Fundamenta Informaticae XX (2018) 1–14 1
DOI 10.3233/FI-2016-0000
IOS Press

Comparison of Heuristics for Optimization of Association Rules

Fawaz Alsolami, Talha Amin, Mikhail Moshkov
Computer, Electrical and Mathematical Sciences and Engineering Division
King Abdullah University of Science and Technology
Thuwal 23955-6900, Saudi Arabia
{fawaz.alsolami, talha.amin, mikhail.moshkov}@kaust.edu.sa

Beata Zielosko, Krzysztof Żabiński
Institute of Computer Science, University of Silesia in Katowice
39, Będzińska St., 41-200 Sosnowiec, Poland
beata.zielosko@us.edu.pl, krzysztof.kamil.zabinski@gmail.com

Abstract. In this paper, seven greedy heuristics for construction of association rules are compared from the point of view of the length and coverage of constructed rules. The obtained rules are compared also with optimal ones constructed by dynamic programming algorithms. The average relative difference between length of rules constructed by the best heuristic and minimum length of rules is at most 4%. The same situation is with coverage.

Keywords: Greedy heuristics, association rules, decision rules, dynamic programming, rough sets

1. Introduction

Association rule mining is one of the important fields of data mining and knowledge discovery. It aims to extract interesting correlations, associations, or frequent patterns in a form of rules among

Address for correspondence: Institute of Computer Science, University of Silesia in Katowice
39, Będzińska St., 41-200 Sosnowiec, Poland
beata.zielosko@us.edu.pl

sets of items in data sets. There are many algorithms for construction of association rules. One of the most popular algorithms is Apriori algorithm based on frequent itemsets [1]. In the literature, many other approaches were proposed. Some of them are based on modifications of Apriori algorithm, e.g., partitioning the data [2], hash based technique [3]. There are also algorithms in the framework of GUHA method [4], algorithms that use vertical data format [5], algorithms that use frequent pattern growth approach [6] and others [7].

The most popular measures for mining association rules are support and confidence [8, 9, 10]. In this work, length and coverage are considered as rule evaluation measures. They are important from the point of view of knowledge representation. The choice of length is connected with the Minimum Description Length Principle [11]. Shorter rules are better from the point of view of understanding and interpreting by experts. Rules with relatively large coverage allow us to discover major patterns in the data.

Greedy algorithms for construction of association rules are studied here since the problems of construction of rules with minimum length or maximum coverage are NP-hard [12, 13, 14]. The most part of approaches, with the exception of brute-force, Apriori algorithm, GUHA method, and extensions of dynamic programming, cannot guarantee the construction of optimal rules (i.e., rules with minimum length or maximum coverage). In [13], it was shown based on results of U. Feige [15] that some greedy algorithm is close to the best polynomial approximate algorithms for minimization of association rule length, under reasonable assumptions on the class NP. To the best of our knowledge, there are no similar results for coverage.

Usually, the applications of rough set theory to the construction of rules for knowledge representation or classification tasks deal with decision tables [16]. Such table contains an attribute designated as a decision attribute and it appears in a rule's consequence. However, associative mechanism of rule constructions assume all attributes can occur as premises or consequences of rules [14]. In this paper, a special kind of associative mechanism of rule constructions is studied, i.e., they are closely connected to decision rules construction. Approach when single attribute is used as the attribute on the right-hand side of an association rule was considered in previous works, inter alia, [10, 14, 17, 18, 19]. Borgelt [20] also implemented Apriori algorithm that constructs association rules with a single item in the consequent (see `http://www.borgelt.net/doc/apriori/apriori.html#conseq`). Similar approach was proposed in [13, 21] where one greedy algorithm for length minimization of association rules was investigated.

In this paper, we extend our previous work [22] on studying greedy heuristics for construction of association rules and comparing them from the point of view of the length and coverage of constructed rules by considering seven heuristics instead of five heuristics. We also present how close the output of each greedy algorithm on average to optimal one constructed by dynamic programming algorithms for considered data sets. The experimental results show that the average relative difference between length of rules constructed by the best heuristic and minimum length of rules is at most 4%. Interestingly, the same situation is with coverage. The comparison of time and number of unique rules constructed by considered seven heuristics and Apropri algorithm is also presented.

The paper consists of five sections. Section 2 contains main notions. In Section 3, we discuss seven greedy heuristics. Section 4 contains experimental results for decision tables from UCI Machine Learning Repository. Finally, Section 5 concludes the paper.

2. Main Notions

An *information system* I is a table with $n+1$ columns labeled with attributes $f_1, \ldots, f_{n+1}$. Rows of this table are filled by nonnegative integers which are interpreted as values of attributes. Formally, information system I is defined as a pair $I = (U, A)$, where $U = \{r_1, \ldots, r_m\}$ is nonempty, finite set of objects (rows), $A = \{f_1, \ldots, f_{n+1}\}$ is nonempty, finite set of attributes, i.e., $f : U \to V_f$ for any $f \in A$ where V_f is the set of values of attribute f, called the domain of f [16]. Figure 1 represents an example of information system.

