let τ be a decision rule over T and τ be equal to (1). The *coverage* of τ is the number of rows in T for which τ is realizable and which are labeled with the decision d. We denote it by $c(\tau)$. The coverage of decision rule (2) is equal to the number of rows in T that are labeled with the decision d. If τ is true for T then $c(\tau) = N(T(f_{i1},a_1)\ldots(f_{im},a_m))$.

3. Directed Acyclic Graph Δ(T)

In this section, we present an algorithm that construct, for a given table T, a directed acycylic graph. Based on this graph we can describe sets of decision rules attached to rows of T. Nodes of the graph are separable subtables of the table T. During each step, the algorithm processes one node and marks it with the symbol *. At the first step, the algorithm constructs a graph containing a single node T that is not marked with *.

Let us assume that the algorithm has already performed p steps. We describe now the step (p+1). If all nodes are marked with the symbol * as processed, the algorithm finishes its work and presents the resulting graph as $\Delta(T)$. Otherwise, choose a node (table) θ, that has not been processed yet. If θ is degenerate, then mark θ with the symbol * and go to the step ($p+2$). Otherwise, for each $f_i \in E(\theta)$, draw an edge from the node θ and label it with pair (f_i,b_i) (b_i is the most frequent value of the attribute f_i). This edge enters to node $\theta(f_i,b_i)$. If such node $\theta(f_i,b_i)$ is absent in the graph then add this node to the graph. We label each row r of θ with the set of attributes $E_{\Delta(T)}(\theta,r) \subseteq E(\theta)$. It is possible that $E_{\Delta(T)}(\theta,r)$ will be empty. Mark the node θ with the symbol * and proceed to the step ($p+2$). The graph $\Delta(T)$ is a directed acyclic graph. A node of this graph will be called terminal if there are no edges leaving this node. Note that a node θ of $\Delta(T)$ is terminal if and only if θ is degenerate.

In the next section, we will describe procedure of optimization of the graph $\Delta(T)$ relative to the coverage. As a result we will obtain a graph G with the same sets of nodes and edges as in $\Delta(T)$. The only difference is that any row r of each nondegenerate table θ from G is labeled with a set of attributes $E_G(\theta,r) \subseteq E_{\Delta(T)}(\theta,r)$, that allows to describe, for a row r, set of decision rules with the maximum coverage. It is possible also that $G = \Delta(T)$.

Now, for each node θ of G and for each row r of θ we describe a set of decision rules $Rul_G(\theta,r)$. If the set $E_G(\theta,r)$ is empty then the set $Rul_G(\theta,r)$ will be empty also. We will move from terminal nodes of G to the node T.

Let θ be a terminal node of G: θ is a degenerate table, so each row is labeled with the same decision d. Then

$$Rul_G(\theta,r) = \{\rightarrow d\}.$$

Let now θ be a nonterminal node of G such that for each child θ′ of θ and for each row r' of θ′ the set of rules $Rul(\theta',r')$ is already defined. Let $r = (b_1,\ldots,b_n)$ be a row of θ labeled with a decision d. For any $f_i \in E_G(\theta,r)$, we define the set of rules $Rul_G(\theta,r,f_i)$ as follows:

$$Rul_G(\theta,r,f_i) = \{f_i = b_i \wedge \alpha \rightarrow d : \alpha \rightarrow d \in Rul_G(\theta(f_i,b_i),r)\}.$$

Then

$$Rul_G(\theta,r) = \cup_{f_i \in E_G(\theta,r)} Rul_G(\theta,r,f_i).$$

To illustrate the presented algorithm we consider a simple decision table T_0 depicted on the top of Fig. 1. We denote the obtained graph $\Delta(T_0)$ by G. Now, for each node θ of the graph G and for each row r of θ we describe the set $Rul_G(\theta,r)$. We will move from terminal nodes of G to the node T_0. Terminal nodes of the graph G are θ1, θ2, θ4, θ6. For these nodes,

$$Rul_G(\theta 1,r1) = Rul_G(\theta 1,r2) = Rul_G(\theta 1,r4) = \{\rightarrow 1\};$$

$Rul_G(\theta 2,r1) = Rul_G(\theta 2,r4) = \{\rightarrow 1\}$;

$Rul_G(\theta 4,r2) = Rul_G(\theta 4,r4) = \{\rightarrow 1\}$;

$Rul_G(\theta 6,r2) = \{\rightarrow 1\}$;

Now, we can describe the sets of rules attached to rows of nonterminal node θ5. For this subtable the child (subtable θ6) is already treated, and we have:

$Rul_G(\theta 5,r2) = \{f_1 = 0 \rightarrow 1\}$; $Rul_G(\theta 5,r3) = \{\ \}$.

