Relationships Between Length and Coverage of Exact Decision Rules

Talha Amin[1], Igor Chikalov[1], Mikhail Moshkov[1], and Beata Zielosko[1,2]

[1] Mathematical and Computer Sciences & Engineering Division
King Abdullah University of Science and Technology
Thuwal 23955-6900, Saudi Arabia
{talha.amin,igor.chikalov,mikhail.moshkov,beata.zielosko}@kaust.edu.sa
[2] Institute of Computer Science, University of Silesia
39, Będzińska St., 41-200 Sosnowiec, Poland

Abstract. The paper describes a new tool for study relationships between length and coverage of exact decision rules. This tool is based on dynamic programming approach. We also present results of experiments with decision tables from UCI Machine Learning Repository.

Keywords: decision rules, dynamic programming, length, coverage

1 Introduction

Decision rules are widely used in applications connected with data mining, knowledge discovery and machine learning.

There are many different approaches to the design and analysis of decision rules: brute-force approach, genetic algorithms [25], simulated annealing [12], Boolean reasoning [18, 20, 24], ant colony optimization [13], algorithms based on decision tree construction [14, 17, 21], algorithms based on a sequential covering procedure [5, 8, 9], different kinds of greedy algorithms [16, 18], and dynamic programming [2–4].We can find many programs which allow data analysis using decision rules based on the approaches mentioned above, e.g., LERS [10], RSES [7], Rosetta [19], Weka [11], TRS library [23], and others.

We propose a new tool for analyzing relationships between the length and coverage of decision rules. This tool is based on a dynamic programming approach and is part of the software system Dagger [1] created in King Abdullah University of Science and Technology.

We study the dependence between length and coverage of exact irredundant decision rules. Such rules are constructed using a dynamic programming algorithm [3]. The length of a rule is the number of descriptors (expressions "attribute=value") on the left-hand side of the rule. The coverage of a rule, for a given decision table, is the number of rows in the table for which the rule is realizable (the left-hand side of the rule is true) and the decision from the right-hand side of the rule is equal to the decision attached to the row. The choice of length as a rule parameter is connected with the Minimum Description Length principle [22]. The rule coverage is important for discovering major patterns in the data.

For a given decision table T, row r of T, and a natural number m, we would like to find the value $F_{T,r}(m)$ which is the maximum coverage of a decision rule that is true for T, realizable for r, and whose length is at most m.

We created an algorithm that allows us to find the value $F_{T,r}(m)$, for all m such that m is at least the minimum length of the considered rules and at most the number of attributes in T. This algorithm is based on ideas of dynamic programming and requires the construction of a graph which nodes are subtables of the decision table T. We also study the reverse relationship which is described by the function $G_{T,r}$. The value of this function, for a given p, is equal to the minimum length among all irredundant decision rules for T and r which have a coverage of at least p. For experimentation, we used decision tables from UCI ML Repository [6].

The chapter consists of seven sections. Section 2 contains main notions and defines irredundant decision rules. In Sect. 3, we present an algorithm for constructing a directed acyclic graph whose nodes are subtables of a given decision table T. Section 4 is devoted to the consideration of an algorithm that constructs the function $F_{T,r}$ for a given decision table T and its row r. In Sect. 5, we describe how it is possible to find values of the function $G_{T,r}$ using values of the function $F_{T,r}$. Section 6 contains the results of our experiments, and Sect. 7 – a short conclusion.

2 Main Notions

In this section, we present definitions of notions corresponding to decision tables and decision 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.

We denote by $N(T)$ the number of rows in the 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 table obtained from T by the removal of some rows is called a *subtable* of the table T. Let T be nonempty and $f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\}$ and $a_1, \ldots, a_m$ be nonnegative integers. We denote by $T(f_{i_1}, a_1) \ldots (f_{i_m}, a_m)$ 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 $f_i \in E(T)$, we denote by $E(T, f_i)$ the set of values of the attribute f_i in T.

The expression

$$f_{i_1} = a_1 \wedge \ldots \wedge f_{i_m} = a_m \rightarrow d \quad (1)$$

is called a *decision rule over* T if $f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\}$, and $a_1, \ldots a_m, d$ are nonnegative integers. It is possible that $m = 0$. In this case (1) is equal to the rule

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

Let $r = (b_1, \ldots, b_n)$ be a row of T. We will say that the rule (1) is *realizable for* r if $a_1 = b_{i_1}, \ldots, a_m = b_{i_m}$. 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 $T' = T(f_{i_1}, a_1) \ldots (f_{i_m}, a_m)$ 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 a rule is true for T and realizable for r, we will say that this is a *decision rule for* T *and* r.

