Association Action Rules

Zbigniew W. Raś
University of North Carolina
Dept. of Computer Science
Charlotte, NC, 28223, USA
ras@uncc.edu

Agnieszka Dardzińska
Bialystok Technical University
Dept. of Computer Science
15-351 Bialystok, Poland
adardzin@uncc.edu

Li-Shiang Tsay
North Carolina A&T State University
School of Technology
Greensboro, NC 27441, USA
ltsay@ncat.edu

Hanna Wasyluk
Medical Center of Postgraduate Education
ul. Marymoncka 99
01-815 Warsaw, Poland
hwasyluk@cmkp.edu.pl

Abstract

Action rules describe possible transitions of objects from one state to another with respect to a distinguished attribute. Previous research on action rule discovery usually required the extraction of classification rules before constructing any action rule. In [19], a new algorithm that discovers action rules directly from a decision system was presented. This paper gives a new approach for generating association-type action rules. The notion of frequent action sets and Apriori-like strategy generating them is proposed. We introduce the notion of a representative action rules and give an algorithm to construct them directly from frequent action sets. Finally, we introduce the notion of a simple association action rule, the cost of association action rule, and give a strategy to construct simple association action rules of a lowest cost.

1. Introduction

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 [21]. We assume that 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. Action rules mining initially was based on comparing profiles of two groups of targeted objects - those that are desirable and those that are undesirable [21]. An action rule was 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 need to be changed so the undesirable objects can be 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 introduced in [21] has been further investigated in [18][19][20][22][26][27]. Paper [8] 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 algorithm proposed by them has some similarity with Apriori [1]. The definition of an action rule in [8] allows changes on stable attributes. Changing the value of an attribute, either stable or flexible, is linked with a cost [27]. In order to rule out action rules with undesired changes on stable attributes, authors assigned 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 dependencies between attribute values which are naturally linked with the cost of rules used either to accept or reject a rule.

Algorithm *ARED*, presented in [11], is based on Pawlak′s model of an information system S [16]. Its 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. Papers [19], [11] present a new strategy for discovering action rules directly from the decision system. In [19], action rules are built from atomic expressions following a strategy similar to *ERID* [3].

This paper gives a new approach for generating association-type action rules. The notion of frequent action sets and Apriori-like strategy generating them is proposed. Finally, we introduce the notion of a representative action rules and give an algorithm to construct them directly from frequent action sets.

2. Background and Objectives

In this section we introduce the notion of an information system and give its example.

By an information system [16] we mean a triple $S = (X, A, V)$, where:

1. X is a nonempty, finite set of objects
2. A is a nonempty, finite set of attributes, i.e. $a : X \longrightarrow V_a$ is a function for any $a \in A$, where V_a is called the domain of a
3. $V = \bigcup\{V_a : a \in A\}$.

For example, Table 1 shows an information system S with a set of objects $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$, set of attributes $A = \{a, b, c, d\}$, and a set of their values $V = \{a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2\}$.

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

Table 1 : Information System S

Additionally, we assume that $A = A_{St} \cup A_{Fl}$, where attributes in A_{St} are called *stable* and attributes in A_{Fl} are called *flexible*. "Date of birth" is an example of a stable attribute. "Interest rate" for each customer account is an example of a flexible attribute.

In earlier works in [20][21][22][26], 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* [7] or *ERID* [3], then, construct action rules either from certain pairs of these rules or from a single classification rule. For instance, algorithm $ARAS$ [22] generates sets of terms (built from values of attributes) around classification rules and constructs action rules directly from them. In this study, we propose a different approach to achieve the following objectives:

1. Extract action rules directly from a decision system without using pre-existing classification rules.
2. Extract action rules that have minimal attribute involvement.

To meet these two goals, we introduce the notion of action sets, frequent action sets, and show how to build action rules from them.

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 $(a, a_1 \rightarrow a_2)$, 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 $(a, a_1 \rightarrow a_1)$, 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 $(a, a_1 \rightarrow a_2)$, $(b, b_1 \rightarrow b_2)$ 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), $(b, b_1 \rightarrow b_2)$, (c, c_2), $(d, d_1 \rightarrow d_2)$ are examples of atomic action sets. Expression $(b, b_1 \rightarrow b_2)$ 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 = $[[(a, a_2) \cdot (b, b_1 \rightarrow b_2)] \Rightarrow (d, d_1 \rightarrow d_2)]$ 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 $\{a, b, d\}$.

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

1. If $(a, a_1 \rightarrow a_2)$ is an atomic action set, then $N_S((a, a_1 \rightarrow a_2)) = [\{x \in X : a(x) = a_1\}, \{x \in X : a(x) = a_2\}]$.