$J =$

	f_1	f_2	f_3
r_1	1	1	1
r_2	2	1	1
r_3	2	0	0

Figure 1. Information system J

An association rule for I is a rule of the kind

$$(f_{i_1} = a_1) \wedge \ldots \wedge (f_{i_m} = a_m) \to f_j = a,$$

where $f_j \in \{f_1, \ldots, f_{n+1}\}$, $f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_{n+1}\} \setminus \{f_j\}$, and a,a_1,…,a_m are nonnegative integers.

The notion of an association rule for I is based on the notions of a decision table and decision rule. We consider two kinds of decision tables: with many-valued decisions and with single-valued decisions.

A *decision table with many-valued decisions* T is a rectangular table with n columns labeled with (conditional) attributes $f_1, \ldots, f_n$. Rows of this table are pairwise different and are filled by nonnegative integers which are interpreted as values of conditional attributes. Each row r is labeled with a finite nonempty set $D(r)$ of nonnegative integers which are interpreted as decisions (values of a decision attribute). For a given row r of T, it is necessary to find a decision from the set $D(r)$.

A *decision table with single-valued decisions* T is a rectangular table with n columns labeled with (conditional) attributes $f_1, \ldots, f_n$. Rows of this table are pairwise different and are filled by nonnegative integers which are interpreted as values of conditional attributes. Each row r is labeled with a nonnegative integer $d(r)$ which is interpreted as a decision (value of a decision attribute). For a given row r of T, it is necessary to find the decision $d(r)$. Decision tables with single-valued decisions can be considered as a special kind of decision tables with many-valued decisions in which $D(r) = \{d(r)\}$ for each row r.

For each attribute $f_i \in \{f_1, \ldots, f_{n+1}\}$, the information system I is transformed into a table I_{f_i}. The column f_i is removed from I and a table with n columns labeled with attributes $f_1, \ldots, f_{i-1}$, $f_{i+1}, \ldots, f_{n+1}$ is obtained. Values of the attribute f_i are attached to the rows of the obtained table I_{f_i} as decisions.

The table I_{f_i} can contain equal rows. We transform this table into two decision tables – with many-valued and single-valued decisions. A decision table $I_{f_i}^{m-v}$ with many-valued decisions is obtained from the table I_{f_i} by replacing each group of equal rows with a single row from the group with the set of decisions attached to all rows from the group. A decision table $I_{f_i}^{s-v}$ with single-valued decisions is obtained from the table I_{f_i} by replacing each group of equal rows with a single row from the group with the most common decision for this group.

$J_{f_1}^{m-v} =$

	f_2	f_3	
r_1	1	1	{1,2}
r_2	0	0	{2}

$J_{f_2}^{m-v} =$

	f_1	f_3	
r_1	1	1	{1}
r_2	2	1	{1}
r_3	2	0	{0}

$J_{f_3}^{m-v} =$

	f_1	f_2	
r_1	1	1	{1}
r_2	2	1	{1}
r_3	2	0	{0}

Figure 2. The set of decision tables $\Phi^{m-v}(J)$ obtained from the information system J

The set $\{I_{f_1}^{m-v}, \ldots, I_{f_{n+1}}^{m-v}\}$ of decision tables with many-valued decisions obtained from the information system I is denoted by $\Phi^{m-v}(I)$. The set $\Phi^{m-v}(J)$ for the information system J (see Fig. 1) is presented in Fig. 2. The set $\{I_{f_1}^{s-v}, \ldots, I_{f_{n+1}}^{s-v}\}$ of decision tables with single-valued decisions obtained from the information system I is denoted by $\Phi^{s-v}(I)$. The set $\Phi^{s-v}(J)$ is presented in Fig. 3. Since decision tables with single-valued decisions are a special case of decision tables with many-valued decisions, we consider the notion of decision rule for tables with many-valued decisions.

$J_{f_1}^{s-v} =$

	f_2	f_3	
r_1	1	1	1
r_2	0	0	2

$J_{f_2}^{s-v} =$

	f_1	f_3	
r_1	1	1	1
r_2	2	1	1
r_3	2	0	0

$J_{f_3}^{s-v} =$

	f_1	f_2	
r_1	1	1	1
r_2	2	1	1
r_3	2	0	0

Figure 3. The set of decision tables $\Phi^{s-v}(J)$ obtained from the information system J

Let $T \in \Phi^{m-v}(I)$. For simplicity, let $T = I_{f_{n+1}}^{m-v}$. The attribute f_{n+1} will be considered as a decision attribute of the table T. We denote by $N(T)$ the number of rows in table T. For a decision a, denote $N(T,a)$ the number of rows r of T such that $a \in D(r)$, and $M(T,a) = N(T) - N(T,a)$. A decision a is a *common* decision of T if $a \in D(r)$ for any row r of T. We denote by $mcd(T)$ the *most common* decision for T which is the minimum decision a such that $N(T,a)$ has maximum value. For example, for a decision table $J_{f_1}^{m-v}$ depicted in Fig. 2, we have the following: $N(T) = 2$, $mcd(J_{f_1}^{m-v}) = 2$, for a decision 2: $N(J_{f_1}^{m-v}, 2) = 2$, $M(J_{f_1}^{m-v}, 2) = 0$.