2.1 Irredundant Decision Rules

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

(i) $f_{i_1} \in E(T)$, and if $m > 1$ then $f_{i_j} \in E(T(f_{i_1}, a_1) \ldots (f_{i_{j-1}}, a_{j-1}))$ for $j = 2, \ldots, m$;

(ii) if $m = 1$ then the table T is nondegenerate, and if $m > 1$ then the table $T(f_{i_1}, a_1) \ldots (f_{i_{m-1}}, a_{m-1})$ is nondegenerate.

If $m = 0$ then the rule (2) is an *irredundant* decision rule for T and r if (2) is a decision rule for T and r, i.e., if T is degenerate and each row of T is labeled with the decision d.

Lemma 1. *[3] Let* T *be a nondegenerate decision table,* $f_{i_1} \in E(T)$*,* $a_1 \in E(T, f_{i_1})$*, and* r *be a row of the table* $T' = T(f_{i_1}, a_1)$*. Then the rule (1) with* $m \geq 1$ *is an irredundant decision rule for* T *and* r *if and only if the rule*

$$f_{i_2} = a_2 \wedge \ldots \wedge f_{i_m} = a_m \rightarrow d \quad (3)$$

is an irredundant decision rule for T' *and* r *(if* $m = 1$ *then (3) is equal to* $\rightarrow d$*).*

Let τ be a decision rule over T and τ be equal to (1).

The number m of descriptors (pairs "attribute=value") on the left-hand side of τ is called the *length* of the rule and is denoted by $l(\tau)$. The length of decision rule (2) is equal to 0.

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 which are labeled with the decision d. If τ is true for T (we consider now this case) then $c(\tau) = N(T(f_{i_1}, a_1) \ldots (f_{i_m}, a_m))$.

Proposition 1. *[3] Let T be a nonempty decision table, r be a row of T and τ be a decision rule for T and r which is not an irredundant decision rule for T and r. Then by removal of some descriptors from the left-hand side of τ we can obtain an irredundant decision rule $irr(\tau)$ for T and r such that $l(irr(\tau)) \leq l(\tau)$ and $c(irr(\tau)) \geq c(\tau)$.*

From Proposition 1 it follows that instead of arbitrary decision rules for T and r we can consider only irredundant decision rules for T and r.

We will say that an irredundant decision rule for T and r is *totally optimal relative to the length and coverage* if it has the minimum length and the maximum coverage among all irredundant decision rules for T and r.

3 Directed Acyclic Graph $\Delta(T)$

Now, we consider an algorithm that constructs a directed acyclic graph $\Delta(T)$. Based on this graph we can describe the set of irredundant decision rules for T and for each row r of T. We can also study relationships between the length and coverage of such rules. 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 *.

Let us assume that the algorithm has already performed p steps. We now describe 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, which 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 a bundle of edges from the node Θ. Let $E(\Theta, f_i) = \{b_1, \ldots, b_t\}$. Then draw t edges from Θ and label these edges with pairs $(f_i, b_1), \ldots, (f_i, b_t)$ respectively. These edges enter to nodes $\Theta(f_i, b_1), \ldots, \Theta(f_i, b_t)$. If some of nodes $\Theta(f_i, b_1), \ldots, \Theta(f_i, b_t)$ are absent in the graph then add these nodes to the graph. 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.

Now, for each node Θ of $\Delta(T)$ and for each row r of Θ we describe a set of decision rules $Rul(\Theta, r)$. Let Θ be a terminal node of $\Delta(T)$, i.e., Θ is a degenerate table in which each row is labeled with the same decision d. Then

$$Rul(\Theta, r) = \{\rightarrow d\}.$$

Let Θ now be a nonterminal node of $\Delta(T)$ such that for each child Θ' of Θ and for each row r' of Θ' the set of rules $Rul(\Theta', r')$ has already been defined. Let $r = (b_1, \ldots, b_n)$ be a row of Θ labeled with a decision d. For any $f_i \in E(\Theta)$, we define the set of rules $Rul(\Theta, r, f_i)$ as follows:

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

Then

$$Rul(\Theta, r) = \bigcup_{f_i \in E(\Theta)} Rul(\Theta, r, f_i).$$

Theorem 1. *[3] For any node Θ of $\Delta(T)$ and for any row r of Θ, the set $Rul(\Theta, r)$ is equal to the set of all irredundant decision rules for Θ and r.*

A detailed example for the construction of the directed acyclic graph and the description of the set of all irredundant decision rules for each row of a given decision table T can be found in [3].

