Title: Optimization of approximate inhibitory rules relative to number of misclassifications

Author: F. Alsolami, I. Chikalov, M. Moshkov, Beata Zielosko

Citation style: Alsolami F., Chikalov I., Moshkov M., Zielosko Beata. (2013). Optimization of approximate inhibitory rules relative to number of misclassifications. "Procedia Computer Science" (Vol. 22, (2013), s. 295-302), doi 10.1016/j.procs.2013.09.106

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 22 (2013) 295 – 302

Procedia Computer Science

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

Optimization of Approximate Inhibitory Rules Relative to Number of Misclassifications

Fawaz Alsolami[a,b,*], Igor Chikalov[a], Mikhail Moshkov[a], Beata Zielosko[a,c]

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

Abstract

In this work, we consider so-called nonredundant inhibitory rules, containing an expression "attribute≠ value" on the right-hand side, for which the number of misclassifications is at most a threshold γ. We study a dynamic programming approach for description of the considered set of rules. This approach allows also the optimization of nonredundant inhibitory rules relative to the length and coverage [1, 2]. The aim of this paper is to investigate an additional possibility of optimization relative to the number of misclassifications. The results of experiments with decision tables from the UCI Machine Learning Repository [3] show this additional optimization achieves a fewer misclassifications. Thus, the proposed optimization procedure is promising.

Selection and peer-review under responsibility of KES International

Keywords: inhibitory rules; number of misclassifications; dynamic programming

1. Introduction

Decision rules are popular way for knowledge representation and pattern discovery in data mining. Many approaches have been proposed in the literature to obtain rules either directly from datasets, for example, boolean reasoning [4, 5], separate-and-conquer [6, 7], or from models such as decision trees [8, 9, 10].

In this paper, we present an approach for optimization of inhibitory rules based on a dynamic programming algorithm. Inhibitory rules have in the consequent part a relation "attribute ≠ value" whereas decision (deterministic) rules have "attribute = value". In [11, 12] it was shown that decision rules cannot describe the whole information contained in some information systems. However, inhibitory rules describe the whole information for every information system [13]. Moreover, classifiers based on inhibitory rules have often better accuracy than classifiers based on decision rules [14, 15, 16].

Due to overfitting problem, approximate rules are more appropriate since real datasets often contain noise. Moreover, exact decision rules are often not feasible in terms of computational resources.

*Corresponding author.
E-mail address: fawaz.alsolami@kaust.edu.sa.

Selection and peer-review under responsibility of KES International
doi:10.1016/j.procs.2013.09.106

In [13] greedy algorithms for inhibitory rules construction were studied. In [1, 2] we presented a dynamic programming approach for construction and optimization of exact inhibitory rules relative to the length and coverage. We also considered sequential optimization of inhibitory rules relative to the length and coverage. Moreover, some comparisons of the length and coverage of inhibitory rules constructed by the greedy algorithm and dynamic programming were presented.

In this paper, we consider so-called nonredundant γ-inhibitory rules for which the number of misclassifications is at most a predefined threshold γ. We study a dynamic programming technique to describe the set of nonredundant γ-inhibitory rules. As mentioned above, this technique allows optimization of inhibitory rules relative to the length and coverage. The main objective of this paper is to investigate an additional possibility of optimization relative to the number of misclassifications. We also present results of experiments with some decision tables from UCI Machine Learning Repository [3] based on our software system Dagger [17]. The results show that the minimum number of misclassifications of nonredundant γ-inhibitory rules is, usually, essentially less than a threshold γ, i.e., the additional optimization relative to the number of misclassifications is useful. A similar study for the optimization of usual decision rules relative to the number of misclassification was conducted in [18]. However, the procedure proposed in this paper achieves a fewer misclassifications even for the case of binary attribute on the right-hand side.

This paper consists of seven sections. Section 2 contains definitions of main notions. In Sect. 3, we study a directed acyclic graph which allows a description of the whole set of nonredundant γ-inhibitory rules. The work of an optimization procedure of nonredundant γ-inhibitory rules relative to the number of misclassifications is well explained in Sect. 4. Section 5 contains results of experiments with decision tables from UCI Machine Learning Repository and finally, Sect. 6 contains the conclusion.