Now, for θ*3* all children θ*4* and θ*5* are already treated, and we have:

$Rul_G(\theta 3,r2) = \{f_1 = 1 \rightarrow 1,\ f_2 = 0 \wedge\ f_1 = 1 \rightarrow 1\}$;

$Rul_G(\theta 3,r3) = \{\ \}$;

$Rul_G(\theta 3,r4) = \{f_1 = 1 \rightarrow 1\}$;

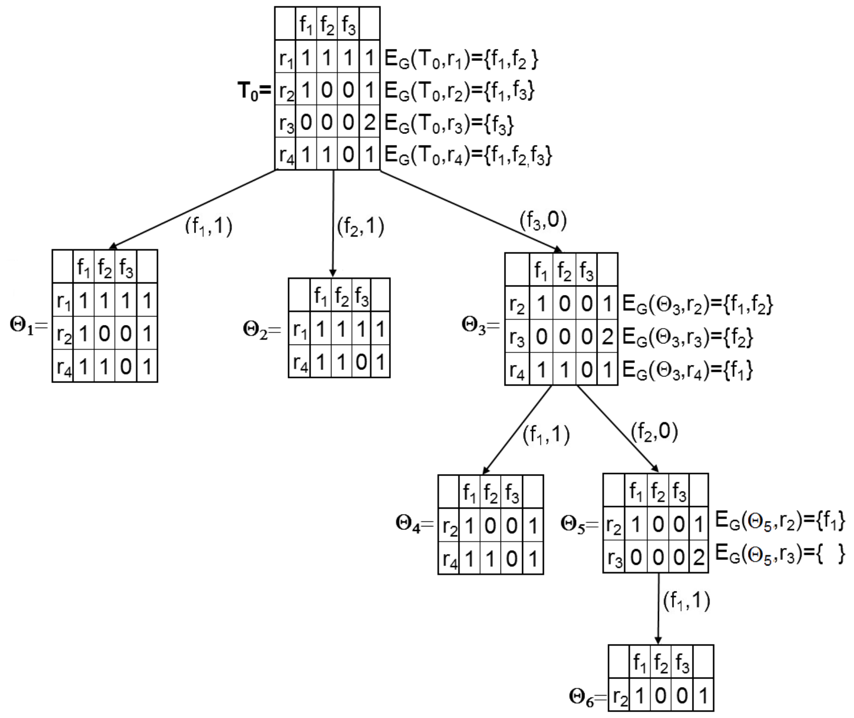
Finally, we can describe the sets of rules attached to rows of *T₀*:

$Rul_G(T_0,r1) = \{f_1 = 1 \rightarrow 1,\ f_2 = 1 \rightarrow 1\}$;

$Rul_G(T_0,r2) = \{f_1 = 1 \rightarrow 1,\ f_3 = 0 \wedge\ f_1 = 1 \rightarrow 1,\ f_3 = 0 \wedge\ f_2 = 0 \wedge\ f_1 = 1 \rightarrow 1\}$;

$Rul_G(\theta 3,r3) = \{\ \}$;

$Rul_G(T_0,r4) = \{f_1 = 1 \rightarrow 1,\ f_2 = 1 \rightarrow 1,\ f_3 = 0 \wedge\ f_1 = 1 \rightarrow 1\}$;



T₀=
f₁ f₂ f₃
r₁ 1 1 1 1 E_G(T₀,r₁)={f₁,f₂}
r₂ 1 0 0 1 E_G(T₀,r₂)={f₁,f₃}
r₃ 0 0 0 2 E_G(T₀,r₃)={f₃}
r₄ 1 1 0 1 E_G(T₀,r₄)={f₁,f₂,f₃}
(f₁,1)
(f₂,1)
(f₃,0)
Θ₁=
r₁ 1 1 1 1
r₂ 1 0 0 1
r₄ 1 1 0 1
Θ₂=
r₁ 1 1 1 1
r₄ 1 1 0 1
Θ₃=
r₂ 1 0 0 1 E_G(Θ₃,r₂)={f₁,f₂}
r₃ 0 0 0 2 E_G(Θ₃,r₃)={f₂}
r₄ 1 1 0 1 E_G(Θ₃,r₄)={f₁}
(f₁,1)
(f₂,0)
Θ₄=
r₂ 1 0 0 1
r₄ 1 1 0 1
Θ₅=
r₂ 1 0 0 1 E_G(Θ₅,r₂)={f₁}
r₃ 0 0 0 2 E_G(Θ₅,r₃)={ }
(f₁,1)
Θ₆=
r₂ 1 0 0 1

