STUDIA INFORMATICA 2012
Volume 33 Number 2A (105)

Fawaz ALSOLAMI, Igor CHIKALOV, Mikhail MOSHKOV
KAUST, MCSE Division
Beata ZIELOSKO
KAUST, MCSE Division
University of Silesia, Institute of Computer Science

OPTIMIZATION OF INHIBITORY DECISION RULES RELATIVE TO LENGTH

Summary. The paper is devoted to the study of an algorithm for optimization of inhibitory rules relative to the length. Such rules on the right-hand side have a relation "attribute ≠ value".

The considered algorithm is based on an extension of dynamic programming. After the procedure of optimization relative to length, we obtain a graph $\Lambda(T)$ which describes all nonredundant inhibitory rules with minimum length.

Keywords: inhibitory decision rules, length, dynamic programming

OPTYMALIZACJA WZBRANIAJĄCYCH REGUŁ DECYZYJNYCH WZGLĘDEM DŁUGOŚCI

Streszczenie. W artykule przedstawiono algorytm dla optymalizacji reguł wzbraniających względem długości. Reguły te w prawej części mają relację „atrybut ≠ wartość”.

Algorytm opiera się na idei dynamicznego programowania. Dla danej tablicy decyzyjnej T konstruowany jest skierowany graf acykliczny $\Lambda(T)$. W wyniku procedury optymalizacji względem długości, na podstawie grafu $\Lambda(T)$ można opisać cały zbiór nienadmiarowych reguł wzbraniających o minimlanej długości.

Słowa kluczowe: wzbraniające reguły decyzyjne, długość, algorytm dynamicznego programowania

1. Introduction

The paper is devoted to the study of an algorithm for inhibitory rule optimization based on extensions of dynamic programming. In contrast with usual rules that have on the right-hand side a relation "attribute = value", inhibitory rules have a relation "attribute ≠ value" on the right hand side.

It was shown in [10, 11] that, for some information systems, usual rules cannot describe the whole information contained in the system. However, inhibitory rules describe the whole information for every information system [7]. Classifiers based on inhibitory rules have often better accuracy than classifiers based on usual rules [4, 5, 6].

Greedy algorithms for inhibitory rule construction were studied in [7]. In this paper, we consider an algorithm for optimization of inhibitory rules relative to the length which is based on an extension of dynamic programming. The choice of length is connected with the Minimum Description Length principle [9]. Similar approach to usual decision rule optimization was studied in [1, 2, 3, 12]. We consider also results of experiments with some decision tables from UCI ML Repository [8].

The paper consists of six sections. In Section 2, we discuss main notions including the notion of nonredundant inhibitory rule. In Section 3, a directed acyclic graph is considered which allows us to describe the whole set of nonredundant inhibitory rules for each row of a decision table. Section 4 contains the descriptions of a procedure of optimization relative to the length. Section 5 contains results of experiments and Section 6 – conclusions.

2. Nonredundant Inhibitory Rules

First, we consider definitions of notions corresponding to decision tables and inhibitory rules. 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 which are interpreted as values of conditional attributes. Rows of T are pairwise different and each row is labeled with a nonnegative integer (decision) which is interpreted as a value of the decision attribute d. We denote by $D(T)$ the set of decisions attached to rows of the table T. We denote by $N(T)$ the number of rows in the table T.

A table obtained from T by the removal of some rows is called a *subtable* of the table T. A subtable T^* of the table T is called *reduced* if $|D(T^*)| < |D(T)|$, and *unreduced* otherwise when $|D(T^*)| = |D(T)|$.

Let T be nonempty, $f_{i(1)},\ldots,f_{i(m)}$ $\{f_1,\ldots,f_n\}$ and $a_1,\ldots,a_m$ be nonnegative integers. We denote by $T(f_{i(1)},a_1)\ldots(f_{i(\mathrm{m})},a_m)$ the subtable of the table T which contains only rows that have numbers $a_1,\ldots,a_m$ at the intersection with columns $f_{i(1)},\ldots,f_{i(m)}$. Such nonempty subtables (including the table T) are called *separable subtables* of T.