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 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 distinct decisions for the table T. We denote by $N(T)$ the number of rows in the table T.

The least common decision for T is a decision from the set $D(T)$ attached to the minimum number of rows in T. If we have a number of such decisions then we choose the minimum one. By $N_{lcd}(T)$ we denote the number of rows in the table T labeled with the least common decision for T.

Let T be nonempty, $f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\}$ and $v_1, \ldots, v_m$ be nonnegative integers. By $T(f_{i_1}, v_1) \ldots (f_{i_m}, v_m)$ we denote a subtable of the table T which contains only rows that have values $v_1, \ldots, v_m$ 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.

A subtable T' of the table T is called *reduced* if and only if $|D(T')| < |D(T)|$ and *unreduced* otherwise. Since at least one decision is missing from the set $D(T)$ for a reduced subtable T', then $N_{lcd}(T') = 0$ for a reduced subtable.

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

The expression

$$f_{i_1} = v_1 \wedge \ldots \wedge f_{i_m} = v_m \rightarrow d \neq c \tag{1}$$

is called an *inhibitory rule over* T if $f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\}$, $v_1, \ldots v_m$ are nonnegative integers, and $c \in D(T)$. It is possible that $m = 0$. In this case (1) is equal to the rule

$$\rightarrow d \neq c \tag{2}$$

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 $v_1 = b_{i_1}, \ldots, v_m = b_{i_m}$. The rule (2) is realizable for any row from Θ.

Let γ be a nonnegative real number. We will say that the rule (1) is γ-*true for* Θ if c is the least common decision for $\Theta' = \Theta(f_{i_1}, v_1) \ldots (f_{i_m}, v_m)$ and $N_{lcd}(\Theta') \leq \gamma$. If $m = 0$ then the rule (2) is γ-true for Θ if c is the least common decision for Θ and $N_{lcd}(\Theta) \leq \gamma$.

If the rule (1) is an inhibitory rule over T which is γ-true for Θ and realizable for r, we will say that (1) is a *γ-inhibitory rule for Θ and r over T*. If $\gamma = 0$ then we have the notion of exact inhibitory rule. Otherwise, we have the notion of approximate inhibitory rule.

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

(i) $f_{i_1} \in E(\Theta)$, and if $m > 1$ then $f_{i_j} \in E(\Theta(f_{i_1}, v_1) \dots (f_{i_{j-1}}, v_{j-1}))$ for $j = 2, \dots, m$;

(ii) $N_{lcd}(\Theta) > \gamma$, and if $m > 1$ then $N_{lcd}(\Theta(f_{i_1}, v_1) \dots (f_{i_j}, v_j)) > \gamma$ for $j = 1, \dots, m-1$.

If $m = 0$ then the rule (2) is a *nonredundant* γ-inhibitory rule for Θ and r over T if (2) is a γ-inhibitory rule for Θ and r over T, i.e., if c is the least common decision for Θ and $N_{lcd}(\Theta) \le \gamma$.

Let Θ be a subtable of T, τ be a rule over T and τ be equal to (1).The *number of misclassifications* of τ relative to Θ is the number of rows in Θ for which τ is realizable and which are labeled with the decision c. We denote it by $\mu(\tau, \Theta)$. The number of misclassifications of the rule (2) relative to Θ is equal to the number of rows in Θ which are labeled with the decision c.

