Int. J. Appl. Math. Comput. Sci., 2001, Vol.11, No.3, 691–704

MINIMAL DECISION RULES BASED ON THE APRIORI ALGORITHM †

María C. FERNÁNDEZ*, Ernestina MENASALVAS*, Óscar MARBÁN*
José M. PEÑA**, Socorro MILLÁN***

Based on rough set theory many algorithms for rules extraction from data have been proposed. Decision rules can be obtained directly from a database. Some condition values may be unnecessary in a decision rule produced directly from the database. Such values can then be eliminated to create a more comprehensible (minimal) rule. Most of the algorithms that have been proposed to calculate minimal rules are based on rough set theory or machine learning. In our approach, in a post-processing stage, we apply the Apriori algorithm to reduce the decision rules obtained through rough sets. The set of dependencies thus obtained will help us discover irrelevant attribute values.

Keywords: rough sets, rough dependencies, association rules, Apriori algorithm, minimal decision rules

1. Introduction

One way to construct a simpler model computed from data, easier to understand and with more predictive power, is to create a set of simplified rules (Shan, 1995). A simplified rule (also referred to as a minimal rule or a kernel rule) is the one in which the number of conditions in its antecedent is minimal. Thus, when dealing with decision rules, some condition values can be unnecessary and can be dropped to generate a simplified rule preserving essential information. In (Pawlak, 1991) an approach to simplify decision tables is presented. Such an approach consists of three steps:

- computation of reducts of condition attributes,
- elimination of duplicate rows,
- elimination of superfluous values of attributes.

† This work was partially supported by the Universidad Politècnica de Madrid under projects "Cooperación UPM-Universidad del Valle" and "Implementación de un Data Warehouse capaz de ser integrado con un sistema de Data Mining genérico".

* DLSIIS Facultad de Informática, U.P.M., Madrid, Spain,
e-mail: {cfbaizan, emenasalvas}@fi.upm.es; omarban@pegaso.ls.fi.upm.es
** DATSI Facultad de Informática, U.P.M., Madrid, Spain, e-mail: jmpena@fi.upm.es
*** Universidad del Valle, Cali, Colombia, e-mail: millan@borabora.edu.co

This approach to the problem is not very useful from the practical point of view, because both the computation of reducts and the superfluous equivalence classes are NP-hard.

Many algorithms and methods have been proposed and developed to generate minimal decision rules. From the point of view of algorithms, some of them are based on inductive learning (Grzymala-Busse, 1993; Michalski *et al.*, 1986; Quinlan, 1986), and some others are based on rough set theory (Bazan, 1996; 1998; Shan, 1995; Shan and Ziarko, 1994; 1995; Skowron, 1995; Stefanowski, 1998). Some of the procedures associated with rough set theory are complex to compute. Thus, rough set theory has been applied to solve data mining problems integrated with other knowledge extraction methods.

Shan (1995) proposes and develops a systematic method for computing all minimal rules, called maximally general rules, based on decision matrices. Based on rough set and Boolean reasoning, Bazan (1996) proposes a method to generate decision rules using the concept of dynamic reducts. Dynamic reducts are stable reducts of a given decision table that appear frequently in random samples of a decision table. Skowron (1995) proposes a method that, when applied over consistent decisions tables, makes it possible to obtain minimal decision rules. Based on the relative discernibility matrix notion, the method enables us to calculate the set of all relative reducts of a decision table. An incremental learning algorithm for computing the set of all minimal decision rules based on the decision matrix method is proposed in (Shan and Ziarko, 1995).

On the other hand, some efforts have been made to integrate different data mining tasks for obtaining better rules. In (Choobineh *et al*, 1997), an approach integrating fuzzy logic and modified rough sets is proposed in order to classify new objects into one of the different decisions categories when this classification cannot be perfectly made. Świniarski (1998a; 1998b) proposes integration of rough sets with statistical methods and neural networks. In the former proposal, Principal Component Analysis (PCA) and Bayesian inference are integrated with rough sets for data mining applied to breast cancer detection. PCA is applied for feature extraction and reduction. Rough set theory is used to find reduct sets from patterns obtained as a result of PCA. In the latter, rough sets and neural networks are combined for data mining and pattern recognition and then implemented as a rough-neural system. Rough sets are used to reduce the size of a knowledge base, while neural networks are used for classification. Furthermore, in (Kosters *et al*, 1999) a method to extract clusters based on the Apriori algorithm is proposed. The method considers the highest possible order association rules. One of these rules with the highest confidence is selected, and all the customers that bought items in the antecedent rule constitute a first cluster. The customers in this cluster are removed from the original data set and the process is iterated, leading to a hierarchical clustering. The method for stopping uses either a maximum number of cluster parameters or the minimum support threshold for the generated rules. Another combined algorithm based on both the Apriori algorithm and rough set theory can be found in (Lin, 1996).