We denote by $E(T)$ the set of attributes from $\{f_1,\ldots,f_n\}$ which are not constant on T. For any f_i $E(T)$, we denote by $E(T,f_i)$ the set of values of the attribute f_i in T.

The expression

$$f_{i(1)} = a_1 \wedge \ldots \wedge f_{i(m)} = a_m \rightarrow d \neq k \tag{1}$$

is called an *inhibitory rule over T* if $f_{i(1)},\ldots,f_{i(m)}$ $\{f_1,\ldots,f_n\}$, $a_1,\ldots a_m$ are nonnegative integers, and k $D(T)$. It is possible that $m = 0$. In this case (1) is equal to the rule

$$\rightarrow \mathrm{d} \neq \mathrm{k}. \tag{2}$$

Let Θ be a subtable of T and $r=(b_1,\ldots,b_n)$ be a row of Θ. We will say that the rule (1) is *realizable for r*, if $a_1=b_{i(1)},\ldots,a_m=b_{i(m)}$. The rule (2) is realizable for any row from Θ.

We will say that the rule (1) is *true for* Θ if each row of Θ for which the rule (1) is realizable has the decision attached to it that is different from k. The rule (2) is true for Θ if and only if each row of Θ is labeled with the decision different from k. If the rule (1) is an inhibitory rule over T which is true for Θ and realizable for r, we will say that (1) is an *inhibitory rule for* Θ *and* r *over* T.

We will say that the rule (1) with $m > 0$ is a *nonredundant* inhibitory rule for Θ and r over T if (1) is an inhibitory rule for Θ and r over T and the following conditions hold:

(i) $f_{i(1)}$ $E(\Theta)$, and if $m > 1$ then $f_{i(j)}$ $E(\Theta(f_{i(1)},a_1)\ldots(f_{i(j-1)},a_{j-1}))$ for $j=2,\ldots,m$;

(ii) if $m=1$ then Θ is unreduced, and if $m>1$ then the subtable $\Theta^*=\Theta(f_{i(1)},a_1)\ldots(f_{i(m-1)},a_{m-1})$ is unreduced.

The rule (2) is a *nonredundant* inhibitory rule for Θ and r over T if (2) is an inhibitory rule for Θ and r over T, i.e., if each row of Θ is labeled with a decision different from k and $k \in D(T)$.

Lemma 1. *Let* Θ *be an unreduced subtable of T with* $f_{i(1)} \in E(\Theta)$, $a_1 \in E(\Theta,f_{i(1)})$, and r *be a row of the table* $\Theta^*= \Theta(f_{i(1)},a_1)$. *Then the rule* (1) *with* $m \geq 1$ *is a nonredundant inhibitory rule for* Θ *and r over T if and only if the rule*

$$f_{i(2)} = a_2 \wedge \ldots \wedge f_{i(m)} = a_m \rightarrow d \neq k \tag{3}$$

is a nonredundant inhibitory rule for Θ^* *and r over T* (*if* m=1 *then* (3) *is equal to* $\rightarrow d \neq k$).

Proof. It is clear that (1) is an inhibitory rule for Θ and r over T if and only if (3) is an inhibitory rule for Θ^* and r over T.

It is easy to show that the statement of lemma holds if $m = 1$. Let now $m > 1$.

Let (1) be a nonredundant inhibitory rule for Θ and r over T. Then from (i) it follows that $f_{i(2)} \in E(\Theta^*)$ and if $m > 2$ then, for $j=3,\ldots,m$, $f_{i(j)} \in E(\Theta^*(f_{i(2)},a_2)\ldots(f_{i(j-1)},a_{j-1}))$. From (ii) it follows that Θ^* is unreduced if m=2, and $\Theta^*(f_{i(2)},a_1)\ldots(f_{i(m-1)},a_{m-1})$ is unreduced if $m > 2$. Therefore (3) is a nonredundant inhibitory rule for Θ^* and r over T.