We denote by $E(T)$ the set of conditional attributes of T which are not constant on T. A table obtained from T by removal some rows is called a subtable of T. We denote by $T(f_{i_1}, a_1), \ldots, (f_{i_m}, a_m)$ a *subtable* of T which consists of rows that at the intersection with columns $f_{i_1}, \ldots, f_{i_m}$ have values $a_1, \ldots, a_m$. Fig. 4 presents a subtable $J_{f_2}^{m-v}(f_1, 2)$ of the table $J_{f_2}^{m-v}$ depicted in Fig. 2.

The expression

$$(f_{i_1} = a_1) \wedge \ldots \wedge (f_{i_m} = a_m) \rightarrow f_{n+1} = a$$

is called a *decision rule over* T if $f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\}, a_1, \ldots, a_m$ are the values of the corresponding attributes, and a is a decision. We correspond to the considered rule the subtable

	f_1	f_3	
r_2	2	1	$\{1\}$
r_3	2	0	$\{0\}$

Figure 4. Subtable $J_{f_2}^{m-v}(f_1, 2)$

$T' = T(f_{i_1}, a_1), \ldots, (f_{i_m}, a_m)$ of the table T. This rule is called *realizable for a row* r of T if r belongs to T'. This rule is called *true* for T if a is a common decision of T'. We say that the considered rule is a *rule for* T *and* r, if this rule is true for T and realizable for r. The number m is called the *length* of the rule. The *coverage* of the rule is the number of rows r from T' for which $a \in D(r)$. If the considered rule is a rule for T and r then its coverage is equal to $N(T')$.

Decision rules which are true for decision tables from $\Phi^{m-v}(I)$ can be considered as association rules (modification for many-valued decision model) that are true for the information system I. Decision rules which are true for decision tables from $\Phi^{s-v}(I)$ can be considered as association rules (modification for single-valued decision model) that are true for the information system I.

3. Greedy Heuristics

We consider the work of seven greedy heuristics on an example of the table $T = I_{f_{n+1}}^{m-v}$.

3.1. Heuristics with Fixed Decision

Let $r = (b_1, \ldots, b_n)$ be a row of T and a be a decision from $D(r)$. We consider five heuristics with fixed decision. A heuristic H constructs a decision rule for T, r, and a. This heuristic starts with a rule whose left-hand side is empty $\rightarrow f_{n+1} = a$, and then sequentially adds conditions to the left-hand side of this rule. Let during the work of the heuristic H, we already constructed the following rule:

$$(f_{i_1} = b_{i_1}) \wedge \ldots \wedge (f_{i_m} = b_{i_m}) \rightarrow f_{n+1} = a.$$

We correspond to this rule the subtable $T' = T(f_{i_1}, b_{i_1}) \ldots (f_{i_m}, b_{i_m})$ of the table T. If a is a common decision for T', then the work of H is finished and the constructed rule is returned. Otherwise, we should select a new attribute $f_{i_{m+1}}$ and construct a new rule:

$$(f_{i_1} = b_{i_1}) \wedge \ldots \wedge (f_{i_m} = b_{i_m}) \wedge (f_{i_{m+1}} = b_{i_{m+1}}) \rightarrow f_{n+1} = a.$$

Denote $T'' = T'(f_{i_{m+1}}, b_{i_{m+1}})$, $M(f_{i_{m+1}}, r, a) = M(T'', a) = N(T'') - N(T'', a)$, and

$$RM(f_{i_{m+1}}, r, a) = (N(T'') - N(T'', a))/N(T'').$$

We denote $\alpha(f_{i_{m+1}}, r, a) = N(T', a) - N(T'', a)$ and $\beta(f_{i_{m+1}}, r, a\ ) = M(T', a) - M(T'', a)$. We describe now how each greedy heuristic algorithm with fixed decision selects the attribute $f_{i_{m+1}}$.

- Heuristic "M" selects an attribute $f_{i_{m+1}} \in E(T')$ which minimizes the value $M(f_{i_{m+1}}, r, a)$.

- Heuristic "RM" selects an attribute $f_{i_{m+1}} \in E(T')$ which minimizes the value $RM(f_{i_{m+1}}, r, a)$.

- Heuristic "maxCov" selects an attribute $f_{i_{m+1}} \in E(T')$ which minimizes the value $\alpha(f_{i_{m+1}}, r, a)$ given that $\beta(f_{i_{m+1}}, r, a) > 0$.

- Heuristic "poly" selects an attribute $f_{i_{m+1}} \in E(T')$ which maximizes the value $\frac{\beta(f_{i_{m+1}},r,a)}{\alpha(f_{i_{m+1}},r,a)+1}$.

- Heuristic "log" selects an attribute $f_{i_{m+1}} \in E(T')$ which maximizes the value $\frac{\beta(f_{i_{m+1}},r,a)}{\log_2(\alpha(f_{i_{m+1}},r,a)+2)}$.