Lemma 1. *Let Θ be an unreduced subtable of T with $f_{i_1} \in E(\Theta)$, $v_1 \in E(\Theta, f_{i_1})$, and r be a row of the table $\Theta' = \Theta(f_{i_1}, v_1)$. Then the rule (1) with $m \ge 1$ is a nonredundant γ-inhibitory rule for Θ and r over T if and only if the rule*

$$f_{i_2} = v_2 \wedge \dots \wedge f_{i_m} = v_m \to d \ne c \tag{3}$$

is a nonredundant γ-inhibitory rule for Θ' and r over T (if $m = 1$ then the rule (3) is equal to $\to d \ne c$)

Proof. It is clear that (1) is a γ-inhibitory rule for Θ and r over T if and only if (3) is a γ-inhibitory rule for Θ' and r over T. It is not difficult to show that the statement of lemma holds if $m = 1$. Let now $m > 1$. Let (1) be a nonredundant γ-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 $j = 3, \dots, m$, $f_{i_j} \in E(\Theta'(f_{i_2}, v_2) \dots (f_{i_{j-1}}, v_{j-1}))$. From (ii) it follows that $N_{lcd}(\Theta') > \gamma$ if $m = 2$, and $N_{lcd}(\Theta'(f_{i_2}, v_2) \dots (f_{i_{m-1}}, v_{m-1})) > \gamma$ when $m > 2$. Therefore (3) is a nonredundant γ-inhibitory rule for Θ' and r over T.

Let (3) be a nonredundant γ-inhibitory rule for Θ' and r over T. Then, for $j = 2, \dots, m$, $f_{i_j} \in E(\Theta(f_{i_1}, v_1) \dots (f_{i_{j-1}}, v_{j-1}))$. Also we know that $f_{i_1} \in E(\Theta)$. Therefore the condition (i) holds. Since (3) is a nonredundant γ-inhibitory rule for Θ' and r over T, we have $N_{lcd}(\Theta(f_{i_1}, v_1)) > \gamma$ if $m = 2$ and $N_{lcd}(\Theta(f_{i_1}, v_1) \dots (f_{i_{m-1}}, v_{m-1})) > \gamma$ if $m > 2$. Therefore the condition (ii) holds, and (1) is a nonredundant γ-inhibitory rule for Θ' and r over T. □

3. Directed Acyclic Graph $\Lambda_\gamma(T)$

We consider an algorithm that constructs a directed acyclic graph $\Lambda_\gamma(T)$ which will be used to describe the set of nonredundant γ-inhibitory rules for T and for each row r of T over 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 which is not marked with the symbol *.

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 $\Lambda_\gamma(T)$. Otherwise, choose a node (table) Θ, which has not been processed yet.

Let c be the least common decision for Θ. If $N_{lcd}(\Theta) \le \gamma$ label the considered node with the decision c, mark it with the symbol * and proceed to the step $(p+2)$. If $N_{lcd}(\Theta) > \gamma$, 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 $E_{\Lambda_\gamma(T)}(\Theta, r) = E(\Theta)$. Mark the node Θ with the symbol * and proceed to the step $(p+2)$.

The graph $\Lambda_\gamma(T)$ 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 $\Lambda_\gamma(T)$ is terminal if and only if $N_{lcd}(\Theta) \le \gamma$.

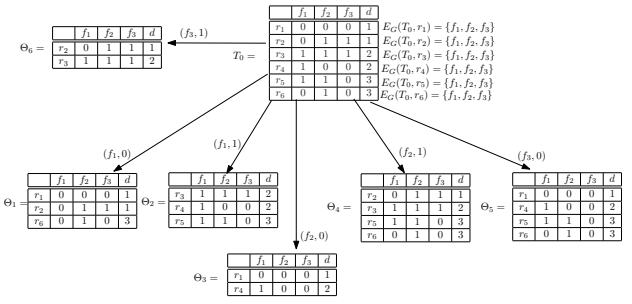

Fig. 1. Graph $G = \Lambda_1(T_0)$

Later, we describe the procedure of optimization of the graph $\Lambda_\gamma(T)$ relative to the number of misclassifications. As a result we obtain a graph $\Lambda_\gamma(T)^\mu$ with the same sets of nodes and edges as in $\Lambda_\gamma(T)$. The only difference is that any row r of each nonterminal node Θ of $\Lambda_\gamma(T)^\mu$ is labeled with a nonempty set of attributes $E_{\Lambda_\gamma(T)^\mu}(\Theta, r) \subseteq E(\Theta)$.