Kryszkiewicz (1998a; 1998b; Kryszkiewicz and Rybinski, 1998) proposed a new algorithm which mines very large databases based on association rules generation. This algorithm is a modification of the Apriori algorithm (Agrawal *et al.*, 1993) especially suited for seeking rules with a user-specified consequent.

We present an approach based on the Apriori algorithm that, in a post-processing stage, will allow us to reduce decision rules obtained by means of rough sets. We will assume that a positive region has already been calculated, so that a set of possible redundant rules is available. The set of dependencies obtained by applying the Apriori will help us discover irrelevant attribute values.

The proposed approach is based on the following idea: *The problem of finding dependencies with a degree greater than a given number* k *can be assimilated to the problem of discovering rules having a confidence* $\geq k$ *from the set of large itemsets obtained by the application of the Apriori algorithm.* Our approach differs from the one proposed in (Kryszkiewicz, 1998a) on the following points: (i) the input of the algorithm in (Kryszkiewicz, 1998a) is a modified decision table while the input in our approach is a table containing a positive region, (ii) the methods used to extract association rules from the frequent itemsets are different, (iii) our contribution also applies a postprocessing step in which rules are analyzed and pruned.

The reminder of the paper is organized as follows: Section 2 presents the fundamentals of rough sets as well as those of the Apriori algorithm. Section 3 describes both the basis of the new approach and the proposed algorithm itself. In Section 4 the algorithm is validated with different sample datasets. Finally, Section 5 presents a discussion of the results as well as ideas for future research.

2. Preliminaries

2.1. Rough Set Theory

The original rough set model was proposed by Pawlak (1991). This model is concerned with the analysis of deterministic data dependencies. According to Ziarko (1993), rough set theory is the discovery representation and analysis of data regularities. In this model, the objects are classified into indiscernibility classes based on pairs (attribute, values).

Let OB be a non-empty set called the universe, and let IND be an equivalence relation over the universe OB, called the indiscernibility relation, which represents a classification of the universe into classes of objects which are indiscernible or identical in terms of the knowledge provided by given attributes. The main notion in rough set theory is that of the approximation space which is formally defined as

$$A = (OB, IND). \tag{1}$$

The equivalence classes of the relation are also called the elementary sets. Any finite union of elementary sets is called a definable set. Let us take $X \subseteq OB$ which represents a concept. It is not always the case that X can be defined exactly as the union of some elementary sets. That is why two new sets are defined: $\underline{Apr}(X) = \{o \in OB/[o] \subseteq X\}$ will be called the *lower approximation*, and $\overline{Apr}(X) = \{o \in OB/[o] \cap X \neq \emptyset\}$ will be called the *upper approximation*. Any set defined in terms of its lower and upper approximations is called a *rough set*.

2.2. Information Systems

The main computational effort in the process of data analysis in rough set theory is associated with the determination of attribute relationships in information systems. An information system is the quadruple $S = (OB, AT, V, f)$, where:

- OB is a set of objects,
- AT is a set of attributes,
- $V = \bigcup V_a$, V_a being the values of an attribute a,
- $f : OB \times AT \rightarrow V$.

2.3. Decision Tables

Formally, a decision table S is the quadruple $S = (OB, C, D, V, f)$. All the concepts are defined similarly to those of information systems; the only difference is that the set of attributes has been divided into two sets, C and D, which are conditions and decisions, respectively.

Let P be a nonempty subset of $C \cup D$, and let x, y be members of OB. Here x, y are indiscernible by P in S if and only if $f(x, p) = f(y, p)$ for all $p \in P$. Thus P defines a partition on OB. This partition is called a classification of OB generated by P. Then for any subset P of $C \cup D$, we can define an approximation space, and for any $X \subset OB$, the lower approximation of X in S and the upper approximation of X in S will be denoted by $\underline{P}(X)$ and $\overline{P}(X)$, respectively.