Let H be one of the previous heuristics. For a row r of the table T, we apply H to the row r and each decision $a \in D(r)$. As a result, we obtain $|D(r)|$ rules. Depending on our aim, we either choose among these rules a rule with minimum length or a rule with maximum coverage. After applying this procedure to each row of T, we obtain a system of rules for T.

Example 3.1. Let us consider the decision table $J^{m-v}_{f_3}$ (see Fig. 2), row r_2 from $I^{m-v}_{f_3}$, and decision 1 from $D(r_2)$. Now, we present how heuristic H constructs a decision rule for $J^{m-v}_{f_3}$, r_2, and 1. This heuristic starts with the following rule: $\rightarrow f_3 = 1$.

Let $T = J^{m-v}_{f_3}$. In decision table T we have only two conditional attributes, so we will consider two subtables: $T'_1 = T(f_1, 2)$ and $T'_2 = T(f_2, 1)$ (see Fig. 5).

$T'_1 = T(f_1, 2) =$

	f_1	f_2	
r_2	2	1	{1}
r_3	2	0	{0}

$T'_2 = T(f_2, 1) =$

	f_1	f_2	
r_1	1	1	{1}
r_2	2	1	{1}

Figure 5. Subtables of the table $T = J^{m-v}_{f_3}$

- Heuristic "M" : $M(f_1, r_2, 1) = 1$, $M(f_2, r_2, 1) = 0$, so the condition $f_2 = 1$ is added to the rule $\rightarrow f_3 = 1$. The decision 1 is a common decision for subtable T'_2. Heuristic "M" returns the rule $f_2 = 1 \rightarrow f_3 = 1$.

- Heuristic "RM" : $RM(f_1, r_2, 1) = \frac{1}{2}$, $RM(f_2, r_2, 1) = 0$, so we obtain the rule $f_2 = 1 \rightarrow f_3 = 1$.

- Heuristic "maxCov" : $\alpha(f_1, r_2, 1) = 1$, $\beta(f_1, r_2, 1) = 0$, $\alpha(f_2, r_2, 1) = 0$, $\beta(f_2, r_2, 1) = 1$ so we obtain the rule $f_2 = 1 \rightarrow f_3 = 1$.

- Heuristic "poly" : $\frac{\beta(f_1,r_2,1)}{\alpha(f_1,r_2,1)+1} = \frac{0}{2}$, $\frac{\beta(f_2,r_2,1)}{\alpha(f_2,r_2,1)+1} = 1$, so we obtain the rule $f_2 = 1 \rightarrow f_3 = 1$.

- Heuristic "log" : $\frac{\beta(f_1,r_2,1)}{\log_2(\alpha(f_1,r_2,1)+2)} = \frac{0}{\log_2 3}$, $\frac{\beta(f_2,r_2,1)}{\log_2(\alpha(f_2,r_2,1)+2)} = \frac{1}{\log_2 2}$, so we obtain the rule $f_2 = 1 \rightarrow f_3 = 1$.

3.2. Heuristics with Most Common Decision

Let $r = (b_1, \dots, b_n)$ be a row of T. Now, we present heuristic methods with most common decision that construct a rule for T and r. Each heuristic H starts with empty rule $\rightarrow$, and then sequentially adds conditions to the left-hand side of this rule. Let during the work of the heuristic H, we already constructed the following rule:

$$(f_{i_1} = b_{i_1}) \wedge \dots \wedge (f_{i_m} = b_{i_m}) \rightarrow$$

We correspond to this rule the subtable $T' = T(f_{i_1}, b_{i_1}) \dots (f_{i_m}, b_{i_m})$ of the table T. If T' has a common decision a for T', then the work of H is finished and the rule

$$(f_{i_1} = b_{i_1}) \wedge \dots \wedge (f_{i_m} = b_{i_m}) \rightarrow a$$

is returned. Otherwise, we should select a new attribute $f_{i_{m+1}}$ and construct a new rule:

$$(f_{i_1} = b_{i_1}) \wedge \dots \wedge (f_{i_m} = b_{i_m}) \wedge (f_{i_{m+1}} = b_{i_{m+1}}) \rightarrow .$$

We describe now how each greedy heuristic algorithm with most common decision selects the attribute $f_{i_{m+1}}$.

- Heuristic "me" selects an attribute $f_{i_{m+1}} \in E(T')$ which minimizes the value $N(T'') - N(T'', mcd(T''))$ where $T'' = T'(f_{i_{m+1}}, b_{i_{m+1}})$.
- Heuristic "mep" selects an attribute $f_{i_{m+1}} \in E(T')$ which minimizes the value $(N(T'') - N(T'', mcd(T'')))/N(T'')$.

Let H be either "me" or "mep" . For each row r of the table T, we apply H to the row r. As a result, we obtain one rule for each row that constitutes a system of rules for T.

Example 3.2. Let us consider the decision table $J_{f_3}^{m-v}$ from Fig. 2. We will present how heuristic H constructs a decision rule for the row r_2 of $J_{f_3}^{m-v}$. This heuristic starts with empty rule $\rightarrow$.