Let G be the graph $\Lambda_\gamma(T)$ or the graph $\Lambda_\gamma(T)^\mu$. For each node Θ of G and for each row r of Θ, we describe a set of γ-inhibitory rules $Rul_G(\Theta, r)$ over T. We move from terminal nodes of G to the node T.

Let Θ be a terminal node of G and c be the least common decision for Θ. Then

$$Rul_G(\Theta, r) = \{\rightarrow d \neq c\}.$$

Let now Θ be a nonterminal node of G such that for each child Θ' of Θ and for each row r' of Θ', a set of rules $Rul_G(\Theta', r')$ is already defined. Let $r = (b_1, \ldots, b_n)$ 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:

$$Rul_G(\Theta, r, f_i) = \{f_i = b_i \wedge \sigma \rightarrow d \neq s : \sigma \rightarrow d \neq s \in Rul_G(\Theta(f_i, b_i), r)\}.$$

Then

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

The computational complexity of the optimization procedure is polynomial in the number of nodes in the directed acyclic graph $\Lambda_\gamma(T)$, which is at most the number of separable subtables of the table T.

Theorem 2. *For any node Θ of $\Lambda_\gamma(T)$ and for any row r of Θ, the set $Rul_{\Lambda_\gamma(T)}(\Theta, r)$ is equal to the set of all nonredundant γ-inhibitory rules for Θ and r over T.*

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

Let Θ be a nonterminal node of $\Lambda_\gamma(T)$ (with $N_{lcd}(\Theta) > \gamma$) and for each child of Θ the statement of the theorem holds. Let $r = (b_1, \ldots, b_n)$ be a row of Θ. Using Lemma 1 we obtain that $Rul_{\Lambda_\gamma(T)}(\Theta, r)$ contains only nonredundant γ-inhibitory rules for Θ and r over T.

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

To illustrate the algorithm presented above, we consider an example based on decision table T_0 (see Fig.1). In the example we set $\gamma = 1$, so during the construction of the graph $\Lambda_1(T_0)$ we stop the partitioning of a subtable Θ of T_0 when $N_{lcd}(\Theta) \leq 1$ (see Fig.1). We denote $G = \Lambda_1(T_0)$.

For each node Θ of the graph G and for each row r of Θ we describe a set $Rul_G(\Theta, r)$. We move from terminal nodes of G to the node T_0. Terminal nodes of the graph G are Θ_1, Θ_2, Θ_3, Θ_4, Θ_5, Θ_6. For these nodes we have:
$Rul_G(\Theta_1, r_1) = Rul_G(\Theta_1, r_2) = Rul_G(\Theta_1, r_6) = \{\rightarrow d \neq 2\}$,
$Rul_G(\Theta_2, r_3) = Rul_G(\Theta_2, r_4) = Rul_G(\Theta_2, r_5) = \{\rightarrow d \neq 1\}$,
$Rul_G(\Theta_3, r_1) = Rul_G(\Theta_3, r_4) = \{\rightarrow d \neq 3\}$,
$Rul_G(\Theta_4, r_2) = Rul_G(\Theta_4, r_3) = Rul_G(\Theta_4, r_5) = Rul_G(\Theta_4, r_6) = \{\rightarrow d \neq 1\}$,
$Rul_G(\Theta_5, r_1) = Rul_G(\Theta_5, r_4) = Rul_G(\Theta_5, r_5) = Rul_G(\Theta_5, r_6) = \{\rightarrow d \neq 1\}$,
$Rul_G(\Theta_6, r_2) = Rul_G(\Theta_6, r_3) = \{\rightarrow d \neq 3\}$.
Now we can describe the sets of rules corresponding to rows of T_0. This is a nonterminal node of G for which all children Θ_1, Θ_2, Θ_3, Θ_4, Θ_5 and Θ_6 are already treated. We have:
$Rul_G(T_0, r_1) = \{f_1 = 0 \rightarrow d \neq 2, f_2 = 0 \rightarrow d \neq 3, f_3 = 0 \rightarrow d \neq 1\}$,
$Rul_G(T_0, r_2) = \{f_1 = 0 \rightarrow d \neq 2, f_2 = 1 \rightarrow d \neq 1, f_3 = 1 \rightarrow d \neq 3\}$,
$Rul_G(T_0, r_3) = \{f_1 = 1 \rightarrow d \neq 1, f_2 = 1 \rightarrow d \neq 1, f_3 = 1 \rightarrow d \neq 3\}$,
$Rul_G(T_0, r_4) = \{f_1 = 1 \rightarrow d \neq 1, f_2 = 0 \rightarrow d \neq 3, f_3 = 0 \rightarrow d \neq 1\}$,
$Rul_G(T_0, r_5) = \{f_1 = 1 \rightarrow d \neq 1, f_2 = 1 \rightarrow d \neq 1, f_3 = 0 \rightarrow d \neq 1\}$,
$Rul_G(T_0, r_6) = \{f_1 = 0 \rightarrow d \neq 2, f_2 = 1 \rightarrow d \neq 1, f_3 = 0 \rightarrow d \neq 1\}$.