2.4. Association Rules

The goal of the association discovery is to find items that imply the presence of other items. An association rule is formally described as follows: Let $I = \{i_1, i_2, \dots, i_n\}$ be a set of literals called items. Let D be a set of transactions where each transaction T is a set of items such that $T \subset I$. An association rule is an implication of the form $X \rightarrow Y$, where $X \subset I$, $Y \subset I$ and $X \cap Y = \emptyset$. The rule $X \rightarrow Y$ holds in the transaction set D with confidence c if $c\%$ of transactions in D that contain X also contain Y. The rule $X \rightarrow Y$ holds in the transaction set D with support s if $s\%$ of transactions in D contain $X \cup Y$.

Given a set of transactions D, the problem of mining association rules is to generate all the association rules that have support and confidence greater than the user-specified minimum support (*minsup*) and minimum confidence (*minconf*), respectively.

In order to derive the association rules, two steps are required:

- find large itemsets for a given *minsup*, and
- compute rules for a given *minconf* based on the itemsets obtained before.

The Apriori algorithm for finding all large itemsets makes multiple passes over the database. In the first pass, the algorithm counts item occurrences to determine large one-item sets. The other passes consist of two steps. First, the large itemsets L_{k-1} found in the $(k-1)$-th pass are used to generate the candidate itemsets C_k; next, all those itemsets which have some $k-1$ subset that is not in L_{k-1} are deleted, yielding C_k. Once the large itemsets are obtained, rules of the form $a \rightarrow (l-a)$ are computed, where $a \subset l$ and l is a large itemset.

3. A New Algorithm to Reduce Decision Rules

In this section we present an algorithm to post-process rules extracted by other classification algorithms. Given the model generated by the previous algorithm to be expressed as a set of rules (*a ruleset*), our objective is to obtain a reduced set of rules in terms of both rule complexity (measured by the number of conditions in the antecedent) and the ruleset size (number of rules). The proposed algorithm calculates the positive region of a decision table by means of Variable Precision Rough Set Model (VPRSM) and takes it as its input.

Each row of the table will be considered as a rule, taking the form $Left_i \rightarrow Right_i$. $Left_i$ is a set of multiple conditions $\{C_1, C_2, \ldots, C_m\}$. These conditions are attribute-value pairs (A_j, v_j) (usually presented in the form $A_j = v_j$) with $A_j \in AT$ and $v_j \in V_{a_j}$ (V_{a_j} is the set of all the possible values attribute A_j can take). $Right_i$ is any possible value of the decision attribute ($Right_i \in V_D$). Thus $R = \{r_1, r_2, \ldots, r_n\}$ will be the set of rules obtained in this way.

There are two different kinds of characteristics in these sets of rules that may increase the complexity of the ruleset:

① The cardinality of the ruleset represents the number of rules the ruleset has. When dealing with real world problems, many algorithms tend to generate a large number of very specialized rules. Each of these rules describes a very reduced set of cases used for training (a very low support).

② Some of the rules in the ruleset have redundant conditions in their antecedents allowing for the removal of one or more conditions from the antecedent, while keeping the meaning and classification power of the rule.

3.1. Formal Representation

In order to present these two abnormalities in a formal way, several definitions have to be introduced.

Rule matching: Let OB be the set of objects and R be a ruleset. We define the rule matching functor as an incomplete function: $Match : OB \times R \longrightarrow V_d$,

$$Match(o, Left \rightarrow Right) = \begin{cases} Right & \text{if } \forall (A = v) \in Left : f(A, o) = v, \\ \text{undefined} & \text{otherwise.} \end{cases} \tag{2}$$

Predicted value: For an object $o \in OB$ and a ruleset R, by the predicted value of o using R we mean

$$\begin{aligned} Pred(o, R) &= Right, \\ \text{if } \exists(Left \rightarrow Right) \in R &: Match(o, Left \rightarrow Right) \neq \text{undefined}. \end{aligned} \tag{3}$$

This functor represents how a ruleset can give a value for each presented object. It is assumed that there is only one predicted value for each object, so that there are no contradictory rules in the ruleset.

