Rule schemas and interesting association action rules mining

Angelina A. Tzacheva

Department of Informatics,
University of South Carolina Upstate,
Spartanburg, SC 29303, USA
E-mail: ATzacheva@uscupstate.edu

Abstract: One of the central problems in knowledge discovery in databases, relies on the very large number of rules that classic rule mining systems extract. This problem is usually solved by means of a post-processing step, that alters the entire volume of extracted rules, in order to output only a few potentially interesting ones. This article presents a new approach that allows the user to explore action rules space locally, without the need to extract and post-process all action rules from a database. This solution is based on rule schemas, a new formalism designed to improve the representation of user beliefs and expectations, and on a novel algorithm for local action rules mining based on schemas.

Keywords: action rules; interesting knowledge discovery.

Reference to this paper should be made as follows: Tzacheva, A.A. (xxxx) 'Rule schemas and interesting association action rules mining', *Int. J. Data Mining, Modelling and Management*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Angelina A. Tzacheva received her PhD in Information Technolgy in 2005 from the University of North Carolina at Charlotte. From 2005 and currently, she is an Assistant Professor at the University of South Carolina Upstate (USA). Her research interests include databases, data mining and human factors in computing.

1 Introduction

Knowledge discovery in databases is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data (Fayyad et al., 1996). The process of knowledge discovery is a complex one, comprising several phases that deal with problem and dataset focusing, data quality and cleaning, pattern or rule mining, evaluation, explanation, reporting and visualisation of discovered knowledge. One of the most important techniques in knowledge discovery is rule mining.

An action rule is a rule extracted from a decision system that describes a possible transition of objects from one state to another with respect to a distinguished attribute called a decision attribute (Raś and Wieczorkowska, 2000). Attributes used to describe objects in a decision system are partitioned into stable and flexible. Values of flexible attributes can be changed. This change can be influenced and controlled by users.

In early approaches, action rules mining was based on comparing profiles of two groups of targeted objects – those that are desirable and those that are undesirable (Raś and Wieczorkowska, 2000). An action rule is defined as a term $[(\omega) \wedge (\alpha \rightarrow \beta)] \Rightarrow (\phi \rightarrow \psi)$, where ω is a conjunction of fixed condition features shared by both groups, $(\alpha \rightarrow \beta)$ represents proposed changes in values of flexible features, and $(\phi \rightarrow \psi)$ is a desired effect of the action.

The discovered knowledge provides an insight of how values of some attributes can be changed so the undesirable objects are shifted to a desirable group. For example, one would like to find a way to improve his or her salary from a low-income to a high-income. Another example, in business area, is when an owner would like to improve his or her company's profits by going from a high-cost, low-income business to a low-cost, high-income business. Action rules tell us what changes within flexible attributes are needed to achieve that goal.

Action rules may provide valuable information from databases. Nevertheless, similarly to association rules (Agrawal and Srikant, 1994), the number of rules extracted is so large that it is impossible to inspect them manually. Moreover, some of the discovered action rules may not be useful to the user, since he/she might be either unable or unwilling to perform the suggested actions.

Focusing on the useful action rules can be done as pruning within a post-processing phase, by using an ontology (Fensel, 1998; Tzacheva, 2009). The latter approach is a solution for the rule reduction problem in action rules mining. However, its main weakness is that the entire set of action rules must be evaluated, and then some of them are eliminated, as non-matching the expert knowledge built within an actions ontology. The process may still leave many action rules which are of no interest to the user.

This paper presents a new approach, in which post-mining principles are introduced into the mining step, with action rules; and, association action rules (AAR) (Raś et al., 2008) in particular . The approach focuses on the interesting action rules without the necessity of extracting all action rules existing in the database. Instead of letting the user inspect huge amounts of output containing thousands of rules, the user may explore the rule space incrementally (Blanchard et al., 2007), starting from his/her own beliefs and knowledge and discovering rules that relate to these beliefs – confirming rules, specialised rules, generalised rules or exception rules. At each step, the user inspects only a small amount of rules and he/she is able to choose the most relevant ones, for further exploration. Global post-processing is avoided in favour of local, focused action rule exploration.