Let (3) be a nonredundant inhibitory rule for Θ^* and r over T. Then, for j=2,…,m, $f_{i(j)} \in E(\Theta(f_{i(1)},a_1)\ldots(f_{i(j-1)},a_{j-1}))$. Also we know that $f_{i(1)} \in E(\Theta)$. Therefore the condition (i) holds. Since (3) is a nonredundant inhibitory rule for Θ^* and r over T, we have $\Theta(f_{i(1)},a_1)$ is unreduced if m=2 and $\Theta(f_{i(1)},a_1)\ldots(f_{i(m)},a_{m-1})$ is unreduced if $m > 2$. Therefore the condition (ii) holds, and (1) is a nonredundant inhibitory rule for Θ and r over T. □

Let Θ be a subtable of T, τ be a nonredundant rule over T, and τ be equal to (1).

The number m of conditions on the left-hand side of τ is called the *length* of this rule and is denoted by $l(\tau)$. The length of inhibitory rule (2) is equal to 0.

Proposition 1. *Let T be a nonempty decision table,* Θ *be a nonempty subtable of T, r be a row of* Θ*, and* τ *be an inhibitory rule for* Θ *and r over T which is not a nonredundant inhibitory rule for* Θ *and r over T. Then by removal of some conditions from the left-hand side of* τ *and changing of the right-hand side of* τ *we can obtain a nonredundant inhibitory rule irr*(τ) *for* Θ *and r* over *T such that* $l(irr(\tau)) \leq l(\tau)$.

Proof. Let τ be equal to (1) and $p \in D(T)\setminus D(\Theta)$. One can show that the rule $\rightarrow d \neq p$ is a nonredundant inhibitory rule for Θ and r over T. We denote this rule by $irr(\tau)$. It is clear that $l(irr(\tau)) \leq l(\tau)$. Let now Θ be unreduced.

Let t be the minimum number from $\{1,\ldots,m\}$ such that $\Theta^*=\Theta(f_{i(1)},a_1)\ldots(f_{i(t)},a_t)$ is reduced. If $t < m$ then we remove from τ the conditions $f_{i(t+1)}=a_{t+1},\ldots,f_{i(m)}=a_m$ and instead of $d \neq k$ we will have $d \neq q$ where $q \in D(T)\setminus D(\Theta^*)$. We denote the obtained rule by τ^*. It is clear that τ^* is an inhibitory rule for Θ and r over T. If $f_{i(1)} \notin E(\Theta)$ then we remove the condition $f_{i(1)}=a_1$ from τ^*. For any $j \in \{2,\ldots,t\}$, if $f_{i(j)} \notin E(\Theta(f_{i(1)},a_1)\ldots(f_{i(j-1)},a_{j-1}))$ then we remove the condition $f_{i(j)}=a_j$ from the left-hand side of the rule τ^*.

One can show that the obtained rule is a nonredundant inhibitory rule for Θ and r over T. We denote this rule by $irr(\tau)$. It is clear that $l(\tau) \geq l(irr(\tau))$. □

3. Directed Acyclic Graph $\Lambda(T)$

Now, we consider an algorithm that constructs a directed acyclic graph $\Lambda(T)$ which will be used to describe the set of nonredundant inhibitory rules for T and for each row r of T over T. Nodes of the graph are some 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 which 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 $\Lambda(T)$. Otherwise, choose a node (table) Θ, which has not been processed yet. If Θ is reduced, then mark Θ with the symbol * and go to the step (p+2). Otherwise, for each $f_i \in E(\Theta)$, draw a bundle of edges from the node Θ. Let $E(\Theta,f_i)=\{b_1,\dots,b_t\}$. Then draw t edges from Θ and label these edges with pairs $(f_i,b_1),\dots,(f_i,b_t)$ respectively. These edges enter to nodes $\Theta(f_i,b_1),\dots,\Theta(f_i,b_t)$. If some of nodes $\Theta(f_i,b_1),\dots,\Theta(f_i,b_t)$ are absent in the graph then add these nodes to the graph. We label each row r of Θ with the set of attributes $E_{\Lambda(T)}(\Theta,r)=E(\Theta)$ (this set can be changed during a procedure of optimization). Mark the node Θ with the symbol * and proceed to the step (p+2).

