Comparison of Heuristics for Optimization of Association Rules

Fawaz Alsolami[1], Talha Amin[1], Mikhail Moshkov[1], and Beata Zielosko[2]

[1] 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
[2] Institute of Computer Science, University of Silesia
39, Bedzinska St., 41-200 Sosnowiec, Poland
beata.zielosko@us.edu.pl

Abstract. In this paper, five 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.

Key words: 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 among sets of items in data set.

There are many algorithms for construction of association rules. One of the most popular is Apriori algorithm based on frequent itemsets [1]. During years, many new algorithms were designed which are based on, e.g., hash based technique [15], partitioning the data [18], and others [7, 10, 19].

The most popular measures for mining association rules are support and confidence [9], however in the paper length and coverage as rule evaluation measures are considered. The choice of length is connected with the Minimum Description Length Principle [17]. Shorter rules are better from the point of view of understanding and interpreting by experts. Search of rules with big coverage allows us to discover major patterns in the data, and it is important from the point of view of knowledge representation.

In the paper, greedy algorithms for construction of association rules are studied since the problems of construction of rules with minimum length or maximum coverage are NP-hard [6, 12, 14]. The most part of approaches, with the exception of brute-force, Apriori algorithm or extensions of dynamic programming, cannot guarantee the construction of optimal rules (i.e., rules with minimum length or maximum coverage).

In the paper [12], it was shown based on results of U. Feige [8] that, under reasonable assumptions on the class NP, some greedy algorithm is close to the best polynomial approximate algorithms for minimization of association rule length. We do not know about similar results for coverage.

Application of rough sets theory to the construction of rules for knowledge representation or classification tasks are usually connected with the usage of decision table [16] as a form of input data representation. In such a table one attribute is distinguished as a decision attribute and it relates to a rule's consequence. However, in the last years, associative mechanism of rule construction, where all attributes can occur as premises or consequences of particular rules, is popular. Association rules can be defined in many ways. In the paper, a special kind of association rules is studied, i.e., they relate to decision rules. Similar approach was considered in [12, 13], where a greedy algorithm for minimization of length of association rules was investigated.

In this paper, we consider five greedy heuristics for construction of association rules and compare them from the point of view of the length and coverage of constructed rules. We also compare the obtained rules with optimal ones constructed by dynamic programming algorithms. We show that 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.

The paper consists of five sections. Section 2 contains main notions. In Sect. 3, we discuss five greedy heuristics. Section 4 contains experimental results for decision tables from UCI Machine Learning Repository, and Sect. 5 – short conclusions.

2 Main Notions

An *information system* I is a rectangular 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.

An association rule for I is a rule of the kind

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

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.

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 $\Phi^{m-v}(I)$. We denote by $\Phi^{s-v}(I)$ 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. 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.

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 $E(T)$ 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 $a_1, \ldots, a_m$.

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 $T' = T(f_{i_1}, a_1), \ldots, (f_{i_m}, a_m)$ 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 five greedy heuristics on an example of the table $T = I^{m-v}_{f_{n+1}}$. Let $r = (b_1, \ldots, b_n)$ be a row of T and a be a decision from $D(r)$. A heuristic H constructs a decision rule for T and r. 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 five greedy heuristics select 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 considered heuristics. For a row r of the table T, we apply it 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.

4 Experimental Results

Experiments were made using data sets from UCI Machine Learning Repository [5] and software system Dagger [2]. Some decision tables contain conditional attributes that take unique value for each row. Such attributes were removed. In some tables there were equal rows with, possibly, different decisions. In this case each group of identical

rows was replaced with a single row from the group with the most common decision for this group. In some tables there were missing values. Each such value was replaced with the most common value of the corresponding attribute. Prepared 12 data sets were considered as information systems (see Table 1 which contains some information about each of these information systems).

Table 1. Data sets considered as information systems

Data set	Rows	Attr
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 information system I, we construct the set $\Phi^{m-v}(I)$ of decision tables with many-valued decisions and the set $\Phi^{s-v}(I)$ of decision tables with single-valued decisions. For each row r of each table $T \in \Phi^{m-v}(I)$, we apply to this row each of the considered five greedy heuristics as it was described at the end of the previous section. We rank five heuristics for row r relative to the length and coverage of constructed rules and find, for each heuristic, the average ranks relative to length and coverage among all rows of all tables from $\Phi^{m-v}(I)$. After that we consider mean of average ranks among all 12 information systems and obtain overall ranks. Results can be found in Table 2. The best three heuristics for length are M, log, and RM. The best three heuristics for coverage are poly, log, and RM. We study in the same way decision tables with single-valued decisions (see Table 2). The best three heuristics for length are M, RM, and log. The best three heuristics for coverage are poly, log, and RM.

For each heuristic and each row r of each table $T \in \Phi^{m-v}(I)$, we compare the length of rule constructed by heuristic for r (we denote it $length\_greedy$) with minimum length of rule (we denote it $length\_min$) and calculate the relative difference $\frac{length\_greedy - length\_min}{length\_min}$ (we assume that $\frac{0}{0} = 0$). The minimum length of rule can be found by dynamic programming algorithms (see [3, 4, 20, 21] for decision tables with single-valued decisions and [11] for decision tables with many-valued decisions). Later, we find average relative difference among all rows of all tables from $\Phi^{m-v}(I)$, and overall average relative difference for all 12 information systems. Results can be found in Table 3. The best three heuristics for the length are M (2% difference), RM (4%), and log (13%). Similar study was done for coverage and decision tables with many-

valued decisions. The relative difference is given by $\frac{coverage\_max - coverage\_greedy}{coverage\_max}$ where $coverage\_greedy$ is the coverage of the rule constructed by greedy heuristic, and $coverage\_max$ is the maximum coverage of the rule calculated by a dynamic programming algorithm. The best three heuristics for the coverage are poly (4% difference), log (8%), and maxCov (14%).