There are two key issues in this approach. The first one is focusing on the expectations and beliefs of the user. A solution for representing the user's beliefs and knowledge is based on the concept of general impressions (GI), first presented in Liu et al. (1997) and later developed in Liu et al. (1999). This paper proposes a novel, more unitary manner of representation – the rule schema, which is not only more flexible and intuitive, but can also use as base elements the concepts from an ontology (Fensel, 1998; Tzacheva, 2009) providing the representation with many possibilities. The ontological aspect is not treated in this paper. Four operations on rule schemas have been proposed, that facilitate the exploration of the action rule space: confirmation and specialisation – discovery of rules with the same conclusion but a more specific condition and a notable improvement the confidence; generalisation – discovery of rules with a more general condition and higher support; exception – discovery of low-support regularities that contradict more general rules (Duval et al., 2007).

The second important aspect of our approach is the mining algorithm. As it is inefficient to extract all action rules in order to filter a few interesting ones, an algorithm is required that focuses on interesting action rules at the time of extraction, in the mining step. Such an algorithm has been designed, that acts in a novel manner: based on the existing rule schemas and the operations performed by the user, the algorithm generates all candidate rules – all possible rules that may result from applying the operations to the rule schemas – and then checks their support against the database. This method is efficient, because the algorithm acts on a local scale, but provides globally valid results.

The paper is organised as follows. Section 2 presents the research domain and reviews related works. Section 3 discusses action rules and how they are extracted. Section 4 examines AAR discovery. Section 4 shows representative AAR. Section 5 describes the rule schema formalism and the operations on rule schemas. Section 6 presents the algorithm for focusing on interesting AAR starting from rule schemas. Finally, Section 7 presents the conclusion.

2 Background and related work

Generally, the interestingness of rules depends on statistical measures as the support and the confidence in the database (the objective aspect of interestingness), but, more importantly, it depends on the database domain, on user background knowledge and on user expectations (the subjective aspect of interestingness). Two main subjective measures of interest exist: unexpectedness – rules are surprising to the user; and, actionability – the user is able to take action based on his/her discovery.

Paper (Wang et al., 2006) introduced the notion of $action$ as a domain- independent way to model the domain knowledge. Given a dataset about actionable features and an utility measure, a pattern is actionable if it summarises a population that can be acted upon towards a more promising population observed with a higher utility. Algorithms for mining actionable patterns (changes within flexible attributes) take into account only numerical attributes. The distinguished (decision) attribute is called utility. Each action A_i triggers changes of attribute values described by terms $[a \downarrow]$, $[b \uparrow]$, and $[c$ (do not know)]. They are represented as an influence matrix built by an expert. While previous approaches used only features – mined directly from the decision system, authors in Wang et al. (2006) define actions as its foreign concepts. Influence matrix shows the link between actions and changes of attribute values and the same shows correlations between some attributes, i.e., if $[a \downarrow]$, then $[b \uparrow]$. Clearly, expert does not know correlations between classification attributes and the decision attribute. Such correlations can be described as action rules and they have to be discovered from the decision system.

Action rules have been introduced in Raś and Wieczorkowska (2000) and investigated further in Tsay and Raś (2006), Raś et al. (2005, 2007), Raś and Dardzińska (2006), Tzacheva and Raś (2007), Greco et al. (2005) and Qiao et al. (2007) Paper (He et al., 2005) was probably the first attempt towards formally introducing the problem of mining action rules without pre-existing classification rules. Authors explicitly formulated it as a search problem in a support-confidence-cost framework. The proposed algorithm has some similarity with apriori (Agrawal and Srikant, 1994). Their definition of an action rule allows changes on stable attributes. Changing the value of an attribute, either stable or flexible, is linked with a cost (Tzacheva and Raś, 2007).

In order to rule out action rules with undesired changes on attributes, authors designated very high cost to such changes. However, that way, the cost of action rules discovery is getting unnecessarily increased. Also, they did not take into account the correlations between attribute values which are naturally linked with the cost of rules used either to accept or reject a rule.

Algorithm *ARED*, presented in Im and Raś (2008) is based on Pawlak's model of an information system S (Pawlak, 1981). The goal was to identify certain relationships between granules defined by the indiscernibility relation on its objects. Some of these relationships uniquely define action rules for S. Paper (Raś and Dardzińska, 2008) presents a strategy for discovering action rules directly from the decision system. Action rules are built from atomic expressions following a strategy similar to *ERID* (Dardzińska and Raś, 2006).