Fig.1. Directed acyclic graph for decision table T₀

Rys. 1. Acykliczny skierowany graf dla tabeli decyzyjnej T₀

4. Procedure of Optimization Relative to Coverage

In this section, we present a procedure of optimization of the graph G relative to the coverage c. For each node θ in the graph G, this procedure describe for rows of θ the set $Rul^c_G(\theta,r)$ of decision rules with the maximum coverage from $Rul_G(\theta,r)$ and the number $Opt^c_G(\theta,r)$ – the maximum coverage of a decision rule from $Rul_G(\theta,r)$. Note, that if $Rul_G(\theta,r) = \varnothing$ then $Rul^c_G(\theta,r) = \varnothing$ and $Opt^c_G(\theta,r) = 0$.

We will move from the terminal nodes of the graph G that are degenerate tables to the node T. We will assign to each row r of each table θ the number $Opt^c_G(\theta,r)$ and we will change the set $E_G(\theta,r)$ attached to the row r in the nonterminal table θ. We denote the obtained graph by G^c.

Let θ be a terminal node of G. Then we assign the number $Opt^c_G(\theta,r) = N(\theta)$ to each row r of θ. Let θ be a nonterminal node and all children of θ have already been treated. Let $r = (b_1,\ldots,b_n)$ be a row of θ. We assign the number

$$Opt^c_G(\theta,r) = \max\{Opt^c_G(\theta(f_i,b_i),r) : f_i \in E_G(\theta,r)\}$$

to the row r in the table θ and we set

$$E_G^c(\theta,r) = \{f_i : f_i \in E_G(\theta,r),\ Opt^c_G(\theta(f_i,b_i),r) = Opt^c_G(\theta,r)\}.$$

Figure 2 presents the directed acyclic graph G^c obtained from the graph G (see Fig. 1) by the procedure of optimization relative to the coverage.