4. Procedure of Optimization Relative to Number of Misclassifications

Let $G = \Lambda_\gamma(T)$. We consider the procedure of optimization of the graph G relative to the number of misclassifications μ. For each node Θ in the graph G, this procedure corresponds to each row r of Θ the set $Rul^\mu_G(\Theta, r)$ of γ-inhibitory rules with the minimum number of misclassifications from $Rul_G(\Theta, r)$ and the number $Opt^\mu_G(\Theta, r)$ – the minimum number of misclassifications of a γ-inhibitory rule from $Rul_G(\Theta, r)$.

For each terminal node Θ of G and for each row r of Θ, the following equalities hold:

$$Rul^\mu_G(\Theta, r) = Rul_G(\Theta, r) = \{\rightarrow d \neq c\}.$$

where c is the least common decision for Θ, and $Opt^\mu_G(\Theta, r)$ is equal to the number of rows in Θ labeled with the decisions c.

Let Θ be a nonterminal node of G, 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 $f_i \in E_G(\Theta, r)$,

$$Rul_G(\Theta, r, f_i) = \{f_i = b_i \wedge \sigma \rightarrow d \neq s : \sigma \rightarrow d \neq s :\in Rul_G(\Theta(f_i, b_i), r)\}.$$

For $f_i \in E_G(\Theta, r)$, we denote by $Rul^\mu_G(\Theta, r, f_i)$ the set of all γ-inhibitory rules with the minimum number of misclassifications from $Rul_G(\Theta, r, f_i)$ and by $Opt^\mu_G(\Theta, r, f_i)$ we denote the minimum number of misclassifications of a γ-inhibitory rule from $Rul_G(\Theta, r, f_i)$.

One can show that

$$Rul^\mu_G(\Theta, r, f_i) = \{f_i = b_i \wedge \sigma \rightarrow d \neq s : \sigma \rightarrow d \neq s \in Rul^\mu_G(\Theta(f_i, b_i), r)\},$$

$$Opt^\mu_G(\Theta, r, f_i) = Opt^\mu_G(\Theta(f_i, b_i), r),$$

and $Opt^\mu_G(\Theta, r) = \min\{Opt^\mu_G(\Theta, r, f_i) : f_i \in E_G(\Theta, r)\} = \min\{Opt^\mu_G(\Theta(f_i, b_i), r) : f_i \in E_G(\Theta, r)\}$. It's easy to see also that

$$Rul^\mu_G(\Theta, r) = \bigcup_{f_i \in E_G(\Theta, r), Opt^\mu_G(\Theta(f_i, b_i), r) = Opt^\mu_G(\Theta, r)} Rul^\mu_G(\Theta, r, f_i).$$

To describe the procedure of optimization of the graph G relative to the number of misclassifications μ, we move from the terminal nodes of the graph G to the node T. We will correspond to each row r of each table Θ the

number $Opt^{\mu}_{G}(\Theta, r)$ which is the minimum number of misclassifications of a γ-inhibitory rule from $Rul_G(\Theta, r)$ and we will change the set $E_G(\Theta, r)$ attached to the row r in Θ if Θ is a nonterminal node of G. We denote the obtained graph by G^{μ}.