It is possible to show (see analysis of similar algorithms in [17], page 64) that the time complexities of algorithms which construct the graph $\Delta(T)$ and make optimization of decision rules relative to length or coverage are bounded above by polynomials on the number of separable subtables of T, and the number of attributes in T. In [15] it was shown that the number of separable subtables for decision tables with attributes from a restricted infinite information systems is bounded from above by a polynomial on the number of attributes in the table. Examples of restricted infinite information system were considered, in particular, in [17].

4 Relationship Between Coverage and Length

Let T be a decision table with n columns labeled with attributes $f_1, \ldots, f_n$, and $r = (b_1, \ldots, b_n)$ be a row of T. Let Θ be a node of the graph $\Delta(T)$ containing the row r.

From Theorem 1 it follows that the set $Rul(\Theta, r)$ is equal to the set of all irredundant decision rules for Θ and r. We denote by $l_{min}(\Theta, r)$ the minimum length of a decision rule from $Rul(\Theta, r)$. We denote

$$B_{\Theta,r} = \{l_{min}(\Theta, r), l_{min}(\Theta, r) + 1, \ldots, n\}.$$

Now we define a function $F_{\Theta,r} : B_{\Theta,r} \to \mathbf{N}$, where $\mathbf{N}$ is the set of natural numbers. For any $m \in B_{\Theta,r}$, we have

$$F_{\Theta,r}(m) = \max\{c(\tau) : \tau \in Rul(\Theta, r), l(\tau) \leq m\}.$$

This function describes the relationship between coverage and length of decision rules: $F_{\Theta,r}(m)$ is equal to the maximum coverage among all irredundant decision rules for Θ and r whose length is at most m.

The function $F_{\Theta,r}$ can be represented by the tuple $(F_{\Theta,r}(t), F_{\Theta,r}(t{+}1), \ldots, n)$, where $t = l_{min}(\Theta, r)$.

We now describe a procedure that allows us to find (describe) function $F_{\Theta,r}$ for each node Θ of the graph $\Delta(T)$ that contains the row r.

Let Θ be a degenerate table that contains the row r. All rows of Θ are labeled with the same decision d. We know that there is exactly one irredundant decision rule $\to d$ for Θ and r. Therefore, $l_{min}(\Theta, r) = 0$ and $F_{\Theta,r}(m) = N(\Theta)$ for any $m \in B_{\Theta,r}$.

Now, let Θ be a nondegenerate table (node) from $\Delta(T)$ containing the row r. We know that

$$Rul(\Theta, r) = \bigcup_{f_i \in E(\Theta)} Rul(\Theta, r, f_i),$$

where $Rul(\Theta, r, f_i) = \{f_i = b_i \wedge \alpha \to d : \alpha \to d \in Rul(\Theta(f_i, b_i), r)\}$. From here it follows that

$$l_{min}(\Theta, r) = \min\{l_{min}(\Theta(f_i, b_i)) : f_i \in E(\Theta)\} + 1$$

and, for any $m \in B_{\Theta,r}$,

$$F_{\Theta,r}(m) = \max\{F_{\Theta(f_i,b_i),r}(m-1) : f_i \in E(\Theta), m-1 \geq l_{min}(\Theta(f_i, b_i), r)\}.$$

We use here the fact that the length of the rule $f_i = b_i \wedge \alpha \to d$ is equal to the length of rule $\alpha \to d$ plus 1, and the coverage of rule $f_i = b_i \wedge \alpha \to d$ for the table Θ is equal to the coverage of rule $\alpha \to d$ for the table $\Theta(f_i, b_i)$.