The graph $\Lambda(T)$ 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 $\Lambda(T)$ is terminal if and only if Θ is reduced.

Later, we will describe a procedure of optimization of the graph $\Lambda(T)$ relative to the length. As a result we will obtain a graph Γ with the same sets of nodes and edges as in $\Lambda(T)$. The only difference is that any row r of each unreduced table Θ from Γ is labeled with a nonempty set of attributes $E_\Gamma(\Theta,r)\quad E(\Theta)$.

Let G be the graph $\Lambda(T)$ or a graph Γ obtained from $\Lambda(T)$ by the procedure of optimization.

Now for each node Θ of G and for each row r of Θ we describe a set of inhibitory rules $Rul_G(\Theta,r)$ over T. Let Θ be a terminal node of G: Θ is a reduced subtable. Then

$$Rul_G(\Theta,r)=\{\to d \neq k\colon k \in D(T)\backslash D(\Theta)\}.$$

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 $Rul_G(\Theta^*,r^*)$ is already defined. Let $r=(b_1,\dots,b_n)$ be a row of Θ. For any $f_i \in E_G(\Theta,r)$, we define the set of rules $Rul_G(\Theta,r,f_i)$ as follows:

$$Rul_G(\Theta,r,f_i)=\{f_i=b_i\quad \alpha\to d\neq k\colon \alpha\to d\neq k \in Rul_G(\Theta(f_i,b_i),r)\}$$

Then

$$Rul_G(\Theta,r)=\bigcup_{f_i\in E_G(\Theta,r)} Rul_G(\Theta,r,f_i$$

Theorem 1. *For any node Θ of $\Lambda(T)$ and for any row r of Θ, the set $Rul_{\Lambda(T)}(\Theta,r)$ is equal to the set of all nonredundant inhibitory rules for Θ and r over T.*

Proof. We will prove this statement by induction on nodes in $\Lambda(T)$. Let Θ be a terminal node of $\Lambda(T)$. One can show that the rules $\to d \neq k$ where $k \in D(T)\backslash D(\Theta)$ are the only rules which are nonredundant inhibitory rules for Θ and r over T. Therefore the set $Rul_{\Lambda(T)}(\Theta,r)$ is equal to the set of all nonredundant inhibitory rules for Θ and r over T.

Let Θ be a nonterminal node of $\Lambda(T)$ and for each child of Θ the statement of theorem hold. Let $r=(b_1,\ldots,b_n)$ be a row of Θ. Using Lemma 1 we obtain that $Rul_{\Lambda(T)}(\Theta,r)$ contains only nonredundant inhibitory rules for Θ and r over T.

Let τ be a nonredundant inhibitory rule for Θ and r over T. Since Θ is unreduced, the left-hand side of τ is nonempty. Therefore τ can be represented in the form $f_i=b_i\wedge\alpha\rightarrow d\neq k$, where $f_i\in E(\Theta)$. Using Lemma 1 we obtain $\alpha\rightarrow d\neq k$ is a nonredundant inhibitory rule for $\Theta(f_i,b_i)$ and r over T. Based on inductive hypothesis we obtain that the rule $\alpha\rightarrow d\neq k$ belongs to the set $Rul_{\Lambda(T)}(\Theta(f_i,b_i),r)$. Therefore $\tau\in Rul_{\Lambda(T)}(\Theta,r)$. □

Let us consider a decision table T_0 depicted in Fig. 1.

T_0 =

f_1	f_2	f_3	d
1	1	1	1
0	0	0	0
1	0	0	1
0	1	0	2

Fig. 1. Decision table T_0
Rys. 1. Tabela decyzyjna T_0

We denote by G_0 the graph $\Lambda(T_0)$ which is depicted in Fig. 2. For each node (subtable) Θ of G_0 which contains the last row r_4 of the table T_0 we add to Θ the set of all nonredundant inhibitory rules for Θ and r_4 over T_0.