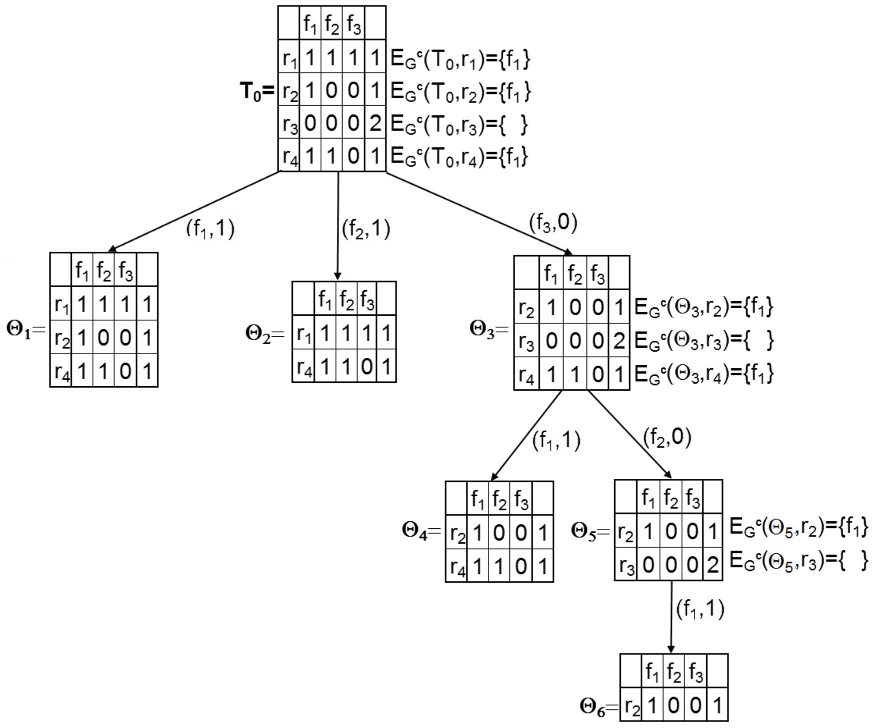

T₀=
f1 f2 f3
r1 1 1 1 1 E_G^c(T₀,r₁)={f₁}
r2 1 0 0 1 E_G^c(T₀,r₂)={f₁}
r3 0 0 0 2 E_G^c(T₀,r₃)={ }
r4 1 1 0 1 E_G^c(T₀,r₄)={f₁}
(f1,1)
(f2,1)
(f3,0)
Θ₁=
r1 1 1 1 1
r2 1 0 0 1
r4 1 1 0 1
Θ₂=
r1 1 1 1 1
r4 1 1 0 1
Θ₃=
r2 1 0 0 1 E_G^c(Θ₃,r₂)={f₁}
r3 0 0 0 2 E_G^c(Θ₃,r₃)={ }
r4 1 1 0 1 E_G^c(Θ₃,r₄)={f₁}
(f1,1)
(f2,0)
Θ₄=
r2 1 0 0 1
r4 1 1 0 1
Θ₅=
r2 1 0 0 1 E_G^c(Θ₅,r₂)={f₁}
r3 0 0 0 2 E_G^c(Θ₅,r₃)={ }
(f1,1)
Θ₆=
r2 1 0 0 1

Fig. 2. Graph G^c
Rys. 2. Graf G^c

Using the graph G^c we can describe for each row r_i, $i = 1,\dots,4$, of the table T_0 the set $Rul^c{}_G(T_0,r_i)$ of decision rules for T_0 and r_i with the maximum coverage. We will give also the value $Opt^c{}_G(T_0,r_i)$ which is equal to the maximum coverage of decision rule for T_0 and r_i. This value was obtained during the procedure of optimization of the graph G relative to the coverage. We have

$Rul^c{}_G\,(T_0,r1) = \{f_1 = 1 \rightarrow 1\}$, $Opt^c{}_G(T_0,r1) = 3$;

$Rul^c{}_G\,(T_0,r2) = \{f_1 = 1 \rightarrow 1\}$, $Opt^c{}_G(T_0,r2) = 3$;

$Rul^c{}_G\,(\theta 3,r3) = \{\ \}$, $Opt^c{}_G(T_0,r3) = 0$;

$Rul^c{}_G\,(T_0,r4) = \{f_1 = 1 \rightarrow 1\}$, $Opt^c{}_G(T_0,r4) = 0$;

5. Experimental Results

We considered a number of decision tables from UCI Machine Learning Repository [5]. Some decision tables contain conditional attributes that take unique value for each row. Such attributes were removed. In some tables there were equal rows with, possibly, different decisions. In this case each group of identical rows was replaced with a single row from the group with the most common decision for this group. In some tables there were missing values. Each such value was replaced with the most common value of the corresponding attribute. Experiments were done in a system Dagger [1] created in KAUST.

Table 1

Minimum, average and maximum coverage of decision rules

Decision table	attr	rows	min	avg	max	without rules
adult-stretch	4	16	0	6.00	8	25%
balance-scale	4	625	0	0.51	5	90%
breast-cancer	9	266	0	4.22	15	33%
cars	6	1728	0	323.00	576	42%
lymphography	18	148	1	20.52	32	0%
monks-1-test	6	432	0	31.00	108	67%
monks-1-train	6	124	0	5.77	17	30%
soybean-small	35	47	10	12.53	17	0%
tic-tac-toe	9	958	5	57.61	90	0%
Zoo	16	59	3	11.07	19	0%

For each of the considered decision tables T and for each row r of the table T, we find the maximum coverage of a decision rule for T and r. If in T exist rows without rules than the maximum coverage of a decision rule for T and r is equal to 0. After that, we find for rows of T the minimum coverage of a decision rule with maximum coverage (column *min*), the maximum coverage of such a rule (column *max*), and the average coverage of rules with maximum coverage (one for each row) (column *avg*). The considered results can be found in Table 1. Column *attr* contains number of conditional attributes, column *rows* – number of rows. The last column denotes a percent of number of rows without rules.

We repeated experiments connected with the coverage for irredundant decision rules [1], in the same way as presented for Table 1, of course in this case we don't have rows without rules. Table 2 contains the minimum, average and maximum coverage of decision rules constructed by the dynamic programming algorithm.

Table 2
Minimum, average and maximum coverage of irredundant decision rules

Decision table	attr	rows	min	avg	max
adult-stretch	4	16	4	7	8
balance-scale	4	625	1	4.21	5
breast-cancer	9	266	1	9.53	25
cars	6	1728	1	332.76	576
lymphography	18	148	2	21.54	32
monks-1-test	6	432	12	45.00	108
monks-1-train	6	124	1	13.45	29
soybean-small	35	47	10	12.53	17
tic-tac-toe	9	958	6	66.68	90
zoo	16	59	3	11.07	19

Table 3 contains comparison of the minimum, average and maximum coverage of rules constructed by proposed algorithm and dynamic programming algorithm. Each cell of this table contains a relative difference that is equal to

(*Optimum_Coverage* - *Coverage*)/*Optimum_Coverage*,

where *Coverage* denotes the coverage of decision rules constructed by proposed algorithm, *Optimum_Coverage* denotes the coverage of irredundant decision rules constructed by the dynamic programming algorithm.