Ruleset coverage: Let R be a ruleset, $D \in AT$ a decision attribute, and OB a set of objects. $Cov(R, D, OB) \subseteq OB$ is the subset of objects classified successfully by the ruleset in terms of the values of attribute D (the equivalence classes defined by D),

$$o \in Cov(R, D, OB) \Longleftrightarrow o \in OB \wedge Pred(o, R) = f(D, o). \tag{4}$$

Coverage set calculation by Cov has the distributive property over the union ($\cup$) and the difference ($-$) of rulesets, which can be proved as follows:

$$Cov(R \cup S, D, OB) = Cov(R, D, OB) \cup Cov(S, D, OB), \tag{5}$$

$$\forall o \in Cov(R \cup S, D, OB) \Leftrightarrow Pred(o, R \cup S) = f(D, o), \tag{6}$$

$$\forall o \in Cov(R \cup S, D, OB) \Leftrightarrow \exists r \in R \cup S : Match(o, r) = f(D, o), \tag{7}$$

$$\forall o \in Cov(R \cup S, D, OB) \Leftrightarrow \begin{cases} \exists r \in R : Match(o, r) = f(D, o), \text{or} \\ \exists r \in S : Match(o, r) = f(D, o), \end{cases} \tag{8}$$

$$\forall o \in Cov(R \cup S, D, OB) \Leftrightarrow \begin{cases} Pred(o, R) = f(D, o), \text{or} \\ Pred(o, S) = f(D, o), \end{cases} \tag{9}$$

$$\forall o \in Cov(R \cup S, D, OB) \Leftrightarrow o \in Cov(R, D, OB) \text{ or } o \in Cov(S, D, OB), \tag{10}$$

$$\forall o \in Cov(R \cup S, D, OB) \Leftrightarrow o \in Cov(R, D, OB) \cup Cov(S, D, OB). \tag{11}$$

The proof in the case of the difference operation is analogous.

Ruleset equivalence: Let OB be the set of objects. Then two rulesets, R and R', are defined as equivalent according to D if

$$R \sim_D R' \Longleftrightarrow Cov(R, D, OB) = Cov(R', D, OB). \tag{12}$$

Rule equivalence: Let OB be the set of objects. Then two rules, r and r', are defined as equivalent if

$$r \sim r' \Longleftrightarrow \forall o \in OB \begin{cases} Match(o, r) = Match(o, r'), \text{or} \\ \text{both } Match(o, r) \text{ and } Match(o, r') \text{ are undefined.} \end{cases} \tag{13}$$

Using these definitions, Problems ① and ② can be expressed as follows:

① If R is the original ruleset, the first objective is to find a new ruleset R' with the restriction $R \sim_D R'$. The new ruleset should have fewer rules than the previous one ($|R'| < |R|$). This can be done in two different ways, namely by deleting redundant rules, or by combining a subset of the ruleset obtaining a reduced subset:

- **Redundant rule deletion**: $r \in R$ is a redundant rule if

$$\forall o \in OB\ \exists r' \in R : r' \neq r \begin{cases} Match(o,r) = Match(o,r'), & \text{or both} \\ Match(o,r) \text{ and } Match(o,r') \text{ are undefined} \end{cases} \tag{14}$$

With this definition, it is easy to prove that $R \sim_D R - \{r\}$. Indeed, suppose that R and $R - \{r\}$ are not equivalent. Then

$$Cov(R, D, OB) \neq Cov(R - \{r\}, D, OB), \tag{15}$$

$$\exists o \in OB : Pred(o, R) \neq f(o, D) \text{ or } Pred(o, R - \{r\}) \neq f(o, D), \tag{16}$$

$$\exists o \in OB : Pred(o, R) \neq Pred(o, R - \{r\}). \tag{17}$$

This can only be true under three possible conditions:

$$\text{First:}\quad \exists o \in OB : \quad \begin{aligned} &Pred(o, R - \{r\}) = v_1 \text{ and} \\ &Match(o, r) = v_2 \text{ with } v_1 \neq v_2. \end{aligned} \tag{18}$$

But in this case, the ruleset R would have contradictory rules.

$$\text{Second:}\quad \exists o \in OB : \quad \begin{aligned} &Pred(o, R - \{r\}) = \text{undefined and} \\ &Match(o, r) = v_2. \end{aligned} \tag{19}$$

$$Pred(o, R - \{r\}) = \text{undefined} \Leftrightarrow \nexists r' \in R : r' \neq r \quad Match(o, r') \neq \text{undefined}. \tag{20}$$

Thus this formula and (14) are contradictory.

$$\text{Third:}\quad \exists o \in OB : \quad \begin{aligned} &Pred(o, R - \{r\}) = v_1 \text{ and} \\ &Match(o, r) = \text{undefined}. \end{aligned} \tag{21}$$

$$Pred(o, R - \{r\}) = v_1 \Leftrightarrow \exists r' \in R : r' \neq r \quad Match(o, r') = v_1. \tag{22}$$

But this formula and (14) are contradictory.