Let us consider an example with a decision table T_0 on the top of Fig. 1. We study the relationship between coverage and length of irredundant decision

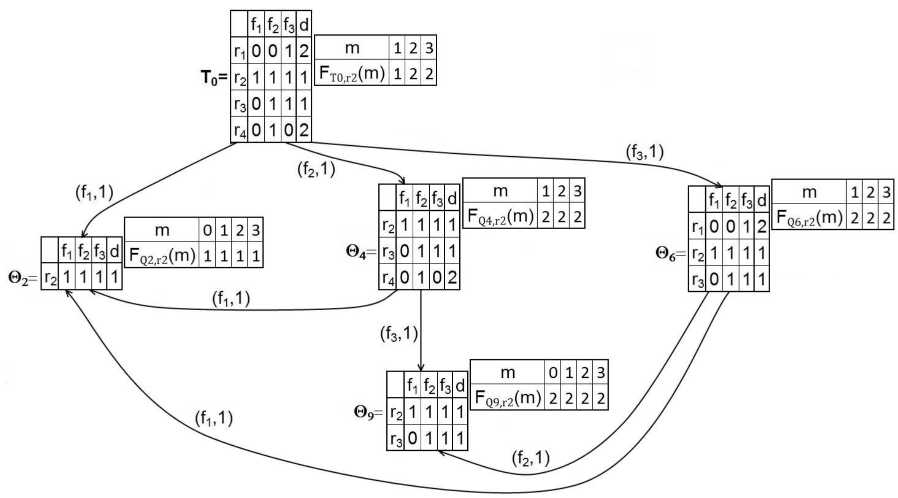

Fig. 1. Relationships between coverage and length of rules for r_2

rules for the row r_2 of the table T_0. For clarity, the example contains only a part of the directed acyclic graph $\Delta(T_0)$ constructed for the decision table T_0; we omitted separable subtables which do not contain the second row of T_0. We attached to each subtable Θ of this graph a two-row table that describes the function F_{Θ,r_2}: the first row contains values $l_{min}(\Theta, r_2), \ldots, 3$, and the second row contains values $F_{\Theta,r_2}(l_{min}(\Theta, r_2)), \ldots, F_{\Theta,r_2}(3)$ respectively. The relationship between coverage and length of irredundant decision rules for the row r_2

is described by the functions $F_{\Theta_2,r_2}, F_{\Theta_9,r_2}, F_{\Theta_4,r_2}, F_{\Theta_6,r_2}$ and F_{T_0,r_2} (see corresponding tables in Fig. 1).

5 Relationships Between Length and Coverage

In this section, we study a reversed relationship, i.e., the dependency between length and coverage of irredundant decision rules.

Let T be a decision table with n columns labeled with attributes $f_1, \ldots, f_n$ and $r = (b_1, \ldots, b_n)$ be a row of T. We know (see Theorem 1) that $Rul(T, r)$ is the set of all irredundant decision rules for T and r. We denote by $c_{max}(T, r)$ the maximum coverage of a decision rule from $Rul(T, r)$. We denote

$$C_{T,r} = \{0, 1, \ldots, c_{max}(T, r)\}.$$

We now define a function $G_{T,r} : C_{T,r} \to Z^+$, where Z^+ is the set of nonnegative integers. For any $p \in C_{T,r}$, we have

$$G_{T,r}(p) = \min\{l(\tau) : \tau \in Rul(T, r), c(\tau) \geq p\}.$$

The value $G_{T,r}(p)$ is equal to the minimum length among all irredundant decision rules for T and r for which the coverage is at least p. To find this value we can use values of the function $F_{T,r}$. Let us show that

$$G_{T,r}(p) = \min\{m : m \in \{l_{min}(T, r), \ldots, n\}, F_{T,r}(m) \geq p\}. \quad (4)$$

We denote

$$m_0 = \min\{m : m \in \{l_{min}(T, r), \ldots, n\}, F_{T,r}(m) \geq p\}.$$

It is clear that there exists a rule $\tau \in Rul(T, r)$ such that $l(\tau) \leq m_0$ and $c(\tau) \geq p$, and there is no rule $\rho \in Rul(T, r)$ such that $l(\rho) < m_0$ and $c(\rho) \geq p$. From here it follows that $G_{T,r}(p) = m_0$.

We should add that $c_{max}(T, r) = F_{T,r}(n)$. It is clear that $F_{T,r}$ is a nondecreasing function. So, to find the value of m_0 we can use binary search which requires $O(\log n)$ comparisons.

6 Experimental Results

In this section, we present experimental results for decision tables from UCI Machine Learning Repository [6] and we show plots depicting relationships between length and coverage of irredundant decision rules.