Let Θ be a terminal node of G and c be the least common decision for Θ. Then we correspond to each row r of Θ the number $Opt^{\mu}_{G}(\Theta, r)$ which is equal to the number of rows in Θ which are labeled with the decision c.

Let Θ be a nonterminal node of G and all children of Θ have already been treated. Let $r = (b_1, \ldots, b_n)$ be a row of Θ. We correspond the number $Opt^{\mu}_{G}(\Theta, r) = \min\{Opt^{\mu}_{G}(\Theta(f_i, b_i), r) : f_i \in E_G(\Theta, r)\}$ to the row r in the table Θ, and we set $E_{G^{\mu}}(\Theta, r) = \{f_i : f_i \in E_G(\Theta, r), Opt^{\mu}_{G}(\Theta(f_i, b_i), r) = Opt^{\mu}_{G}(\Theta, r)\}$. From the reasoning before the description of the procedure of optimization relative to the number of misclassifications (the beginning of Section 4) the next statement follows.

Theorem 3. *For each node Θ of the graph G^{μ} and for each row r of Θ, the set $Rul_{G^{\mu}}(\Theta, r)$ is equal to the set $Rul^{\mu}_{G}(\Theta, r)$ of all γ-inhibitory rules with the minimum number of misclassifications from the set $Rul_G(\Theta, r)$.*

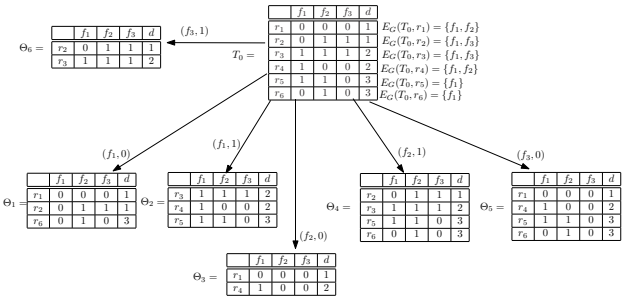

Fig. 2. Graph $G^{\mu} = \Lambda_1(T_0)^{\mu}$

Figure 2 presents the directed acyclic graph G^{μ} obtained from the graph G (see Fig. 1) by the procedure of optimization relative to the number of misclassifications. Using the graph G^{μ} we can describe for each row r_i, $i = 1, \ldots, 6$, of the table T_0 the set $Rul^{\mu}_{G}(T_0, r_i)$ of all nonredundant 1-inhibitory rules for T_0 and r_i over T_0 with the minimum number of misclassifications. We give also the value $Opt^{\mu}_{G}(T_0, r_i)$ which is equal to the minimum number of misclassifications of a nonredundant 1-inhibitory rule for T_0 and r_i over T_0. This value was obtained during the procedure of optimization of the graph G relative to the number of misclassifications. We have:
$Rul_G(T_0, r_1) = \{f_1 = 0 \to \neq 2, f_2 = 0 \to \neq 3\}$, $Opt^{\mu}_{G}(T_0, r_1) = 0$,
$Rul_G(T_0, r_2) = \{f_1 = 0 \to \neq 2, f_3 = 1 \to \neq 3\}$, $Opt^{\mu}_{G}(T_0, r_2) = 0$,
$Rul_G(T_0, r_3) = \{f_1 = 1 \to \neq 1, f_3 = 1 \to \neq 3\}$, $Opt^{\mu}_{G}(T_0, r_3) = 0$,
$Rul_G(T_0, r_4) = \{f_1 = 1 \to \neq 1, f_2 = 0 \to \neq 3\}$, $Opt^{\mu}_{G}(T_0, r_4) = 0$,
$Rul_G(T_0, r_5) = \{f_1 = 1 \to \neq 1\}$, $Opt^{\mu}_{G}(T_0, r_5) = 0$,
$Rul_G(T_0, r_6) = \{f_1 = 0 \to \neq 2\}$, $Opt^{\mu}_{G}(T_0, r_6) = 0$.