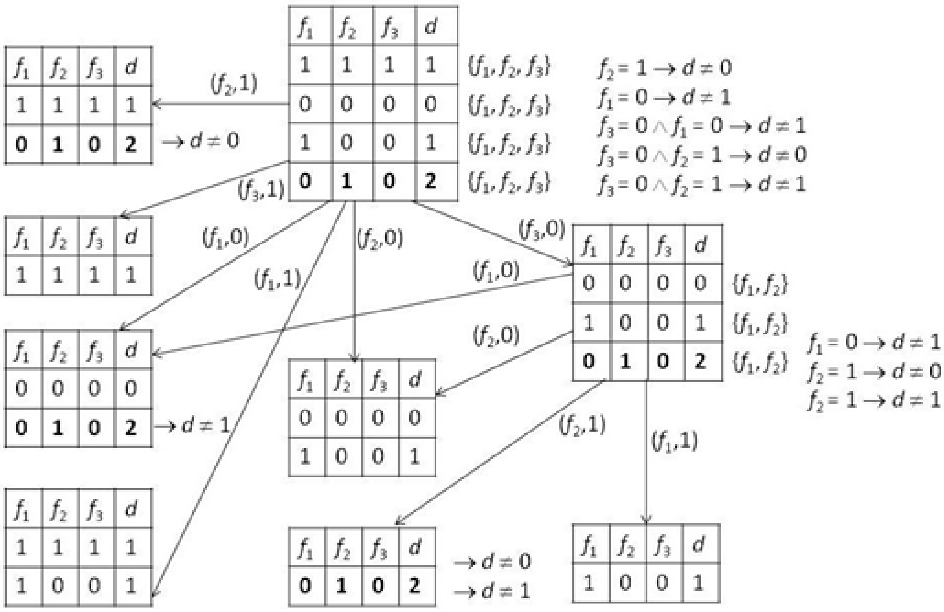

Fig. 2. Graph $G_0=\Lambda(T_0)$
Rys. 2. Graf $G_0=\Lambda(T_0)$

4. Procedure of Optimization Relative to Length

Let now $G = \Lambda(T)$. We consider the procedure of optimization of the graph G relative to the length l. For each node Θ in the graph G, this procedure assigns to each row r of Θ the set $Rul_G^l(\Theta,r)$ of inhibitory rules with minimum length from $Rul_G(\Theta,r)$ and the number $Opt_G^l(\Theta,r)$ – the minimum length of an inhibitory rule from $Rul_G(\Theta,r)$.

The idea of the procedure is simple. It is clear that for each terminal node Θ of G and for each row r of Θ the following equalities hold:

$$Rul_G^l(\Theta,r) = Rul_G(\Theta,r) = \{\rightarrow d \neq k\colon k \in D(T)\backslash D(\Theta)\},$$

and

$$Opt_G^l(\Theta,r) = 0.$$

Let Θ be a nonterminal node, and $r=(b_1,\ldots,b_n)$ be a row of Θ. We know that

$$Rul_G(\Theta,r) = \bigcup_{f_i \in E_G(\Theta,r)} Rul_G(\Theta,r,f_i$$

and, for $f_i \in E_G(\Theta,r)$,

$$Rul_G(\Theta,r,f_i) = \{f_i=b_i \wedge \alpha\rightarrow d \neq k\colon \alpha\rightarrow d \neq k \in Rul_G(\Theta(f_i,b_i),r)\}.$$

For $f_i \in E_G(\Theta,r)$, we denote by $Rul_G^l(\Theta,r,f_i)$ the set of all rules with the minimum length from $Rul_G(\Theta,r,f_i)$ and by $Opt_G^l(\Theta,r,f_i)$ – the minimum length of an inhibitory rule from $Rul_G(\Theta,r,f_i)$.