Based on presented results we can see that the biggest relative difference exists for the minimum values of the maximum coverage. It follows from the fact, that for the proposed algorithm there exist rows without decision rules. Only for decision tables *soybean-small* and *zoo* there are equal values of the minimum, average and maximum coverage. If we consider values of the maximum coverage, the relative difference exists only for two decision tables: *breast-cancer* and *monks-1-train*, for the rest of decision tables the values are equal. For the average values of the maximum coverage the smallest relative difference exists for *cars* – 3% and *lymphography* – 5%. The biggest relative difference exists for decision table *balance-scale* – 88%.

Table 3

Comparison of coverage of decision rules

Decision table	attr	rows	min	avg	max
adult-stretch	4	16	1	0.14	0
balance-scale	4	625	1	0.88	0
breast-cancer	9	266	1	0.56	0.40
cars	6	1728	1	0.03	0
lymphography	18	148	0.5	0.05	0
monks-1-test	6	432	1	0.31	0
monks-1-train	6	124	1	0.57	0.41
soybean-small	35	47	**0**	**0**	**0**
tic-tac-toe	9	958	0.17	0.14	0
zoo	16	59	**0**	**0**	**0**

Table 4 presents size of the directed acyclic graph, i.e., number of nodes and number of edges for graph constructed by the proposed algorithm (column *proposed algorithm*) and graph constructed by the dynamic programming algorithm (column *dynamic programming*).

Table 4

Size of the directed acyclic graph

	proposed algorithm		dynamic programmic	
Decision table	nodes	edges	nodes	edges
adult-stretch	12	12	72	108
balance-scale	31	36	1212	3420
breast-cancer	1362	4325	6001	60387
cars	40	52	7007	19886
lymphography	25078	177090	40928	814815
monks-1-test	56	92	2772	7878
monks-1-train	200	300	1168	4592
soybean-small	3012	35825	3592	103520
tic-tac-toe	7626	14493	42532	294771
zoo	4067	34010	4568	83043

Table 5 contains comparison of the size of the directed acycylic graph for the proposed algorithm and dynamic programming algorithm. Each cell of this table is equal to number of nodes/edges for the dynamic programming algorithm divided by the number of nodes/edges for proposed algorithm.

Presented results show that for each decision table size of the directed acyclic graph for proposed algorithm is lesser. The biggest difference exists for decision tables: cars, balance-scale and monks-1-test.

Table 5
Comparison of size of the directed acyclic graph

Decision table	nodes	edges
adult-stretch	6.00	9.00
balance-scale	39.10	95.00
breast-cancer	4.41	13.96
cars	**175.18**	**382.42**
lymphography	1.63	4.60
monks-1-test	49.50	85.63
monks-1-train	5.84	15.31
soybean-small	1.19	2.89
tic-tac-toe	5.58	20.34
zoo	1.12	2.44

6. Conclusions

In the paper, a heuristic based on dynamic programming algorithm for optimization of exact decision rules relative to coverage was presented. Based on experimental results we can observe differences in size of the directed acycylic graph – that is important from the point of view of scalability, size of decision tables. There are only two decision tables for which exist relative differences connected with the values of the maximum coverage. However disadvantage for the proposed algorithm is that for a given decision table we can find rows without decision rules. So in the further work we need to improve proposed algorithm, for example, connect approach based on the most frequent value for a given attribute with a greedy heuristics or attribute with the minimum number of values.

BIBLIOGRAPHY

1. Alkhalid A., Amin T., Chikalov I., Hussain S., Moshkov M., Zielosko B.: Dagger: A tool for analysis and optimization of decision trees and rulet, [in:] Ficarra F. V. C. (ed.): Computational Informatics, Social Factors and New Information Technologies: Hypermedia Perspectives and Avant-Garde Experiences in the Era of Communicability Expansion. Blue Herons, Bergamo, Italy 2011, p. 29÷39.
2. Alsolami F., Chikalov I., Moshkov M., Zielosko B.: Optimization of inhibitory decision rules relative to length. Studia Informatica, Vol. 33, No. 2A (105), Gliwice 2012, p. 395÷406.