5. Experimental Results

For experiments we use decision tables from the UCI Machine Learning Repository [3]. We preprocess the decision tables by eliminating attributes which, each row, take unique value such as ID number, merging identical rows into a single row with the most common decision for the group of identical rows, and estimating missing values with the most common value of the corresponding attribute.

Let T be one of these decision tables. We consider for this table the value of $N_{lcd}(T)$ and values of γ from the set $\Gamma(T) = \{\lfloor N_{lcd}(T) \times 0.2 \rfloor, \lfloor N_{lcd}(T) \times 0.3 \rfloor, \lfloor N_{lcd}(T) \times 0.5 \rfloor, \lfloor N_{lcd}(T) \times 0.6 \rfloor\}$. These parameters can be found in Table 1, where (i) column "Rows" contains the number of rows, (ii) column "Attributes" contains number of conditional attributes, (iii) column "$N_{lcd}(T)$" contains the number of rows with the least common decision for T, (iv) column "$\gamma \in \Gamma(T)$" contains values from $\Gamma(T)$.

Table 2 presents the results of the procedure of optimization relative to the number of misclassifications of nonredundant γ-inhibitory rules. For each row r of T, we find the minimum number of misclassifications of

Table 1. Parameters of decision tables and values of γ

Decision table	Rows	Attributes	$N_{lcd}(T)$	$\gamma \in \Gamma(T)$			
				$\lfloor N_{lcd}(T) \times 0.2 \rfloor$	$\lfloor N_{lcd}(T) \times 0.3 \rfloor$	$\lfloor N_{lcd}(T) \times 0.5 \rfloor$	$\lfloor N_{lcd}(T) \times 0.6 \rfloor$
Balance-scale	625	4	49	9	14	24	29
Breast-cancer	266	9	76	15	22	38	45
Cars	1728	6	65	13	19	32	39
Hayes-roth-data	69	4	18	3	5	9	10
Shuttle-landing-control	15	6	6	1	1	3	3
Soybean-small	47	35	10	2	3	5	6
Zoo	59	16	4	0	1	2	2
Tic-tac-toe	959	9	332	66	99	166	199

Table 2. Minimum number of misclassifications of γ-inhibitory rules for $\gamma \in \Gamma(T)$

Decision Table	$\gamma \in \Gamma(T)$											
	$\lfloor N_{lcd}(T) \times 0.2 \rfloor$			$\lfloor N_{lcd}(T) \times 0.3 \rfloor$			$\lfloor N_{lcd}(T) \times 0.5 \rfloor$			$\lfloor N_{lcd}(T) \times 0.6 \rfloor$		
	min	avg	max	min	avg	max	min	avg	max	min	avg	max
Balance-scale	0	0.92	9	9	9.13	11	9	9.13	11	9	9.13	11
Breast-cancer	0	0.71	4	0	1.31	5	0	3.20	11	0	4.55	17
Cars	0	0	0	0	0	0	0	0.37	18	0	0.56	18
Hayes-roth-data	0	0.10	1	0	0.54	1	0	2.26	4	0	2.26	4
Shuttle-landing-control	0	0.07	1	0	0.07	1	0	0.20	2	0	0.20	2
Soybean-small	0	0	0	0	0	0	0	0	0	0	0	0
Zoo	0	0	0	0	0	0	0	0	0	0	0	0
Tic-tac-toe	5	10.52	18	13	16.59	34	48	66.28	101	48	66.28	101

a nonredundant γ-inhibitory rule for T and r over T. After that, we find for rows of T the minimum number of misclassifications of an inhibitory rule with minimum number of misclassifications (column "min"), the maximum number of misclassifications of such a rule (column "max"), and the average number of misclassifications of rules with minimum number of misclassifications – one for each row (column "avg").

The obtained results show that the number of misclassificaions for the constructed rules is often essentially less than a threshold γ. For example, the average number of misclassifications for constructed rules in case of $\gamma = \lfloor N_{lcd}(T) \times 0.6 \rfloor$ is three times less than γ for each of the considered tables. Experiments were done using software system Dagger [17] which is implemented in C++ and uses Pthreads and MPI libraries for managing threads and processes respectively.