Let $T = J_{f_3}^{m-v}$. In decision table T we have only two conditional attributes, so we will consider two subtables: $T_1' = T(f_1, 2)$ and $T_2' = T(f_2, 1)$ depicted in Fig. 5, where $mcd(T_1') = 0$, and $mcd(T_2') = 1$.

- Heuristic "me" : $N(T_1') - N(T_1', 0) = 1$, $N(T_2') - N(T_2', 1) = 0$, so we obtain the rule $f_2 = 1 \rightarrow f_3 = 1$.
- Heuristic "mep" : $(N(T_1') - N(T_1', 0))/N(T_1') = \frac{1}{2}$, $(N(T_2') - N(T_2', 1))/N(T_2') = 0$, so we obtain the rule $f_2 = 1 \rightarrow f_3 = 1$.

4. Experimental Results

We carried out the experiments on data sets from UCI Machine Learning Repository [23] using software system Dagger [24]. We performed some preprocessing such as conditional attributes that take unique value for each row were removed. Also, 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. Missing values were replaced with the most common value of the corresponding attributes. Table 1 contains names of 12 data sets that were considered as information systems along with information of the number of rows and attributes for each data set.

Table 1. Data sets considered as information systems

Data set	Rows	Attributes
adult-stretch	16	5
balance-scale	625	5
breast-cancer	266	10
cars	1728	7
hayes-roth-data	69	5
lenses	24	5
monks-1-test	432	7
monks-3-test	432	7
shuttle-landing	15	7
teeth	23	9
tic-tac-toe	958	10
zoo-data	59	17

For each heuristic method H and each table $T \in \Phi^{m-v}(I)$, we first compute the average length of rules for system of rules constructed by H for T, denoted by $length\_greedy$. Second, we compute the average length of rules for optimal relative to length system of rules constructed by dynamic programming algorithm for T (see [25, 26, 27, 28] for decision tables with single-valued decisions and [29] for decision tables with many-valued decisions), denoted by $length\_min$. Finally, we calculate for each $T \in \Phi^{m-v}(I)$ the relative difference $\frac{length\_greedy - length\_min}{length\_min}$ (we assume that $\frac{0}{0} = 0$), and then sum the relative differences and average the summation over $|\Phi^{m-v}(I)|$. Table 2 shows the obtained average relative difference (ARD) for each information system I along with the overall ARD on the all information systems. For the best heuristic algorithm "M", the overall ARD between length of rules constructed by this heuristic and optimal rules is at most 2%.

Similar study was done for each information system from Table 1 in the case of coverage. First, we compute the average coverage of rules for system of rules constructed by H for T, denoted by $coverage\_greedy$. Second, we compute the average coverage of rules for optimal relative to coverage system of rules constructed by dynamic programming algorithm for T, denoted by $coverage\_max$. Third, we calculate the value of relative difference $\frac{coverage\_max - coverage\_greedy}{coverage\_max}$ for each $T \in \Phi^{m-v}(I)$, and then sum the relative differences and average the summation over $|\Phi^{m-v}(I)|$. Table 3 contains the obtained ARD for each information system and overall ARD on the all information systems. The

Table 2. Decision tables with many-valued decisions: ARD for length

Dataset	poly	log	maxCov	me	mep	M	RM
adult-stretch	0.00	0.00	0.41	0.15	0.15	0.00	0.00
balance-scale	0.03	0.03	0.06	0.04	0.04	0.03	0.03
breast-cancer	0.64	0.14	1.43	0.05	0.30	0.02	0.18
cars	0.06	0.04	0.39	0.06	0.09	0.04	0.06
hayes-roth-data	0.03	0.02	0.21	0.07	0.08	0.02	0.02
lenses	0.00	0.00	0.48	0.06	0.05	0.03	0.00
monks-1-test	0.00	0.00	0.82	0.03	0.03	0.00	0.00
monks-3-test	0.01	0.01	0.65	0.02	0.02	0.00	0.00
shuttle-landing	0.27	0.01	0.66	0.06	0.16	0.00	0.03
teeth	0.90	0.69	1.90	0.00	0.00	0.00	0.00
tic-tac-toe	0.21	0.07	0.71	0.15	0.26	0.07	0.14
zoo-data	1.25	0.59	2.22	0.05	0.17	0.01	0.05
Overall ARD	0.28	0.13	0.83	0.06	0.11	0.02	0.04

Table 3. Decision tables with many-valued decisions: ARD for coverage

Dataset	poly	log	maxCov	me	mep	M	RM
adult-stretch	0.00	0.00	0.06	0.09	0.09	0.07	0.07
balance-scale	0.00	0.00	0.00	0.00	0.00	0.00	0.00
breast-cancer	0.13	0.35	0.40	0.63	0.54	0.62	0.53
cars	0.01	0.00	0.03	0.27	0.27	0.27	0.27
hayes-roth-data	0.05	0.04	0.10	0.15	0.12	0.12	0.08
lenses	0.00	0.00	0.08	0.06	0.06	0.06	0.05
monks-1-test	0.01	0.01	0.14	0.02	0.02	0.01	0.01
monks-3-test	0.01	0.01	0.07	0.05	0.05	0.05	0.05
shuttle-landing	0.00	0.01	0.02	0.31	0.31	0.31	0.31
teeth	0.01	0.01	0.04	0.29	0.29	0.29	0.29
tic-tac-toe	0.27	0.47	0.60	0.67	0.54	0.61	0.43
zoo-data	0.03	0.05	0.11	0.38	0.37	0.37	0.36
Overall ARD	0.04	0.08	0.14	0.24	0.22	0.23	0.20

results seem to be promising for the best heuristic, “poly”, because the overall ARD for this heuristic is at most 4%.