3. Amin T., Chikalov I., Moshkov M., Zielosko B.: Dynamic programming approach for exact decision rule optimization, [in:] Skowron A., Suraj Z. (eds.): Rough Sets and Intelligent Systems – Professor Zdzisław Pawlak in Memoriam. Intelligent Systems Reference Library, Vol. 42, Springer, 2013, p. 211÷228.
4. Ann A., Cercone N.: Rule quality measures improve the accuracy of rule induction: an experimental approach, [in:] Raś Z. W., Ohsuga S. (eds.): ISMIS 2000. LNCS, Vol. 1932, Springer, 2000, p. 119÷129.
5. Asuncion A., Newman D. J.: UCI Machine Learning Repository. 2007, http://www.ics.uci.edu/mlearn/.
6. Błaszczyński J., Słowiński R., Szeląg M.: Sequential covering rule induction algorithm for variable consistency rough set approaches. Information Science, Vol. 181, 2011, p. 987÷1002.
7. Dembczyński K., Kotłowski W., Słowiński R.: Ender: a statistical framework for boosting decision rules. Data Mining and Knowledge Discovery, Vol. 21, 2010, p. 52÷90.
8. Nguyen H. S: Approximate boolean reasoning: foundations and applications in data mining, [in:] Peters J. F., Skowron A. (eds.): T. Rough Sets. LNCS, Vol. 4100, Springer, 2006, p. 334÷506.
9. Pawlak Z., Skowron A.: Rough sets and boolean reasoning. Information Science, Vol. 177, 2007, p. 41÷73.
10. Sikora M., Wróbel Ł.: Data-driven adaptive selection of rules quality measures for improving the rules induction algorithm, [in:] Kuznetsov S. O., Ślęzak D., Hepting D. H., Mirkin B. (eds.): RSFDGrC 2011. LNCS, Vol. 6743, Springer, 2011, p. 278÷285.
11. Skowron A., Rauszer C.: The discernibility matrices and functions in information systems, [in:] Słowiński R. (ed.): Intelligent Decision Support, Handbook of Applications and Advances of the Rough Set Theory. Kluwer Academic Publishers, Dordrecht 1992, p. 331÷362.
12. Zielosko B., Moshkov M.: Approximate algorithm for β-decision rule optimization. Studia Informatica, Vol. 32, No. 2A (96), Gliwice 2011, p. 329÷335.

Wpłynęło do Redakcji 5 stycznia 2013 r.

Omówienie

W artykule zaproponowano heurystykę na podstawie algorytmu dynamicznego programowania dla optymalizacji dokładnych reguł decyzyjnych odnośnie do pokrycia [3]. Dla danej tablicy decyzjnej *T* konstruowany jest skierowany graf acykliczny Δ(*T*). Węzłami tego grafu są podtabele tabeli *T* opisane przez system deskryptorów „atrybut=wartość”. W porównaniu z klasycznym algorytmem zaproponowanym w [3] podtabele tworzone są tylko dla jednej wartości każdego atrybutu. Jest to wartość, która występuje dla największej liczby wierszy. W związku z tym rozmiar konstruowanego grafu jest mniejszy, co ma znaczenie z punktu widzenia skalowalności, rozmiaru danej tablicy decyzyjnej. Na podstawie grafu Δ(*T*) można opisać zbiór reguł dla tablicy *T* i wiersza *r*. Następnie, w wyniku optymalizacji grafu Δ(*T*) względem pokrycia, można opisać zbiór reguł dla tablicy *T* i wiersza *r* o maksymalnym pokryciu.

Celem przeprowadzonych badań jest: (i) zbadanie pokrycia reguł konstruowanych za pomocą proponowanego algorytmu oraz porównanie z pokryciem reguł konstruowanych za pomocą algorytmu dynamicznego programowania, (ii) zbadanie rozmiaru grafu (liczba węzłów i krawędzi w skierowanym grafie acyklicznym) skonstruowanego za pomocą proponowanego algorytmu oraz porównanie go z rozmiarem grafu skonstruowanego za pomocą algorytmu dynamicznego programowania.

Address

Beata ZIELOSKO: King Abdullah University of Science and Technology, Computer, Electrical and Mathematical Sciences and Engineering Division, Thuwal, 23955-6900, Saudi Arabia, beata.zielosko@kaust.edu.sa; University of Silesia, Institute of Computer Science, 39, Będzińska St., Sosnowiec, 41-200, Poland, beata.zielosko@us.edu.pl.