- **Rule combination**: A set of rules R' (a subset of the original ruleset R) can be combined to create a new set of rules R'' under the following conditions:

$$R' \cap R'' = \emptyset, \tag{23}$$

$$Cov(R', D, OB) = Cov(R'', D, OB), \tag{24}$$

$$R - R' \cup R'' \text{ has no contradictory rules.} \tag{25}$$

This definition can be used to prove that $R \sim_D ((R - R') \cup R'')$. Indeed, the equivalence relationship between rulesets is calculated based on the coverage sets:

$$R \sim_D ((R-R')\cup R'') \Longleftrightarrow Cov(R, D, OB) = Cov((R-R')\cup R''), D, OB). \quad (26)$$

Using the distributive property of the coverage over the ruleset union and difference, we see that $R \sim_D ((R - R') \cup R'')$ iff

$$Cov(R, D, OB) = Cov(R, D, OB) - Cov(R', D, OB) \cup Cov(R'', D, OB). \quad (27)$$

Using (24), we get

$$Cov(R, D, OB) = Cov(R, D, OB) - Cov(R', D, OB) \cup Cov(R', D, OB), \quad (28)$$

$$Cov(R, D, OB) = Cov(R, D, OB). \quad (29)$$

② The second objective deals with the reduction of the rule complexity. For each rule in R $((Left \rightarrow Right) \in R)$, the left-hand side has one or more predicate conditions that are able to describe the objects covered by the rule. Therefore a value is predicted for all the objects described by these conditions.

One of these conditions $((A_i = v_i) \in Left)$ is called superfluous if

$$\forall o \in OB : Match(o, Left \rightarrow Right) = Match(o, Left - \{(A_i, v_i)\} \rightarrow Right). \quad (30)$$

These conditions are equivalent to

$$\begin{aligned} &\forall (A_j = v_j) \in Left : f(A_j, o) = v_j, \\ &\forall (A_j = v_j) \in Left : f(A_j, o) = v_j \wedge (A_j = v_j) \neq (A_i = v_i). \end{aligned} \quad (31)$$

This formula is true when

$$\forall (A_j = v_j) \in Left : f(A_j, o) = v_j \wedge (A_j = v_j) \neq (A_i = v_i) \Leftrightarrow f(A_i, o) = v_i. \quad (32)$$

3.2. Post-Processing Algorithm

Input: The algorithm takes a ruleset R_o extracted by applying the positive region procedures of VPRSM to a decision table.[2] OB are the objects successfully classified by R_o using the attribute $D \in AT$ as a decision attribute.

Output: A new ruleset R_f that is able to classify the same objects OB.

Algorithm:

❶ The Apriori algorithm is executed to obtain the itemsets of size k or less.

[2] The ruleset is in fact a set of equivalence classes.

❷ A set of association rules is calculated with minimal support and confidence. This set of rules is called AR and the rules have the form

$$(A_1 = v_1) \wedge (A_2 = v_2) \wedge \cdots \wedge (A_n = v_n) \longrightarrow (B = v_b). \tag{33}$$

❸ Each of the rules is processed with the following criteria:

- ❒ If $D = B$, the association rule has the decision attribute as a predicate on its right-hand side.This rule is included in R_f. This kind of rules is called the *direct a-priori rules*. Once this rule is moved to the final ruleset, all the rules in AR and R whose left-hand sides are a supersets of the left-hand sides of this association rule are deleted.
- ❒ If $D \in \{A_1, A_2, \ldots, A_n\}$, the decision attribute appears on the left-hand side of the rule. In these cases the rule is ignored.
- ❒ If D do not belong to the rule, then all the rules in R_o are scanned. If all the conditions from the association rule $(A_1 = v_1) \wedge (A_2 = v_2) \wedge \cdots \wedge (A_n = v_n)$ as well as the right-hand side $B = v_b$ belong to the left-hand side of the decision rule, then the condition $B = v_b$ is deleted from the decision rule, because it is redundant:

$$\text{The association rule : } (\mathit{ALeft} \rightarrow \mathit{ARight}) \in AR, \tag{34}$$

$$\text{All the decision rules : } \forall(\mathit{Left} \rightarrow \mathit{Right}) \in R. \tag{35}$$

$$\begin{aligned} &\text{If } (\mathit{ALeft} \cup \mathit{ARight}) \subseteq \mathit{Left}, \\ &\text{the new decision rule is } (\mathit{Left} - \mathit{ARight}) \rightarrow \mathit{Right}. \end{aligned} \tag{36}$$

❹ Once all the rules in AR have been processed, the remaining rules in R_o are moved to R_f. These rules are called the *reduced rules*.