6. Conclusions

We considered a dynamic programming approach for the representation of the set of nonredundant γ-inhibitory rules and optimization of these rules relative to the number of misclassifications. The experiments indicated the usefulness of the proposed approach. Further investigations will be devoted to the sequential optimization of γ-inhibitory rules relative to the length, coverage, and the number of misclassifications.

References

[1] F. Alsolami, I. Chikalov, M. Moshkov, B. Zielosko, Length and coverage of inhibitory decision rules, in: Computational Collective Intelligence. Technologies and Applications, Vol. 7654 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 325–334.

[2] F. Alsolami, I. Chikalov, M. Moshkov, B. Zielosko, Optimization of inhibitory decision rules relative to length and coverage, in: Rough Sets and Knowledge Technology, Vol. 7414 of Lecture Notes in Computer Science, Springer, 2012, pp. 149–154.

[3] A. Asuncion, D. J. Newman, UCI Machine Learning Repository (`http://www.ics.uci.edu/~mlearn/`, 2007).

[4] H. Nguyen, Approximate boolean reasoning: Foundations and applications in data mining, in: Transactions on Rough Sets V, Vol. 4100 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2006, pp. 334–506.

[5] Z. Pawlak, A. Skowron, Rough sets and boolean reasoning, Information Sciences 177 (1) (2007) 41–73.

[6] K. Dembczyński, W. Kotłowski, R. Słowiński, Ender: a statistical framework for boosting decision rules, Data Mining and Knowledge Discovery 21 (1) (2010) 52–90.

[7] N. Lavrač, J. Fürnkranz, D. Gamberger, Explicit feature construction and manipulation for covering rule learning algorithms, in: Advances in Machine Learning I, Vol. 262 of Studies in Computational Intelligence, Springer, 2010, pp. 121–146.

[8] S. Michalski, J. Pietrzykowski, iAQ: A program that discovers rules, AAAI-07 AI Video Competition (`http://videolectures.net/aaai07_michalski_iaq/`, 2007).

[9] M. Moshkov, B. Zielosko, Combinatorial Machine Learning, Vol. 360 of Studies in Computational Intelligence, Springer Berlin Heidelberg, 2011.
[10] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.
[11] Z. Suraj, Some remarks on extensions and restrictions of information systems, in: Rough Sets and Current Trends in Computing, Vol. 2005 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2001, pp. 204–211.
[12] A. Skowron, Z. Suraj, Rough sets and concurrency, Bulletin of the Polish Academy of Sciences 41(3) (1993) 237–254.
[13] P. Delimata, M. J. Moshkov, A. Skowron, Z. Suraj, Inhibitory Rules in Data Analysis: A Rough Set Approach, Vol. 163 of Studies in Computational Intelligence, Springer, 2009.
[14] P. Delimata, M. Moshkov, A. Skowron, Z. Suraj, Two families of classification algorithms, in: Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, Vol. 4482 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, pp. 297–304.
[15] P. Delimata, M. J. Moshkov, A. Skowron, Z. Suraj, Comparison of lazy classification algorithms based on deterministic and inhibitory decision rules, in: Rough Sets and Knowledge Technology, Vol. 5009 of Lecture Notes in Computer Science, Springer, 2008, pp. 55–62.
[16] P. Delimata, M. J. Moshkov, A. Skowron, Z. Suraj, Lazy classification algorithms based on deterministic and inhibitory decision rules, in: Information Processing and Management of Uncertainty in Knowledge-Based Systems, 2008, pp. 1773–1778.
[17] A. Alkhalid, T. Amin, I. Chikalov, S. Hussain, M. Moshkov, B. Zielosko, Dagger: A tool for analysis and optimization of decision trees and rules, in: 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, pp. 29–39.
[18] T. Amin, I. Chikalov, M. Moshkov, B. Zielosko, Optimization of approximate decision rules relative to number of misclassifications: Comparison of greedy and dynamic programming approaches, in: Knowledge Engineering, Machine Learning and Lattice Computing with Applications, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013.