Table 2. Overall ranks for the heuristics

		Heuristics				
		poly	log	maxCov	M	RM
Single-valued decisions	Length	3.38	2.25	5.00	2.17	2.21
	Coverage	1.67	1.83	4.00	4.21	3.29
Many-valued decisions	Length	3.33	2.33	5.00	1.79	2.54
	Coverage	1.67	1.83	3.67	4.21	3.62

We study in the same way decision tables with single-valued decisions (see results in Table 3). The best three heuristics for the length are RM (4% difference), M (5%), and log (14%). The best three heuristics for the coverage are poly (4% difference), log (8%), and maxCov (15%).

Table 3. Overall average relative differences for the heuristics

		Heuristics				
		poly	log	maxCov	M	RM
Single-valued decisions	Length	0.27	0.14	0.84	0.05	0.04
	Coverage	0.04	0.08	0.15	0.24	0.21
Many-valued decisions	Length	0.29	0.13	0.83	0.02	0.04
	Coverage	0.04	0.08	0.14	0.23	0.20

From the considered results it follows that, for the length minimization, we should use the heuristic M and, probably, the heuristic RM. For the coverage maximization we should use the heuristic poly.

5 Conclusions

We compared five heuristics for construction of association rules in the frameworks of both multi-valued and single-valued decision approaches. We 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. In the future, we are planning to use the best heuristic for coverage in algorithms constructing relatively small systems of rules covering almost all objects in information systems.

Acknowledgements

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

The authors wish to express their gratitude to anonymous reviewers for useful comments.

References

1. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: SIGMOD '93, pp. 207–216. ACM (1993)
2. 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)
3. Amin, T., Chikalov, I., Moshkov, M., Zielosko, B.: Dynamic programming approach for partial decision rule optimization. Fundam. Inform. 119(3-4), 233–248 (2012)
4. Amin, T., Chikalov, I., Moshkov, M., Zielosko, B.: Dynamic programming approach to optimization of approximate decision rules. Inf. Sci. 221, 403–418 (2013)
5. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository (`http://wwwicsuciedu/~mlearn/`, 2007), `http://www.ics.uci.edu/~mlearn/`
6. Bonates, T., Hammer, P.L., Kogan, A.: Maximum patterns in datasets. Discrete Applied Mathematics 156(6), 846–861 (2008)
7. Borgelt, C.: Simple algorithms for frequent item set mining. In: Koronacki, J., Raś, Z.W., Wierzchoń, S.T., Kacprzyk, J. (eds.) Advances in Machine Learning II, Studies in Computational Intelligence, vol. 263, pp. 351–369. Springer Berlin Heidelberg (2010)
8. Feige, U.: A threshold of $\ln n$ for approximating set cover. In: Leighton, F.T. (ed.) Journal of the ACM (JACM), vol. 45, pp. 634–652. ACM New York (1998)
9. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann (2000)
10. Herawan, T., Deris, M.M.: A soft set approach for association rules mining. Knowledge-Based Systems 24(1), 186–195 (2011)
11. Moshkov, M., Zielosko, B.: Combinatorial Machine Learning - A Rough Set Approach, Studies in Computational Intelligence, vol. 360. Springer (2011)
12. Moshkov, M.J., Piliszczuk, M., Zielosko, B.: Greedy algorithm for construction of partial association rules. Fundam. Inform. 92(3), 259–277 (2009)
13. Moshkov, M.J., Piliszczuk, M., Zielosko, B.: On construction of partial association rules. In: Wen, P., Li, Y., Polkowski, L., Yao, Y., Tsumoto, S., Wang, G. (eds.) RSKT, LNCS, vol. 5589, pp. 176–183. Springer (2009)
14. Nguyen, H.S., Ślęzak, D.: Approximate reducts and association rules - correspondence and complexity results. In: Zhong, N., Skowron, A., Ohsuga, S. (eds.) RSFDGrC, LNCS, vol. 1711, pp. 137–145. Springer (1999)
15. Park, J.S., Chen, M.S., Yu, P.S.: An effective hash based algorithm for mining association rules. In: Carey, M.J., Schneider, D.A. (eds.) SIGMOD Conference, pp. 175–186. ACM Press (1995)
16. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Inf. Sci. 177(1), 3–27 (2007)
17. Rissanen, J.: Modeling by shortest data description. Automatica 14(5), 465–471 (1978)
18. Savasere, A., Omiecinski, E., Navathe, S.B.: An efficient algorithm for mining association rules in large databases. In: Dayal, U., Gray, P.M.D., Nishio, S. (eds.) VLDB, pp. 432–444. Morgan Kaufmann (1995)

19. Wieczorek, A., Słowiński, R.: Generating a set of association and decision rules with statistically representative support and anti-support. Information Sciences 277, 56–70 (2014)
20. Zielosko, B.: Sequential optimization of γ-decision rules. In: Ganzha, M., Maciaszek, L.A., Paprzycki, M. (eds.) FedCSIS, pp. 339–346 (2012)
21. Zielosko, B., Chikalov, I., Moshkov, M., Amin, T.: Optimization of decision rules based on dynamic programming approach. In: Faucher, C., Jain, L.C. (eds.) Innovations in Intelligent Machines (4), Studies in Computational Intelligence, vol. 514, pp. 369–392. Springer (2014)