In earlier works: (Raś and Wieczorkowska, 2000; Tsay and Raś, 2006; Raś et al. 2005, 2007; Raś and Dardzińska, 2006) action rules are constructed from classification rules. This means that we use pre-existing classification rules or generate them using a rule discovery algorithm, such as *LERS* (Grzymala-Busse, 1997) or *ERID* (Dardzińska and Raś, 2006), then, construct action rules either from certain pairs of these rules or from a single classification rule. For instance, algorithm $ARAS$ (Raś et al., 2007) generates sets of terms (built from values of attributes) around classification rules and constructs action rules directly from them.

A later work, Raś et al. (2008) proposes an approach for generating association-type action rules. Authors use apriori-like (Agrawal and Srikant, 1994) strategy for generating frequent action sets. This also allows for extraction of action rules directly from a decision system and without using pre-existing classification rules. Authors also discuss, representative AAR. They, similarly to Kryszkiewicz (1998), form a small subset of AAR, called a covering, from which the remaining AAR are generated – an objective approach.

In this work, we focus on user beliefs, i.e., subjective interestingness. Thus, next we discuss work attempting to reduce the number of classical association rules (Agrawal and Srikant, 1994) through user beliefs.

The concept of rule templates is introduced in Klemettinen et al. (1994), as items in two lists: inclusion – rules that are interesting, and restriction – rules that are not interesting.

Padmanabhan and Tuzhilin (1998) suggests a logical representation and comparison for user beliefs – a fairly limited approach. An important proposition for the representation of user beliefs is presented in Liu et al. (1997) and later developed in Liu et al. (1999). It contains three levels of specification: GI, reasonably precise concepts (RPC), and precise knowledge (PK). All three formalisms use items in a taxonomy, therefore allowing only for is-a relations between items. Moreover, using three different levels might be difficult to use, if the user wants to combine their features. For instance, the user might know that an item A leads to an item B and knowing that item C relates to them but without being sure on what side of the implication it might be. Representing this kind of knowledge is not possible with the formalism given in Liu et al. (1999).

Authors in Olaru et al. (2009) propose incorporating user beliefs through rule schemas with association rule mining. The approach helps the user focus on the search of interesting rules mined locally. The algorithm, thus, avoids extraction of all rules.

The advantages of using ontologies are presented in Phillips and Buchanan (2001). Different manners of integrating ontologies are possible (Nigro et al., 2007), either in postprocessing or in a pre-processing step, for filtering of transactions before mining (Bellandi et al., 2007).

In summary, the related work, consists of the following approaches:

- numerous papers, reducing the space of classical *association rules* through user beliefs (subjective interestingness)
- one paper (Tzacheva, 2009), reducing the space of classical *action rules*, through user beliefs with actions ontology (subjective interestingness)
- one paper (Raś et al., 2008), reducing the space of *association action rules*, through representative rules, or coverings (objective interestingness)

In this work, we propose a new approach – to reduce the number of *AAR* through incorporating user beliefs, i.e., subjective interestingness.

We are not aware of any other work attempting to reduce the number AAR through user beliefs.

3 Action rules

Let $S = (X, A, V)$ is an information system, where $V = \bigcup\{V_a : a \in A\}$. First, we introduce the notion of an atomic action set.

By an *atomic action set* we mean an expression $(a, a_1 \to a_2)$, where a is an attribute and $a_1, a_2 \in V_a$. If $a_1 = a_2$, then a is called stable on a_1. Instead of $(a, a_1 \to a_1)$, we often write (a, a_1) for any $a_1 \in V_a$.

By *action sets* we mean a smallest collection of sets such that:

1 if t is an atomic action set, then t is an action set

2 if t_1, t_2 are action sets and '$\cdot$' is a 2-argument functor called *composition*, then $t_1 \cdot t_2$ is a candidate action set

3 if t is a candidate action set and for any two atomic action sets $(a, a_1 \to a_2)$, $(b, b_1 \to b_2)$ contained in t we have $a \neq b$, then t is an action set.

By the domain of an action set t, denoted by $Dom(t)$, we mean the set of all attribute names listed in t.

By an *action rule* we mean any expression $r = [t_1 \Rightarrow t_2]$, where t_1 and t_2 are action sets. Additionally, we assume that $Dom(t_2) \cup Dom(t_1) \subseteq A$ and $Dom(t_2) \cap Dom(t_1) = \emptyset$. The domain of action rule r is defined as $Dom(t_1) \cup Dom(t_2)$.