On the left-hand side of Fig. 2 we can see the function $F_{cars,row488}$ (relationship between coverage and length), and on the right-hand side – the function $G_{cars,row488}$ (relationship between length and coverage). Both functions are constant, which means that for row 488 of decision table "cars" there exists a totally optimal rule relative to the length and coverage, i.e., rule with minimum (among

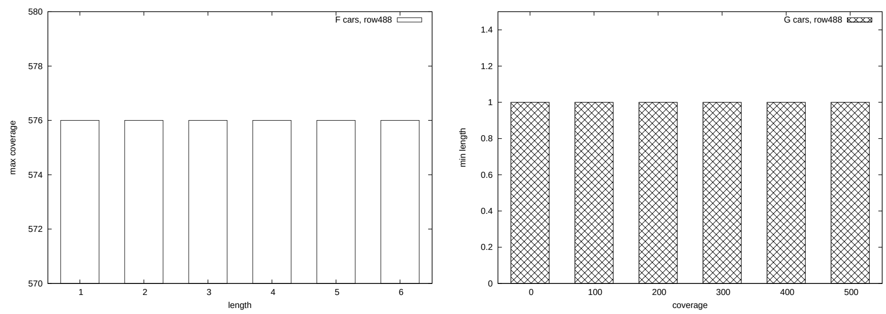

Fig. 2. Relationships between length and coverage of irredundant decision rules for the row 488 of decision table "cars"

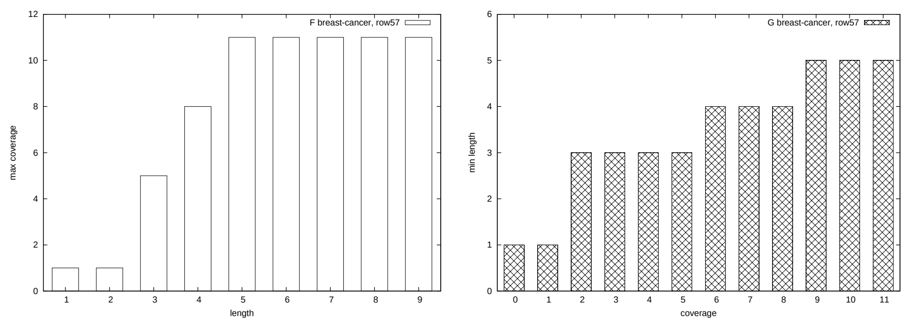

Fig. 3. Relationships between length and coverage of irredundant decision rules for the row 57 of decision table "breast-cancer"

all irredundant rules) length equal to 1 and maximum (among all irredundant rules) coverage equal to 576.

Functions presented in Fig. 3, $F_{breast-cancer,row57}$ and $G_{breast-cancer,row57}$ are increasing, which means that for row 57 of decision table "breast-cancer" there are no totally optimal rules relative to the length and coverage.

Table 1 presents, for a given decision table T, the number of rows r such that there exists a totally optimal irredundant decision rule for T and r.

We denote by $Row(T)$ the set of rows from T. We have $|Row(T)| = N(T)$. Let S be a system of irredundant decision rules for T where, for each row $r \in Row(T)$, we have a decision rule $rule$ which is realizable for r and true for T. The number of rules is equal to the number of rows in T. By $P(T)$ we denote the set of such systems.

By $l_{max}(S)$ we denote the maximum length of rules from S. By $c_{avg}(S)$ we denote the average coverage of rules from S,

$$c_{avg}(S) = \frac{\sum_{rule \in S} c(rule)}{|Row(T)|}.$$

Table 1. Existence of totally optimal rules

Decision table	Number of rows	Number of rows with tot. opt. rules
Adult-stretch	16	16
Agaricus-lepiota	8124	612
Balance-scale	625	625
Breast-cancer	266	133
Cars	1728	1728
Flags	193	53
Hayes-roth-data	69	69
Hause-votes-84	279	101
Lymphography	148	52
Nursery	12960	12960
Shuttle-landing-control	15	13
Soybean-small	47	37
Spect-test	169	108
Teeth	23	23
Tic-tac-toe	958	942
Zoo	59	44

Let $B_T = \{l_{min}(T), \ldots, n\}$, where n is the number of conditional attributes in T and

$$l_{min}(T) = \max\{l_{min}(T, r) : r \in Row(T)\}.$$

Our aim is to find for each $m \in B_T$ the value

$$F_T^{avg}(m) = \max\{c_{avg}(S) : S \in P(T), l_{max}(S) \leq m\}.$$

The value of F_T^{avg} can be calculated in the following way

$$F_T^{avg}(m) = \frac{\sum_{r \in Row(T)} F_{T,r}(m)}{|Row(T)|}.$$

So, $F_T^{avg}(m)$ is equal to the maximum value of the average coverage of rules among all systems $S \in P(T)$, where the length of each rule is at most m.