One can show that

$$Rul_G^l(\Theta,r,f_i) = \{f_i=b_i \wedge \alpha\rightarrow d \neq k\colon \alpha\rightarrow d \neq k \in Rul_G^l(\Theta(f_i,b_i),r)\},$$

$$Opt_G^l(\Theta,r,f_i) = Opt_G^l(\Theta(f_i,b_i),r) + 1,$$

and

$$Opt_G^l(\Theta,r) = \min \{Opt_G^l(\Theta,r,f_i)\colon f_i \in E_G(\Theta,r)\}$$

$$=\min\{Opt_G^l(\Theta(f_i,b_i),r) + 1\colon f_i \in E_G(\Theta,r)\}.$$

It is easy to see also that

$$Rul_G^l(\Theta,r) = \bigcup_{f_i \in E_G(\Theta,r), Opt_G^l(\Theta(f_i,b_i),r)+1=Opt_G^l(\Theta,r)} Rul_G^l(\Theta,r,f_i).$$

We now describe the procedure of optimization of the graph G relative to the length l. We will move from the terminal nodes of the graph G which are reduced subtables to the node T. We will assign to each row r of each table Θ the number $Opt_G^l(\Theta,r)$ which is the minimum length of an inhibitory rule from $Rul_G(\Theta,r)$ 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 $G(l)$.

Let Θ be a terminal node of G. Then we assign to each row r of Θ the number $Opt_G^l(\Theta,r) = 0$.

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

$$Opt_G^l(\Theta,r) = \min\{Opt_G^l(\Theta(f_i,b_i),r) + 1:\ f_i \in E_G(\Theta,r)\}$$

to the row r in the table Θ and we set

$$E_{G(l)}(\Theta,r) = \{f_i:\ f_i \in E_G(\Theta,r),\ Opt_G^l(\Theta(f_i,b_i),r) + 1 = Opt_G^l(\Theta,r)\}.$$

From the reasoning before the description of the procedure of optimization relative to the length the next statement follows.

Theorem 2. *For each node Θ of the graph $G(l)$ and for each row r of Θ the set $Rul_{G(l)}(\Theta,r)$ is equal to the set $Rul_G^l(\Theta,r)$ of all rules with the minimum length from the set $Rul_G(\Theta,r)$.*

Fig. 3 presents the directed acyclic graph $G_0(l)$ obtained from the graph G_0 (see Fig. 2) by the procedure of optimization relative to the length. For each node (subtable) Θ of $G_0(l)$ which contains the last row r_4 of the table T_0 we add to Θ the set of all nonredundant inhibitory rules for Θ and r_4 over T_0 with minimum length.