4. Validation of the Algorithm

This algorithm has been used to reduce the models generated by the positive region calculation with several datasets from the UCI machine learning repository (Blake and Merz, 2001). The datasets selected for this test have the following characteristics:

❶ All the attributes must be discrete (nominal), because we do not deal with continuous values.

❷ The dataset should have more than 8 attributes and 100 objects. When the positive region is calculated for datasets with either few attributes or few objects, the number of rules and their complexity are less significant than in large datasets. Because our algorithm deals with complex datasets, we reduce the cases studied to those of these sizes.

With these conditions only six datasets were selected.

Table 1. Experimental results.

	Positive region			Using the post-processing algorithm									
					A-priori rules			Reduced rules			Total rules		
Dataset	RS	MS	MRC	Size	RS	MS	MRC	RS	MS	MRC	RS	MS	MRC
breast-cancer	176	1.03	9.00	5	16	7.38	4.06	109	1.02	7.33	125	1.83	6.91
led24	200	1.00	24.00	5	10	5.00	5.00	151	1.00	14.28	161	1.25	13.70
mushroom	8107	1.00	22.00	3	170	10.33	2.93	128	1.00	11.01	298	6.32	6.40
soybean-large	629	1.08	35.00	3	121	8.13	2.99	434	1.05	13.18	555	2.60	10.96
vote	342	1.27	16.00	2	3	46.33	2.00	121	1.42	14.49	124	2.51	14.19
zoo	42	1.60	16.00	2	3	35.33	2.00	14	1.64	6.00	17	7.59	5.29

In order to assess the results, the following parameters were calculated:

- *Ruleset size (RS):* The number of rules belonging to the ruleset.
- *Mean support (MS):* The mean number of objects successfully classified by the rules.
- *Mean rule complexity (MRC):* The mean number of conditions on the left-hand side of the rules.

The results obtained are presented in Table 1. The columns are grouped in two blocks:

- The first three columns are the values measured when the positive region algorithm is executed. As can be seen, the mean support (MS) is very low, and in most cases only one sample is described by each rule. The positive region rules are also very complex since all the condition attributes appear in each rule.
- The second block of columns belong to the results extracted by the post-processing algorithm, using as the input the rules from the positive region calculation. These results are divided into three groups:
  - the rules extracted directly by the Apriori algorithm (association rules with the decision attribute on the right-hand sides of the rules),
  - the original decision rules after the post-processing algorithm; some rules were deleted because they are described by direct *a-priori* rules, and the others are filtered, removing redundant conditions and getting rules with less complexity,
  - the measures of both set of rules (direct *a-priori* rules and post-processed rules).

4.1. Evaluation of the Results

As is shown, the ruleset complexity was reduced in terms of both the number of rules and the number of conditions per rule. Some of the rulesets were reduced to 3.68% (from 8107 rules to 298), and the number of conditional predicates to 29.09% (from 22.00 to 6.40).

There is also another main difference betwen the original model and the final ruleset. The rules extracted by any rough set algorithm are independent of one another (there is no object described by more than one rule). The rules extracted by our algorithm, and especially the rules derived directly by the Apriori algorithm, may overlap one another, and therefore there would exist objects described by more than one rule. On the one hand, this characteristic alters the mean support measure (MS) in terms of the numbers of objects classified successfully by direct *a-priori* rules; on the other hand, this MS is correct for evaluating the strength of the rules in terms of how general they are.

The accuracy measures, like confusion matrices, are not included because our post-processing algorithm does not change this feature of the ruleset. If the association rules are extracted by the Apriori with 100% accuracy, the deletion of rules and redundant conditions does not modify the prediction power of the model.

The *Size* parameter in Table 1 defines the maximum itemset size when executing the Apriori algorithm. The selected value depends on the characteristics of each dataset. There is an important trend when different values are used: as the size of the itemset increases, the number of direct *a-priori* rules goes up too and, of course, the number of post-processed rules decreases. An example of how the number of rules changes depending on the values if this parameter is presented in Fig. 1.