By $c_{min}(S)$ we denote the minimum coverage of rules from S. By $l_{avg}(S)$ we denote the average length of rules from S,

$$l_{avg}(S) = \frac{\sum_{rule \in S} l(rule)}{|Row(T)|}.$$

Let $C_T = \{0, 1, \ldots, c_{max}(T)\}$, where

$$c_{max}(T) = \min\{c_{max}(T, r) : r \in Row(T)\}.$$

Our aim is to find for each $p \in C_T$ the value

$$G_T^{avg}(p) = \min\{l_{avg}(S) : S \in P(T), c_{min}(S) \geq p\}.$$

The value of G_T^{avg} can be calculated in the following way

$$G_T^{avg}(p) = \frac{\sum_{r \in Row(T)} G_{T,r}(p)}{|Row(T)|}.$$

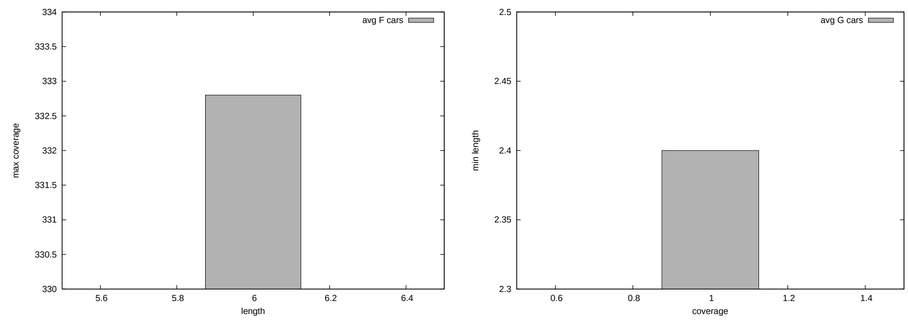

Fig. 4. Relationships between length and coverage of irredundant decision rules for decision table "cars"

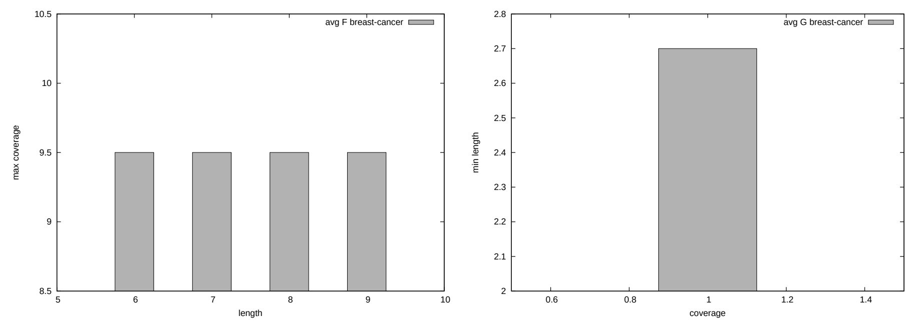

Fig. 5. Relationships between length and coverage of irredundant decision rules for decision table "breast-cancer"

So, $G_T^{avg}(p)$ is equal to the minimum value of the average length of rules among systems $S \in P(T)$, where the coverage of each rule is at least p.

Functions F_T^{avg} and G_T^{avg} for the decision table "cars" are presented in Fig. 4, and for the decision table "breast-cancer" – in Fig. 5.

The study of relationships between length and coverage of irredundant decision rules can be considered as a tool that supports designing of classifiers. To predict the value of a decision attribute for a new object, we can have a classifier use only totally optimal rules, rules with the maximum coverage, or with the minimum length. We can study the accuracy of classifiers, and based on a tool for relationships, we can try to find associations between length and coverage of rules which give the best result of classification. Besides, short rules which cover many objects can be useful in knowledge representation. In this case, rules with smaller number of descriptors are more understandable.