Now, we give an example of action rules assuming that the information system S is represented by Table 1, a, c are stable and b, d are flexible attributes. Expressions (a, a_2), $(b, b_1 \to b_2)$, (c, c_2), $(d, d_1 \to d_2)$ are examples of atomic action sets. Expression $(b, b_1 \to b_2)$ means that the value of attribute b is changed from b_1 to b_2. Expression (c, c_2) means that the value c_2 of attribute c remains unchanged. Expression $r = [[(a, a_2) \cdot (b, b_1 \to b_2)] \Rightarrow (d, d_1 \to d_2)]$ is an example of an action rule. The rule says that if value a_2 remains unchanged and value b will change from b_1 to b_2, then it is expected that the value d will change from d_1 to d_2. The domain $Dom(r)$ of action rule r is equal to $\{a, b, d\}$.

Table 1 Information system S

	a	b	c	d
x_1	a_1	b_1	c_1	d_1
x_2	a_2	b_1	c_2	d_1
x_3	a_2	b_2	c_2	d_1
x_4	a_2	b_1	c_1	d_1
x_5	a_2	b_3	c_2	d_1
x_6	a_1	b_1	c_2	d_2
x_7	a_1	b_2	c_2	d_1
x_8	a_1	b_2	c_1	d_3

Standard interpretation N_S of action sets in $S = (X, A, V)$ is defined as follow:

1 If $(a, a_1 \to a_2)$ is an atomic action set, then

$$N_S((a, a_1 \to a_2)) = [\{x \in X : a(x) = a_1\}, \{x \in X : a(x) = a_2\}].$$

2 If $t_1 = (a, a_1 \to a_2) \cdot t$ and $N_S(t) = [Y_1, Y_2]$, then

$$N_S(t_1) = [Y_1 \cap \{x \in X : a(x) = a_1\}, Y_2 \cap \{x \in X : a(x) = a_2\}].$$

Let us define $[Y_1, Y_2] \cap [Z_1, Z_2]$ as $[Y_1 \cap Z_1, Y_2 \cap Z_2]$ and assume that $N_S(t_1) = [Y_1, Y_2]$ and $N_S(t_2) = [Z_1, Z_2]$. Then, $N_S(t_1 \cdot t_2) = N_S(t_1) \cap N_S(t_2)$.

If t is an action set and $N_S(t) = \{Y_1, Y_2\}$, then the support of t in S is defined as $sup(t) = min\{card(Y_1), card(Y_2)\}$.

Now, let $r = [t_1 \Rightarrow t_2]$ is an action rule, where $N_S(t_1) = [Y_1, Y_2]$, $N_S(t_2) = [Z_1, Z_2]$. Support and confidence of r are defined as follow:

$$sup(r) = min\{card(Y_1 \cap Z_1), card(Y_2 \cap Z_2)\}.$$

$$conf(r) = \left[\frac{card(Y_1 \cap Z_1)}{card(Y_1)}\right] \cdot \left[\frac{card(Y_2 \cap Z_2)}{card(Y_2)}\right].$$

The definition of a confidence should be interpreted as an optimistic confidence. It requires that $card(Y_1) \neq 0$ and $card(Y_2) \neq 0$. Otherwise, the confidence of action rule is undefined.

Coming back to the example of S given in Table 1, we can find many action rules associated with S. Let us take $r = [[(a, a_2) \cdot (b, b_1 \to b_2)] \Rightarrow (d, d_1 \to d_2)]$ as an example of action rule. Then,

$$N_S((a, a_2)) = [\{x_2, x_3, x_4, x_5, x_6, x_7\}, \{x_2, x_3, x_4, x_5, x_6, x_7\}],$$

$$N_S((b, b_1 \to b_2)) = [\{x_1, x_2, x_5, x_7\}, \{x_3, x_4, x_6, x_8\}],$$

$$N_S((d, d_1 \to d_2)) = [\{x_1, x_2, x_5, x_8\}, \{x_3, x_4, x_6, x_7\}],$$

$$N_S((a, a_2) \cdot (b, b_1 \to b_2)) = [\{x_2, x_5, x_7\}, \{x_3, x_4, x_6\}].$$

Clearly, $sup(r) = 2$ and $conf(r) = 2/3 \cdot 1 = 2/3$.

Now, let us assume that $S = (X, A, V)$ is an information system and λ_1, λ_2 denote minimum support and minimum confidence assigned to action rules, respectively. The algorithm for constructing frequent action sets is similar to Agrawal's algorithm in Agrawal and Srikant (1994).

3.1 Generating frequent action sets

Let t_a is an atomic action set, where $N_S(t_a) = [Y_1, Y_2]$ and $a \in A$. We say that t_a is called frequent if $card(Y_1) \geq \lambda_1$ and $card(Y_2) \geq \lambda_1$.