In the same way, we compare rules obtained on tables from $\Phi^{s-v}(I)$ by the considered heuristics with the optimal rules. Results can be found in Tables 4 and 5. It turns out that the best heuristic algorithms are “RM” and “poly” for length and coverage, respectively.

Avoiding bias during averaging (see Tables 2 – 5) can be achieved by considering ranks of the heuristics. Table 6 shows overall ranks separately for the heuristics on tables from $\Phi^{m-v}(I)$ and $\Phi^{s-v}(I)$ for the two rule characteristics length and coverage. From the considered results, it follows that, for the length minimization, we should use the heuristic “M”. For the coverage maximization we should use the heuristic “poly”.

The number of constructed association rules can be considered as an important factor from the point of view of knowledge representation. Table 7 presents the number of unique rules relative to

Table 4. Decision tables with single-valued decisions: ARD for length

Dataset	poly	log	maxCov	me	mep	M	RM
adult-stretch	0.00	0.00	0.68	0.47	0.47	0.34	0.00
balance-scale	0.00	0.00	0.05	0.03	0.03	0.01	0.00
breast-cancer	0.62	0.14	1.40	0.05	0.29	0.02	0.17
cars	0.06	0.03	0.25	0.12	0.15	0.04	0.03
hayes-roth-data	0.04	0.02	0.19	0.06	0.09	0.02	0.02
lenses	0.07	0.07	0.51	0.13	0.12	0.09	0.04
monks-1-test	0.00	0.00	0.77	0.10	0.03	0.00	0.00
monks-3-test	0.01	0.01	0.89	0.29	0.04	0.00	0.00
shuttle-landing	0.24	0.00	0.56	0.06	0.16	0.00	0.02
teeth	0.90	0.69	1.90	0.00	0.00	0.00	0.00
tic-tac-toe	0.21	0.07	0.71	0.15	0.26	0.07	0.14
zoo-data	1.11	0.60	2.12	0.05	0.21	0.01	0.07
Overall ARD	0.27	0.14	0.84	0.12	0.15	0.05	0.04

Table 5. Decision tables with single-valued decisions: ARD for coverage

Dataset	poly	log	maxCov	me	mep	M	RM
adult-stretch	0.00	0.00	0.13	0.09	0.09	0.06	0.00
balance-scale	0.00	0.00	0.00	0.00	0.00	0.00	0.00
breast-cancer	0.12	0.35	0.39	0.63	0.54	0.62	0.53
cars	0.00	0.00	0.02	0.27	0.27	0.27	0.27
hayes-roth-data	0.04	0.03	0.08	0.15	0.12	0.11	0.07
lenses	0.00	0.00	0.10	0.05	0.04	0.04	0.02
monks-1-test	0.00	0.00	0.10	0.02	0.02	0.00	0.00
monks-3-test	0.01	0.01	0.13	0.09	0.04	0.04	0.04
shuttle-landing	0.01	0.01	0.03	0.48	0.48	0.48	0.48
teeth	0.01	0.01	0.04	0.29	0.29	0.29	0.29
tic-tac-toe	0.27	0.47	0.60	0.67	0.54	0.61	0.43
zoo-data	0.03	0.04	0.11	0.38	0.36	0.37	0.36
Overall ARD	0.04	0.08	0.15	0.26	0.23	0.24	0.21

Table 6. Overall ranks for the heuristics

		Heuristics						
		poly	log	maxCov	me	mep	M	RM
Single-valued decisions	Length	4	3	7	5	6	1	2
	Coverage	1	2	5	7	6	4	3
Many-valued decisions	Length	4	2	7	5	6	1	3
	Coverage	1	2	4	7	6	5	3

an upper bound on the number of rules for information system I (data sets presented in Table 1), for single-valued decision model. An upper bound on the number of rules for information system I it is a number of rows muliplicated by the number of attributes. It is possible to see that on average, the number of unique rules is much smaller than the upper bound on the number of association rules, for all considered heuristics.