7 Conclusions

In this paper, we considered relationships between length and coverage of irredundant decision rules. Such a study can be useful from the point of view of knowledge representation. It can also give useful information for the construction of rule based classifiers.

References

1. Alkhalid, A., Amin, T., Chikalov, I., Hussain, S., Moshkov, M., Zielosko, B.: 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 (2011) 29–39
2. 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, September 28–30, Pułtusk, Poland, Białystok University of Technology (2011) 10–16
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. Springer (2012) to appear.
4. Amin, T., Chikalov, I., Moshkov, M., Zielosko, B.: Dynamic programming approach for partial decision rule optimization. Fundam. Inform. (2012) to appear.
5. An, A., Cercone, N.: ELEM2: a learning system for more accurate classifications. In: Proceedings of the 12th Biennial Conference of the Canadian Society for Computational Studies of Intelligence on Advances in Artificial Intelligence, London, UK, Springer-Verlag (1998) 426–441
6. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository (`http://www.ics.uci.edu/~mlearn/`, 2007)
7. Bazan, J.G., Szczuka, M.S., Wojna, A., Wojnarski, M.: On the evolution of rough set exploration system. In Tsumoto, S., Słowinski, R., Komorowski, H.J., Grzymała-Busse, J.W., eds.: RSCTC 2004. Volume 3066 of LNCS. Springer (2004) 592–601
8. Błaszczyński, J., Słowiński, R., Szeląg, M.: Sequential covering rule induction algorithm for variable consistency rough set approaches. Inf. Sci. **181**(5) (2011) 987–1002
9. Clark, P., Niblett, T.: The CN2 induction algorithm. Mach. Learn. **3**(4) (1989) 261–283
10. Grzymała-Busse, J.W.: LERS – a system for learning from examples based on rough sets. In Słowiński, R., ed.: Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory. Kluwer Academic Publishers (1992) 3–18
11. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. SIGKDD Explorations **11**(1) (2009) 10–18
12. Jensen, R., Shen, Q.: Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches. IEEE Trans. Knowl. Data Eng. **16**(12) (2004) 1457–1471

13. Liu, B., Abbass, H.A., McKay, B.: Classification rule discovery with ant colony optimization. In: IAT 2003. IEEE Computer Society (2003) 83–88
14. Michalski, S., Pietrzykowski, J.: iAQ: A program that discovers rules. AAAI-07 AI Video Competition (`http://videolectures.net/aaai07_michalski_iaq/`, 2007)
15. Moshkov, M., Chikalov, I.: On algorithm for constructing of decision trees with minimal depth. Fundam. Inform. **41**(3) (2000) 295–299
16. Moshkov, M., Piliszczuk, M., Zielosko, B.: Partial Covers, Reducts and Decision Rules in Rough Sets – Theory and Applications. Volume 145 of Studies in Computational Intelligence. Springer, Heidelberg (2008)
17. Moshkov, M., Zielosko, B.: Combinatorial Machine Learning – A Rough Set Approach. Volume 360 of Studies in Computational Intelligence. Springer, Heidelberg (2011)
18. Nguyen, H.S.: Approximate boolean reasoning: foundations and applications in data mining. In Peters, J.F., Skowron, A., eds.: T. Rough Sets. Volume 4100 of LNCS. Springer (2006) 334–506
19. Øhrn, A., Komorowski, J., Skowron, A., Synak, P.: The design and implementation of a knowledge discovery toolkit based on rough sets: The ROSETTA system. In: Rough Sets in Knowledge Discovery 1: Methodology and Applications. Volume 18 of Studies in Fuzziness and Soft Computing. Physica-Verlag (1998) 376–399
20. Pawlak, Z., Skowron, A.: Rough sets and boolean reasoning. Inf. Sci. **177**(1) (2007) 41–73
21. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc. (1993)
22. Rissanen, J.: Modeling by shortest data description. Automatica **14**(5) (1978) 465–471
23. Sikora, M.: Decision rule-based data models using TRS and NetTRS – methods and algorithms. T. Rough Sets **11** (2010) 130–160
24. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In Słowinski, R., ed.: Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory. Kluwer Academic Publishers, Dordrecht (1992) 331–362
25. Ślęzak, D., Wróblewski, J.: Order based genetic algorithms for the search of approximate entropy reducts. In Wang, G., Liu, Q., Yao, Y., Skowron, A., eds.: RSFDGrC 2003. Volume 2639 of LNCS. Springer (2003) 308–311