The operation of generating $(k+1)$-element candidate action sets from frequent k-element action sets is performed in two steps:

- *Merging step*: Merge pairs (t_1, t_2) of frequent k-element action sets into $(k+1)$-element candidate action set if all elements in t_1 and t_2 are the same except the last elements.
- *Pruning step*: Delete each $(k+1)$-element candidate action set t if either it is not an action set or some k-element subset of f is not a frequent k-element action set.

Now, if t is a $(k+1)$-element candidate action set, $N_S(t) = [Y_1, Y_2]$, $card(Y_1) \geq \lambda_1$, and $card(Y_2) \geq \lambda_1$, then t is a frequent $(k+1)$-element action set.

We say that t is a frequent action set in S if t is a frequent k-element action set in S, for some k. Assume now that the expression $[t - t_1]$ denotes the action set containing all atomic action sets listed in t but not listed in t_1.

Now, we can show how to discover the set $AAR_S(\lambda_1, \lambda_2)$ of AAR from S.

Let t be a frequent action set in S and t_1 is its subset. Any action rule $r = [(t - t_1) \Rightarrow t_1]$ is an AAR in $AAR_S(\lambda_1, \lambda_2)$ if $conf(r) \geq \lambda_2$.

4 Representative AAR

The concept of representative association rules was introduced by Kryszkiewicz (1998). They form a small subset of association rules from which the remaining association rules can be generated. Similar approach was proposed for AAR in Raś et al. (2008).

By a cover C of AAR $r = [t_1 \Rightarrow t]$ we mean $C(t_1 \Rightarrow t) = \{t_1 \cdot t_2 \to t_3 : t_2, t_3$ are not overlapping subterms of $t\}$.

For example, let us assume that $r = [(e, e_1 \to e_2) \Rightarrow (b, b_1 \to b_2) \cdot (c, c_1 \to c_2) \cdot (d, d_1 \to d_2)]$ is an AAR. Then, $[(e, e_1 \to e_2) \cdot (b, b_1 \to b_2) \Rightarrow (c, c_1 \to c_2)] \in C(r)$.

The following fact has been proved in (Raś et al., 2008):

Property 1: If $r \in AAR_S(\lambda_1, \lambda_2)$, then each rule $r_1 \in C(r)$ also belongs to $AAR_S(\lambda_1, \lambda_2)$.

Now, assume that $conf(r) = \frac{card(Y_1 \cap Y_2 \cap Y_3 \cap Y_4)}{card(Y_1)} \cdot \frac{card(Z_1 \cap Z_2 \cap Z_3 \cap Z_4)}{card(Z_1)} \geq \lambda_2$.

Clearly, $\frac{card(Y_1 \cap Y_2 \cap Y_4)}{card(Y_1 \cap Y_2)} \cdot \frac{card(Z_1 \cap Z_2 \cap Z_4)}{card(Z_1 \cap Z_2)} \geq \lambda_2$. The same $conf(r_1) \geq \lambda_2$.

By a set of representative AAR, with minimum support λ_1 and minimum confidence λ_2 we mean

$$\begin{aligned} RAAR_S(\lambda_1, \lambda_2) = \{&r \in AAR_S(\lambda_1, \lambda_2) :\sim (\exists r_1) \\ &\in AAR_S(\lambda_1, \lambda_2)[[r_1 \neq r] \wedge [r \in C(r_1)]\}. \end{aligned}$$

The following two facts have been proved in (Raś et al., 2008):

Property 2: Representative AAR $RAAR_S(\lambda_1, \lambda_2)$ form a least set of association action rules that covers all AAR $AAR_S(\lambda_1, \lambda_2)$.

Property 3: All AAR $AAR_S(\lambda_1, \lambda_2)$ can be derived from representative AAR $RAAR_S(\lambda_1, \lambda_2)$ by means of cover operator.

The process of how to construct representative AAR, from which r can be generated, is given below.

- Procedure I:

1 find t_i in t such that $conf(r_i(t)) \geq \lambda_2$

2 if succeeded, then $t := [t - t_i]$, $s := s \cdot t_i$, go back to (1). Otherwise, procedure stops.

Assume that $[t \Rightarrow s]$ is that rule and $T = \{t_1, t_2, ..., t_m\}$ is a set of all atomic action terms not listed in s.