A very important element in the performance of the algorithm is the number of the association rules extracted by the Apriori algorithm. This set of rules is scanned by our algorithms in order to classify them according to whether the decision attribute belongs to either the left- or the right-side of the rule or it is not present. The size of this set of rules increases when the itemset size increases (see Fig. 2). too

5. Discussion

In this paper, a new approach in which the Apriori algorithm is used to obtain minimal rules has been presented. The knowledge discovery by means of rough set results in a set of rules all having the features of high confidence but very low supports. On the other hand, all condition attributes will appear in each rule. In order to reduce the set of condition attributes, reduct calculation and other methods have been traditionally used. Our approach is an alternative to reducing both the number of condition attributes and the number of rules. Besides, the average support of rules will increase with the application of our method. In this paper, we have presented some experimental results using datasets with more than 100 records described by more than 8 attributes (large datasets). The results highlight the fact that in all cases the approach enhances the output of the classification algorithm.

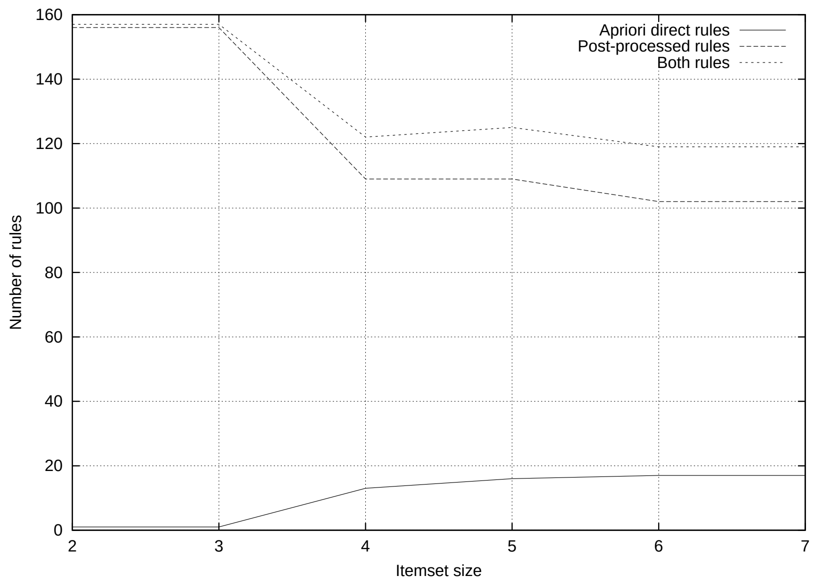

Fig. 1. Number of rules generated by the algorithm.

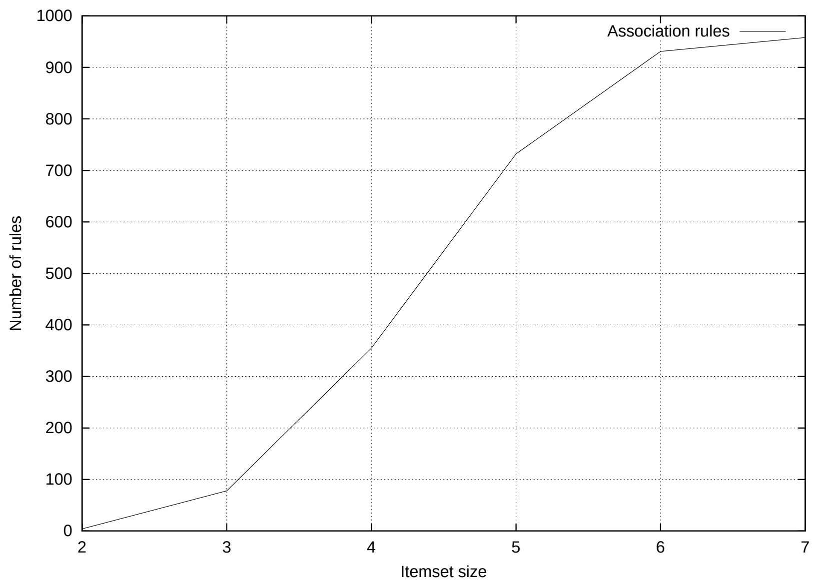

Fig. 2. Number of association rules extracted by the Apriori algorithm.

The main drawback of our approach is the complexity of the resulting algorithm. There are two main nested loops in it: one scans the association rules extracted by the Apriori algorithm and the other the original rules. Nevertheless, let us observe that dealing with huge datasets the original ruleset is neither significant nor comprehensible. So some approach is needed to improve the quality of rules.

As the algorithm effectiveness has been proven, we are currently working on the improvement of its efficiency. We are also working on the application of this approach to models obtained by other classification methods.