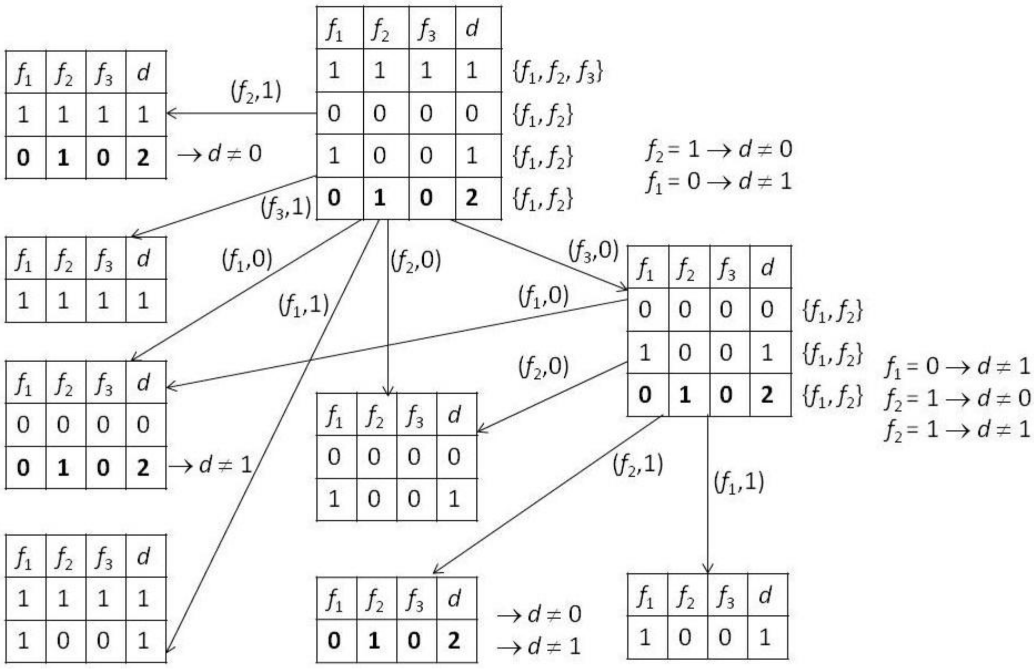

Fig. 3. Graph $G_0(l)$
Rys. 3. Graf $G_0(l)$

5. Experimental Results

We considered a number of decision tables from UCI Machine Learning Repository [8]. 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.

For each such decision table T we constructed the directed acyclic graph $\Lambda(T)$ and applied to it the procedure of optimization relative to the length. Average length of obtained rules (among all rows of T) can be found in Table 1 (column “DP”).

We used also a greedy algorithm to construct inhibitory rules. For a given row r of a decision table T, we form the set $U(T,r)$ of all rows of T with decisions different from the decision attached to r. During each step we choose an attribute that separates from r the maximum number of rows from $U(T,r)$ not yet separated. We will stop when lose at least one decision from the set $D(T)$. Average length of obtained rules (among all rows of T) can be found in Table 1 (column “Greedy”).

Table 1

Length and coverage of inhibitory rules

Decision table	Rows	Attr	DP	Greedy
adult-stretch	16	4	1.250	1.250
balance-scale	625	4	2.672	2.704
breast-cancer	266	9	2.665	2.726
cars	1727	6	1.047	1.459
hayes-roth-data	69	4	1.667	1.667
lymphography	148	18	1.000	1.135
monks-1-test	432	6	2.250	2.250
monks-1-train	124	6	2.266	2.476
monks-2-test	432	6	4.523	4.861
monks-2-train	169	6	3.497	3.692
monks-3-test	432	6	1.750	1.750
monks-3-train	122	6	2.311	2.336
nursery	12960	8	1.000	1.129
shuttle-landing	15	6	1.400	1.400

Based on presented results we can see that for nine from 14 decision tables the average length of optimal rules is less than the average length of rules constructed by the greedy algorithm.

6. Conclusions

In the paper, we considered algorithm for exact inhibitory rule optimization relative to the length which is based on an extension of dynamic programming. Further, we will study approximate inhibitory rules also.

BIBLIOGRAPHY

1. Amin T., Chikalov I., Moshkov M., Zielosko B.: Dynamic programming algorithm for optimization of β-decision rules, [in:] Szczuka M., Czaja L., Skowron A., Kacprzak M. (eds.): 20th International Workshop Concurrency, Specification and Programming CS&P 2011, Białystok University of Technology, Pułtusk, Poland 2011, p. 10÷16.
2. Amin T., Chikalov I., Moshkov M., Zielosko B.: Dynamic programming approach for exact decision rule optimization, [in:] Skowron A., Suraj Z. (eds.): Special volume in Series Intelligent Systems Reference Library dedicated to the memory of Professor Zdzisław Pawlak, Springer, 2011.
3. Amin T., Chikalov I., Moshkov M., Zielosko B.: Dynamic programming approach to optimization of approximate decision rules. Information Sciences, Elsevier (submitted).
4. Delimata P., Moshkov M., Skowron A., Suraj Z.: Two families of classification algorithms, [in:] An A., Stefanowski J., Ramanna S., Butz C. J., Pedrycz W., Wang G. (eds.): RSFDGrC 2007, LNCS (LNAI), Vol. 4482. Springer, Heidelberg 2007, p. 297÷304.
5. Delimata P., Moshkov M., Skowron A., Suraj Z.: Comparison of lazy classification algorithms based on deterministic and inhibitory decision rules, [in:] Wang G., Li T., Grzymała-Busse J. W., Miao D., Skowron A., Yao Y. (eds.): RSKT2008, LNCS (LNAI), Vol. 5009. Springer, Heidelberg 2008, p. 55÷62.
6. Delimata P., Moshkov M., Skowron A., Suraj Z.: Lazy classification algorithms based on deterministic and inhibitory rules, [in:] Magdalena L., Ojeda-Aciego M., Verdegay J. L. (eds.): IPMU'08, Torremolinos (Malaga), Spain, June 22-27, 2008, p. 1773÷1778.
7. Delimata P., Moshkov M., Skowron A., Suraj Z.: Inhibitory Rules in Data Analysis: A Rough Set Approach. Computational Intelligence, Vol. 163, Springer, Heidelberg 2009.
8. Frank A., Asuncion A.: UCI ML Repository, http://archive.ics.uci.edu/ml.
9. Rissanen J.: Modeling by shortest data description. Automatica, Vol. 14, 1978, p. 465÷471.