- Procedure II (it extends the decision part of a rule generated by Procedure I):

1 find t_i in T such that $sup(t \Rightarrow s \cdot t_i) \geq \lambda_1$

2 if succeeded, then $s := s \cdot t_i$, $T := T - \{t_i\}$, go back to (1). Otherwise, procedure stops.

The resulting AAR is a representative rule from which the initial rule r can be generated.

5 Interesting AAR

5.1 Association action rules schema

Focusing on action rules that are interesting to the user means finding action rules that are in a certain relation with his/her current beliefs. This relation can be one of confirmation or one of contradiction. A formalism for representing the user's beliefs and knowledge has been designed: the rule schema is based on the three levels of specification similar to the proposed in Liu et al. (1997), but comprises their advantages into one single level of specification. The association action rule schema (AARS) represents what the user believes about the associative relations among action terms in the database. An AARS is represented as follows:

$$RS([Condition \;\Rightarrow\; Conclusion][General][s\%, c\%]$$

The *Condition* and the *Conclusion* are action terms that the user believes are present in the antecedent and, respectively, in the consequent of the rule. In complement, the *General* is a set of atomic action terms that the user is not sure if and which one exists in the antecedent and which one in the consequent. The *AARS* also contains optional constraints of support and confidence (Olaru et al., 2009).

An example of an *AARS* is given below:

$$[(a, a_2) \cdot (b, b_1 \to b_2)] \Rightarrow [(d, d_1 \to d_2)][(c, c_1)][s\%c\%]$$

As we already stated, the *General* part of an *AARS* is defined as a set of atomic action terms seen as the optional part of an AAR. It means atomic action terms listed in *General* do not have to appear in the rule.

For instance, in a case of the above example, the following AAR can be taken into consideration:

$$[(a, a_2) \cdot (b, b_1 \to b_2)] \Rightarrow [(d, d_1 \to d_2)],$$

$[(a, a_2) \cdot (b, b_1 \to b_2) \cdot (c, c_1)] \Rightarrow [(d, d_1 \to d_2)]$, /specialisation/

$[(a, a_2) \cdot (b, b_1 \to b_2)] \Rightarrow [(c, c_1) \cdot (d, d_1 \to d_2)]$.

This formalism completely covers the three levels of specification presented in Liu et al. (1999). If the *Condition* and *Conclusion* are used, the schema is more like a PRC or PK. If the *General* part is used, the schema is more like a general impression. The improvement is that *AARS* may use these all three parts.

Example: Assume that attribute *a* represents *salary*, attribute *b* represents *bank*, attribute *c* represents *amount*, attribute *d* means *interest percent*, binary attribute *e* is representing *real estate*, and attribute *f* means *bank profit*, where $Dom(f) = \{f_1, f_2, f_3\}$.

Assume the user knows that if her salary remains the same and the name of her bank changes from b_1 to b_2, then the amount on her account remains the same, however the interest percent increases from d_1 to d_2. Assume also, that these changes are associated with Real Estate, however the user does not know whether real estate should be listed on the left side or the right side of the AAR. Then, the corresponding $AARS$ is:

$$[(a, a_2) \cdot (b, b_1 \to b_2)] \to [(c, c_1) \cdot (d, d_1 \to d_2)][(e, yes)]$$

Further, if the user feels that the action $(f, f_1 \to f_2)$ may influence the right side of $AARS$ and also wishes to filter out the output based on the constraints $support = 12$ and $confidence = 80\%$, new elements will be added to the *AARS*, which now becomes:

$$[(a, a_2) \cdot (b, b_1 \to b_2)] \Rightarrow [(c, c_1) \cdot (d, d_1 \to d_2) \cdot (f, f_1 \to f_2)][(e, yes)][12, 80]$$

6 Local AAR mining algorithm

Before we present the (*AAR*) mining algorithm, the notions of specialisation and exception of the rule will be recalled.

The specialisation allows the user to find action rules from the *AARS* by increasing the number of atomic action terms listed in a conditional part and keeping the same conclusion, with the condition that the confidence of the specialised rule must be higher than the confidence of the more general rule. The exception produces AAR with an unexpected conclusion in the context of a more specialised condition. That is, for an action rule of the form $t_1 \Rightarrow (d, d_1 \to d_2)$, exceptions are of the form $[t_1 \cdot t_2] \Rightarrow (d, d_1 \to d_3)$, where t_2 is an action term. We assume here that $d_1, d_2, d_3 \in Dom(d)$ are all different.