Table 7. The number of rules constructed by heuristics relative to an upper bound

Dataset	Upper bound	M	RM	maxCov	poly	log	me	mep
adult-stretch	80	0.20	0.20	0.20	0.20	0.20	0.20	0.20
balance-scale	3125	0.20	0.20	0.20	0.20	0.20	0.20	0.20
breast-cancer	2660	0.10	0.10	0.10	0.10	0.10	0.10	0.10
cars	12096	0.14	0.14	0.14	0.14	0.14	0.14	0.14
hayes-roth-data	345	0.20	0.20	0.20	0.20	0.20	0.20	0.20
lenses	120	0.20	0.20	0.20	0.20	0.20	0.20	0.20
monks-1-test	3024	0.14	0.14	0.14	0.14	0.14	0.14	0.14
monks-3-test	3024	0.14	0.14	0.14	0.14	0.14	0.14	0.14
shuttle-landing	105	0.14	0.14	0.14	0.14	0.14	0.14	0.14
teeth	207	0.11	0.11	0.11	0.11	0.11	0.11	0.11
tic-tac-toe	9580	0.10	0.10	0.10	0.10	0.10	0.10	0.10
zoo-data	1003	0.06	0.06	0.06	0.06	0.06	0.06	0.06
Average		0.14	0.14	0.14	0.14	0.14	0.14	0.14

Table 8 presents times of constructing rules (in [ms]), for considered heuristics.

Table 8. Generation times of rules constructed by heuristics [ms]

Dataset	M	RM	maxCov	poly	log	me	mep
adult-stretch	12	16	17	14	20	15	14
balance-scale	6732	3803	4680	4113	4287	3856	4071
breast-cancer	5327	6519	11685	10082	4255	5044	6313
cars	57846	73456	62120	83020	85352	61206	91524
hayes-roth-data	98	119	75	109	114	96	109
lenses	58	37	27	39	22	40	29
monks-1-test	6290	6156	7259	4772	5128	5127	4903
monks-3-test	4903	4685	6191	4549	4713	4789	4428
shuttle-landing	19	22	43	38	28	30	16
teeth	55	46	64	47	52	52	42
tic-tac-toe	3530	43817	48244	45278	37560	35319	45078
zoo-data	843	781	1940	775	876	1089	893
Average	11859.42	11621.42	12028.75	12736.33	11867.25	9721.92	13118.33

Presented approach for association rule construction if different from the approach based on frequent itemsets, however it is possible to compare the number of rules and times of their constructing. Table 9 presents the number and generation times of assocciation rules (in [ms]) induced by Apriori algorithm implemented by C. Borgelt (`http://www.borgelt.net/apriori.html`). The column Upper bound denotes the number of rules calculated by brute-force approach. The total number of possible rules extracted from a data set that contains d items equals $3^d - 2^{d+1} + 1$ [30]. Column No. rules presents number of unique association rules constructed by Apriori algorithm which was running with default settings i.e., minimal support: 10% and minimal confidence: 80%. Column Ratio presents the number of unique rules relative to an upper bound, column Time [ms] presents generation times of assocciation rules.

Based on the results presented in Tables 7, 8 and 9 we can see that taking into account compu-

Table 9. The number of rules constructed by Apriori algorithm relative to an upper and time of their generation

Dataset	Upper bound	No. rules	Ratio	Time [ms]
adult-stretch	57002	68	1.19E-03	4
balance-scale	94126401612	0	0.00E+00	6
breast-cancer	3.28257E+20	1026	3.13E-18	11
cars	10456158900	16	1.53E-09	27
hayes-roth	386896202	3	7.75E-09	4
lenses	18660	39	2.09E-03	4
monks-1-test	1161212892	8	6.89E-09	7
monks-3-test	1161212892	15	1.29E-08	6
shuttle	3484687250	532	1.53E-07	7
teeth	6.17669E+14	1270	2.06E-12	7
tic-tac-toe	6.86293E+13	20	2.91E-13	16
zoo	3.28257E+20	926117	2.82E-15	562
average	5.47E+19	77426.17	2.74E-04	55.08

tational times, Apriori algorithm is much faster for association rules construction than the proposed heuristics. However, taking into account the number of generated rules, it turns out that Apriori algorithm is strongly dependent on the number of attributes and number of values of these attributes. It is still only a fraction of all possible combinations generated by the brute-force approach. When comparing to the proposed heuristics, it is evident that they are much more immune and dependent to data characteristics.

5. Conclusions

In the paper, seven heuristics for construction of association rules in the frameworks of both multi-valued and single-valued decision approaches, were compared. It was shown that the average relative difference between coverage of rules constructed by the best heuristic and maximum coverage of rules is at most 4%. The same situation is with length.

The presented approach for association rules construction is different from the known one based on frequent itemsets. However, it allows us to obtain “important” in some sense rules (from the point of view of length and coverage), in a reasonable time. So, studied heuristics can be considered as significant from the point of view of knowledge representation.

Future works will be connected with an application of the best heuristics in a feature selection process.

Acknowledgements

Research reported in this publication was supported by the King Abdullah University of Science and Technology (KAUST) and University of Silesia in Katowice.

References

[1] Agrawal R, Imieliński T, Swami A. Mining Association Rules between Sets of Items in Large Databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207–216. ACM, 1993.

[2] Savasere A, Omieciński E, Navathe SB. An Efficient Algorithm for Mining Association Rules in Large Aatabases. In: Proceedings of 21th International Conference on Very Large Data Bases, pp. 432–444. Morgan Kaufmann, 1995.

[3] Park JS, Chen MS, Yu PS. An Effective Hash Based Algorithm for Mining Association Rules. In: Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, pp. 175–186. ACM Press, 1995.