References

Agrawal R., Imielinski T., Swami A. (1993): *Mining association rules between sets of items in large databases.* — Proc. ACM SIGMOD Int. Conf. *Management of Data*, Washington, pp.207–216.

Bazan J.G. (1996): *Dynamic reducts and statistical inference.* — Proc. 5-th Int. Conf. *Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU'96*, Granada, Spain, pp.1147–1151.

Bazan J.G. (1998): *A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision tables*, In: Rough Sets in Knowledge Discovery (Polkowski L. and Skowron A., Eds.). — Heidelberg: Physica-Verlag, pp.321–365.

Blake C.L. and Merz C.J. (2001): *UCI Repository of machine learning databases.* — Irvine, CA: University of California, Department of Information and Computer Science, `http://www.ics.uci.edu/∼mlearn/MLRepository.html`.

Grzymala-Busse J.W. (1993): *LERS: A system for learning from examples based on rough sets*, In: Intelligent Decision Support: Handbook of Applicactions and Advances of Rough Set Theory (Slowinski R., Ed.). — Banff, Alberta: Kluwer Netherlands, pp.3–18.

Choobineh F., Paule M., Silkker W. and Hashemei R. (1997): *On integration of modified rough set and fuzzy logic as classifiers.* — Proc. Joint Conf. *Information Sciences*, North Carolina, pp.255–258.

Kosters W., Marchiori E. and Oerlemans A. (1999): *Mining cluster with association rules.* — Lecture Notes in Computer Science 1642, Springer, pp.39–50.

Kryszkiewicz M. (1998a): *String rules in large databases.* — Proc. 7-th Int. Conf. *Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU98*, Paris, Vol.2, pp.1520–1527.

Kryszkiewicz M. (1998b): *Fast discovery of representative association rules.* — Proc. 1-st Int. Conf. *Rough Sets and Current Trends in Computing, RSCTC'98*, Warsaw, Poland, pp.214–221.

Kryszkiewicz M. and Rybinski H. (1998): *Knowledge discovery from large databases using rough sets .* — Proc. 6-th Europ. Congr. *Intelligent Techniques and Soft Computing EUFIT'98*, Aachen, Germany, Vol.1, pp.85–89.

Lin T.Y. (1996): *Rough set theory in very large databases.* — Proc. *Computational Enfineering in Systems Applications, CESA'96*, Lille, France, Vol.2, pp.936–941.

Michalski R., Carbonell J. and Mitchell T.M. (1986): *Machine Learning: An Artificial Intelligence Approach, Vol. 1.* — Palo Alto CA: Tioga Publishing.

Pawlak Z. (1991): *Rough Sets—Theoretical Aspects of Reasoning about Data.* — Dordrecht: Kluwer.

Quinlan J.R. (1986): *Induction of decision trees.* — Mach. Learn., Vol.1, pp.81–106.

Shan N. (1995): *Rule discovery from data using decision matrices.* — M.Sc. Thesis, University of Regina.

Shan N. and Ziarko W. (1994): *An incremental learning algorithm for constructing decision rules*, In: Rough Sets, Fuzzy Sets and Knowledge Discovery (W. Ziarko, Ed.). — Berlin: Springer, pp.326–334.

Shan N. and Ziarko W. (1995): *Data-base acquisition and incremental modification of classification rules.* — Comput. Intell., Vol.11, No.2, pp.357–369.

Skowron A. (1995): *Extracting laws from decision tables: A rough set approach.* — Comput. Intell., Vol.11, No.2, pp.371–387.

Stefanowski J. (1998): *On rough set based approaches to induction of decision rules*, In: Rough Sets in Knowledge Discovery (Polkowski L. and Skowron A., (Eds.). — Heidelberg: Physica-Verlag, pp.500–529.

Świniarski R. (1998a): *Rough sets and Bayesian methods applied to cancer detection.* — Proc. *Rough Sets and Current Trends in Computing, RSCTC'98*, Lecture Notes in Artificial Intelligence 1424, Berlin, pp.609–616.

Świniarski R. (1998b): *Rough sets and neural networks application to handwritten character recognition by complex Zernike moments.* — Proc. *Rough Sets and Current Trends in Computing, RSCTC'98*, Lecture Notes in Artificial Intelligence 1424, Berlin, pp.616–624.

Ziarko W. (1993): *The discovery, analysis, and representation of data dependencies in databases* . — Proc. *Knowledge Discovery in Databases, KDD-93*, Washington, pp.195–209.