The first step of AAR mining algorithm is to generate all candidates for interesting AAR from all *AARS*. For instance, if $[t_1 \Rightarrow t_2][t_3 \cdot t_4]$ is an *AARS* and t_3, t_4 are atomic action terms, then the following candidates will be generated: $[[t_1 \cdot t_3] \Rightarrow t_2]$, $[[t_1 \cdot t_4] \Rightarrow t_2]$, $[[t_1 \cdot t_3 \cdot t_4] \Rightarrow t_2]$, $[t_1 \Rightarrow [t_2 \cdot t_3]]$, $[t_1 \Rightarrow [t_2 \cdot t_4]]$, $[t_1 \Rightarrow [t_2 \cdot t_3 \cdot t_4]]$, $[[t_1 \cdot t_4] \Rightarrow [t_2 \cdot t_3]]$, $[[t_1 \cdot t_3] \Rightarrow [t_2 \cdot t_4]]$.

If additionally t_2 is an atomic action term, then all possible exception action rules produced from *AARS* will be added to the list of candidates. For instance, assuming that $t_2 = (d, d_1 \to d_2)$ and $d_1, d_2, d_3 \in Dom(d)$, then the following action rules will be added to the above list of candidates: $[[t_1 \cdot t_3] \Rightarrow (d, d_1 \to d_3)]$, $[[t_1 \cdot t_4] \Rightarrow (d, d_1 \to d_3)]$, $[[t_1 \cdot t_3 \cdot t_4] \Rightarrow (d, d_1 \to d_2)]$.

The next step of the algorithm is the confirmation of the candidate rules. It is done by checking their confidence and support requirements against the database. Candidate AAR meeting these two requirements are presented to the user.

7 Conclusions

The number of discovered AAR in a database is much larger than the number of association rules so clearly new strategies are needed to have it reduced as much as possible. One way to approach this problem is to discover representative AAR from which all the remaining AAR can be generated. However, the reduction in the number of AAR in most of the cases is not sufficient. In this paper, we assume that users have some expectations and beliefs concerning AAR they like to discover. This assumption leads us to the notion of *AARS* which are used as filter at the AAR discovery process. Rules discovered by filtering strategy are called interesting. By using the same filter at the representative AAR discovery, the number of interesting AAR is reduced further.

References

Agrawal, R. and Srikant, R. (1994) 'Fast algorithm for mining association rules', *Proceeding of the Twentieth International Conference on VLDB*, pp.487–499.

Bellandi, A., Furletti, B., Grossi, V. and Romei, A. (2007) 'Ontologydriven association rule extraction: a case study', *Proceedings of the Workshop – Context and Ontologies: Representation and Reasoning*, pp.1–10.

Blanchard, J., Guillet, F. and Briand, H. (2007) 'Interactive visual exploration of association rules with rule-focusing methodology', *Knowledge and Information Systems*, Vol. 13, pp.43–75.

Dardzińska, A. and Raś, Z. (2006) 'Extracting rules from incomplete decision systems', in *Foundations and Novel Approaches in Data Mining, Studies in Computational Intelligence*, Springer, Vol. 9, pp.143–154.

Duval, B., Salleb, A. and Vrain, C. (2007) 'On the discovery of exception rules: a survey', *Quality Measures in Data Mining*, pp.77–98.

Fayyad, U.M., Piatetsky-Shapiro, G. and Smyth, P. (1996) 'From data mining to knowledge discovery: an overview', *Advances in Knowledge Discovery and Data Mining*, Vol. 1, pp.1–34.

Fensel, D. (1998) *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*, Springer-Verlag.

Greco, S., Matarazzo, B., Pappalardo, N. and Slowiński, R. (2005) 'Measuring expected effects of interventions based on decision rules', *J. Exp. Theor. Artif. Intell.*, Vol. 17, Nos. 1–2, pp.103–118.

Grzymala-Busse, J. (1997) 'A new version of the rule induction system LERS', *Fundamenta Informaticae*, Vol. 31, No. 1, pp.27–39.

He, Z., Xu, X., Deng, S. and Ma, R. (2005) 'Mining action rules from scratch', *Expert Systems with Applications*, Vol. 29, No. 3, pp.691–699.

Im, S. and Raś, Z.W. (2008) 'Action rule extraction from a decision table: ARED', in A. An et al. (Eds.): *Foundations of Intelligent Systems, Proceedings of ISMIS'08*, Vol. 4994, pp.160–168, Toronto, Canada, LNAI, Springer.

Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H. and Verkamo, A.I. (1994) 'Finding interesting rules from large sets of discovered association rules', in *Proceedings of the Third International Conference on Information and Knowledge Management*, pp.401–407.

Kryszkiewicz, M. (1998) 'Representative association rules', in *Proceedings of the Second Pacific-Asia Conference on Research and Development in Knowledge Discovery and Data Mining*, Springer, LNCS, Vol. 1394, pp.198–209.

Liu, B., Hsu, W. and Chen, S. (1997) 'Using general impressions to analyze discovered classification rules', in *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, Vol. 1, pp.31–36.

Liu, B., Hsu, W., Wang, K. and Chen, S. (1999) 'Visually aided exploration of interesting association rules', *PAKDD '99: Proceedings of the Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining*, pp.380–389.

Nigro, H.O., Cisaro, S.G. and Xodo, D. (2007) *Data Mining with Ontologies: Implementations, Findings, and Frameworks*, Chapter Preface, IGI Global, pp.xi–xv.

Olaru, A., Marinica, C. and Guillet, F. (2009) 'Local mining of association rules with rule schemas', in *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2009*, pp.118–124.

Padmanabhan, B. and Tuzhilin, A. (1998) 'A belief-driven method for discovering unexpected patterns', in *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, Vol. 1, pp.94–100.

Pawlak, Z. (1981) 'Information systems – theoretical foundations', *Information Systems Journal*, Vol. 6, pp.205–218.

Phillips, J. and Buchanan, B.G. (2001) 'Ontology-guided knowledge discovery in databases', in *Proceedings of the International Conference on Knowledge Capture*, pp.123–130.

Qiao, Y., Zhong, K., Wang, H-A. and Li, X. (2007) 'Developing event-condition-action rules in real-time active database', *Proceedings of the 2007 ACM Symposium on Applied Computing*, ACM, New York, pp.511–516.

Raś, Z., Wyrzykowska, E. and Wasyluk, H. (2007) 'ARAS: action rules discovery based on agglomerative strategy', in *Mining Complex Data, Post-Proceedings of 2007 ECML/PKDD Third International Workshop (MCD 2007)*, LNAI, Springer, Vol. 4944, pp.196–208.

Raś, Z.W. and Dardzińska, A. (2006) 'Action rules discovery, a new simplified strategy', *Foundations of Intelligent Systems*, LNAI, Springer, Vol. 4203, pp.445–453.

Raś, Z.W. and Dardzińska, A. (2008) 'Action rules discovery without pre-existing classification rules', *Proceedings of the International Conference on Rough Sets and Current Trends in Computing (RSCTC 2008)* LNAI, Springer, Vol. 5306, pp.181–190.

Raś, Z.W. and Wieczorkowska, A. (2000) 'Action-rules: how to increase profit of a company', in *Principles of Data Mining and Knowledge Discovery, Proceedings of PKDD 2000*, Lyon, France, LNAI, Springer, Vol. 1910, pp.587–592.

Raś, Z.W., Dardzińska, A., Tsay, L-S. and Wasyluk, H. (2008) 'Association action rules', in *Proceedings of the 2008 IEEE International Conference on Data Mining Workshops: IEEE/ICDM Workshop on Mining Complex Data (MCD 2008)*, in Pisa, Italy, *IEEE Computer Society*, pp.283–290.

Raś, Z.W., Tzacheva, A., Tsay, L-S. and Gürdal, O. (2005) 'Mining for interesting action rules', *Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2005)*, Compiegne University of Technology, France, pp.187–193.

Tsay, L-S. and Raś, Z.W. (2006) 'Action rules discovery system DEAR3', in *Foundations of Intelligent Systems, Proceedings of ISMIS 2006*, Bari, Italy, LNAI, Springer, Vol. 4203, pp.483–492.

Tzacheva, A.A. (2009) 'Action rule discovery in ontology based information system', *Journal of the Technical University at Plovdiv, Fundamental Sciences and Applications*, Vol. 14, Plovdiv, Bulgaria.

Tzacheva, A.A. and Raś, Z.W. (2007) 'Constraint based action rule discovery with single classification rules', in *Proceedings of the Joint Rough Sets Symposium (JRS07)*, LNAI, Springer, Vol. 4482, pp.322–329.

Wang, K., Jiang, Y. and Tuzhilin, A. (2006) 'Mining actionable patterns by role models', in *Proceedings of the 22nd International Conference on Data Engineering, IEEE Computer Society*, pp.16–25.