10. Skowron A., Suraj Z.: Rough sets and concurrency. Bulletin of the Polish Academy of Sciences, Vol. 41(3), 1993, p. 237÷254.
11. Suraj Z.: Some remarks on extensions and restrictions of information systems, [in:] Ziarko W., Yao Y. Y. (eds.): RSCTC 2000, LNCS (LNAI), Vol. 2005, Springer, Heidelberg 2001, p. 204÷211.
12. Zielosko B., Moshkov M., Chikalov I.: Optimization of decision rules based on methods of dynamic programming. Vestnik of Lobachevsky State University of Nizhni Novgorod 6, 2010, p. 195÷200 (in Russian).

Wpłynęło do Redakcji 23 grudnia 2011 r.

Omówienie

W artykule został przedstawiony algorytm dla optymalizacji wzbraniających reguł decyzyjnych względem długości. W porównaniu do zwykłych reguł decyzyjnych, które w prawej części relację „atrybut = wartość", reguły wzbraniające w prawej części mają relację „atrybut ≠ wartość".

Przedstawiony algorytm opiera się na ideii dynamicznego programowania. Dla danej tablicy decyzyjnej T konstruowany jest skierowany graf acykliczny $\Lambda(T)$. Węzłami grafu są podtabele tabeli T, opisane przez system równań „atrybut = wartość". Podział tabeli na podtabele kończy się, kiedy podtabela ma mniej różnych wartości decyzji niż tabela T. Na podstawie grafu $\Lambda(T)$ można opisać cały zbiór tzw. nienadmiarowych reguł wzbraniających. W wyniku optymalizacji wzgledem długości uzyskujemy zmieniony graf $\Lambda(T)$, który pozwala opisać cały zbiór nienadmiarowych reguł wzbraniających o minimalnej długości.

Wyniki eksperymentów przedstawiają średnią długość reguł, opisanych na podstawie algorytmu dynamicznego programowania, oraz średnią długość reguł konstruowanych za pomocą algorytmu zachłannego.

Addresses

Fawaz ALSOLAMI: King Abdullah University of Science and Technology, Mathematical and Computer Sciences &Engineering Division, Thuwal, 23955-6900, Saudi Arabia, fawaz.alsolami@kaust.edu.sa.

Igor CHIKALOV: King Abdullah University of Science and Technology, Mathematical and Computer Sciences &Engineering Division, Thuwal, 23955-6900, Saudi Arabia, igor.chikalov@kaust.edu.sa.

Mikhail MOSHKOV: King Abdullah University of Science and Technology, Mathematical and Computer Sciences &Engineering Division, Thuwal, 23955-6900, Saudi Arabia, mikhail.moshkov@kaust.edu.sa.

Beata ZIELOSKO: King Abdullah University of Science and Technology, Mathematical and Computer Sciences &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.