[4] Rauch J. Observational Calculi and Association Rules, volume 469 of *Studies in Computational Intelligence*. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-11736-7.

[5] Borgelt C. Simple Algorithms for Frequent Item Set Mining. In: Advances in Machine Learning II, volume 263 of *Studies in Computational Intelligence*, pp. 351–369. Springer Berlin Heidelberg, 2010.

[6] Han J, Pei J, Yin Y, Mao R. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. *Data Min. Knowl. Discov.*, 2004. **8**(1):53–87.

[7] Herawan T, Deris MM. A Soft Set Approach for Association Rules Mining. *Knowledge-Based Systems*, 2011. **24**(1):186–195.

[8] Geng L, Hamilton HJ. Interestingness Measures for Data Mining: A Survey. *ACM Comput. Surv.*, 2006. **38**(3).

[9] Han J, Kamber M. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000.

[10] Wieczorek A, Słowiński R. Generating a Set of Association and Decision Rules with Statistically Representative Support and Anti-support. *Information Sciences*, 2014. **277**:56–70.

[11] Rissanen J. Modeling by Shortest Data Description. *Automatica*, 1978. **14**(5):465–471.

[12] Bonates T, Hammer PL, Kogan A. Maximum Patterns in Datasets. *Discrete Applied Mathematics*, 2008. **156**(6):846–861.

[13] Moshkov M, Piliszczuk M, Zielosko B. Greedy Algorithm for Construction of Partial Association Rules. *Fundamenta Informaticae*, 2009. **92**(3):259–277.

[14] Nguyen HS, Ślęzak D. Approximate Reducts and Association Rules - Correspondence and Complexity Results. In: Zhong N, Skowron A, Ohsuga S (eds.), RSFDGrC, volume 1711 of *LNCS*, pp. 137–145. Springer, 1999.

[15] Feige U. A Threshold of $\ln n$ for Approximating Set Cover. In: Journal of the ACM (JACM), volume 45, pp. 634–652. ACM New York, 1998.

[16] Pawlak Z, Skowron A. Rudiments of Rough Sets. *Information Sciences*, 2007. **177**(1):3–27.

[17] Agrawal R, Mannila H, Srikant R, Toivonen H, Verkamo AI. Fast Discovery of Association Rules. In: Fayyad UM, Piatetsky-Shapiro G, Smyth P, Uthurusamy R (eds.), Advances in Knowledge Discovery and Data Mining, pp. 307–328. American Association for Artificial Intelligence, 1996.

[18] Zaki MJ, Parthasarathy S, Ogihara M, Li W. New Algorithms for Fast Discovery of Association Rules. Technical report, Rochester, NY, USA, 1997.

[19] Zielosko B. Application of Dynamic Programming Approach to Optimization of Association Rules Relative to Coverage and Length. *Fundamenta Informaticae*, 2016. **148**(1-2):87–105.

[20] Borgelt C, Kruse R. Induction of Association Rules: Apriori Implementation. In: 15th Conference on Computational Statistics (Compstat 2002, Berlin, Germany), pp. 395–400. Physica Verlag, Heidelberg, 2002.

[21] Moshkov M, Piliszczuk M, Zielosko B. On Construction of Partial Association Rules. In: Rough Sets and Knowledge Technology, 4th International Conference, volume 5589 of *Lecture Notes in Computer Science*, pp. 176–183. Springer, 2009.

[22] Alsolami F, Amin T, Moshkov M, Zielosko B. Comparison of Heuristics for Optimization of Association Rules. In: Proceedings of the 24th International Workshop on Concurrency, Specification and Programming, Rzeszow, Poland, September 28-30, 2015. 2015 pp. 4–11.

[23] Asuncion A, Newman DJ. UCI Machine Learning Repository, `http://www.ics.uci.edu/~mlearn/`, accessed Feb. 2016.

[24] 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, pp. 29–39. Blue Herons, 2011.

[25] Amin T, Chikalov I, Moshkov M, Zielosko B. Dynamic Programming Approach for Partial Decision Rule Optimization. *Fundamenta Informaticae*, 2012. **119**(3-4):233–248.

[26] Amin T, Chikalov I, Moshkov M, Zielosko B. Dynamic Programming Approach to Optimization of Approximate Decision Rules. *Information Sciences*, 2013. **221**:403–418.

[27] Zielosko B. Sequential Optimization of γ-decision Rules. In: Federated Conference on Computer Science and Information Systems, pp. 339–346. 2012.

[28] Zielosko B, Chikalov I, Moshkov M, Amin T. Optimization of Decision Rules Based on Dynamic Programming Approach. In: Innovations in Intelligent Machines (4), volume 514 of *Studies in Computational Intelligence*, pp. 369–392. Springer, 2014.

[29] Moshkov M, Zielosko B. Combinatorial Machine Learning - A Rough Set Approach, volume 360 of *Studies in Computational Intelligence*. Springer, 2011.

[30] Tan PN, Steinbach M, Karpatne A, Kumar V. Introduction to Data Mining (2Nd Edition). Pearson, 2nd edition, 2018.