2. If $t_1 = (a, a_1 \rightarrow a_2) \cdot t$ 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:

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

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

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 = $[[(a, a_2) \cdot (b, b_1 \rightarrow b_2)] \Rightarrow (d, d_1 \rightarrow d_2)]$ as an example of action rule. Then,

$N_S((a, a_2))$=
$[\{x_2, x_3, x_4, x_5, x_6, x_7\}, \{x_2, x_3, x_4, x_5, x_6, x_7\}]$,
$N_S((b, b_1 \rightarrow b_2)) =$
$[\{x_1, x_2, x_5, x_7\}, \{x_3, x_4, x_6, x_8\}]$,
$N_S((d, d_1 \rightarrow d_2))$=
$[\{x_1, x_2, x_5, x_8\}, \{x_3, x_4, x_6, x_7\}]$,
$N_S((a, a_2) \cdot (b, b_1 \rightarrow b_2))$=
$[\{x_2, x_5, x_7\}, \{x_3, x_4, x_6\}]$.

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 [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.

The set $AAR_S(\lambda_1, \lambda_2)$ of association action rules in S is constructed in the following way:

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 association action rule in $AAR_S(\lambda_1, \lambda_2)$ if $conf(r) \geq \lambda_2$.

4. Association Action Rules, an Example

Let us assume that an information system S is represented by Table 1 with $\{a, c\}$ as stable attributes. We take $\lambda_1 = 2$ and $\lambda_2 = 4/9$. The following frequent action sets can be constructed:

(a, a_1) - support 2
(a, a_2) - support 6
(b, b_1) - support 4
(b, b_2) - support 4
$(b, b_1 \rightarrow b_2)$ - support 4
$(b, b_2 \rightarrow b_1)$ - support 4
(c, c_1) - support 5
(c, c_2) - support 3
(d, d_1) - support 4
(d, d_2) - support 4
$(d, d_1 \rightarrow d_2)$ - support 4
$(d, d_2 \rightarrow d_1)$ - support 4

$(a, a_1) \cdot (b, b_1)$ - support 1 (not frequent)
$(a, a_1) \cdot (b, b_2)$ - support 1 (not frequent)
$(a, a_1) \cdot (b, b_1 \to b_2)$ - support 1 (not frequent)
$(a, a_1) \cdot (b, b_2 \to b_1)$ - support 1 (not frequent)
$(a, a_1) \cdot (c, c_1)$ - support 1 (not frequent)
$(a, a_1) \cdot (c, c_2)$ - support 1 (not frequent)
$(a, a_1) \cdot (d, d_1)$ - support 2
$(a, a_1) \cdot (d, d_2)$ - support 0 (not frequent)
$(a, a_1) \cdot (d, d_1 \to d_2)$ - support 0 (not frequent)
$(a, a_1) \cdot (d, d_2 \to d_1)$ - support 0 (not frequent)

$(a, a_2) \cdot (b, b_1)$ - support 3
$(a, a_2) \cdot (b, b_2)$ - support 3
$(a, a_2) \cdot (b, b_1 \to b_2)$ - support 3
$(a, a_2) \cdot (b, b_2 \to b_1)$ - support 3
$(a, a_2) \cdot (c, c_1)$ - support 4
$(a, a_2) \cdot (c, c_2)$ - support 2
$(a, a_2) \cdot (d, d_1)$ - support 2
$(a, a_2) \cdot (d, d_2)$ - support 4
$(a, a_2) \cdot (d, d_2 \to d_1)$ - support 2
$(a, a_2) \cdot (d, d_1 \to d_2)$ - support 2

$(b, b_1) \cdot (c, c_1)$ - support 3
$(b, b_1) \cdot (c, c_2)$ - support 1 (not frequent)
$(b, b_1) \cdot (d, d_1)$ - support 3
$(b, b_1) \cdot (d, d_2)$ - support 1 (not frequent)

$(b, b_2) \cdot (c, c_1)$ - support 2
$(b, b_2) \cdot (c, c_2)$ - support 2
$(b, b_2) \cdot (d, d_1)$ - support 1 (not frequent)
$(b, b_2) \cdot (d, d_2)$ - support 3

$(b, b_1 \to b_2) \cdot (c, c_1)$ - support 2
$(b, b_1 \to b_2) \cdot (c, c_2)$ - support 1 (not frequent)
$(b, b_1 \to b_2) \cdot (d, d_1)$ - support 1 (not frequent)
$(b, b_1 \to b_2) \cdot (d, d_2)$ - support 1 (not frequent)
$(b, b_1 \to b_2) \cdot (d, d_2 \to d_1)$ - support 1 (not frequent)
$(b, b_1 \to b_2) \cdot (d, d_1 \to d_2)$ - support 3

$(b, b_2 \to b_1) \cdot (c, c_1)$ - support 2
$(b, b_2 \to b_1) \cdot (c, c_2)$ - support 1 (not frequent)
$(b, b_2 \to b_1) \cdot (d, d_1)$ - support 1 (not frequent)
$(b, b_2 \to b_1) \cdot (d, d_1 \to d_2)$ - support 1 (not frequent)
$(b, b_2 \to b_1) \cdot (d, d_2 \to d_1)$ - support 3

$(c, c_1) \cdot (d, d_1)$ - support 3
$(c, c_1) \cdot (d, d_2)$ - support 2
$(c, c_1) \cdot (d, d_2 \to d_1)$ - support 2
$(c, c_1) \cdot (d, d_1 \to d_2)$ - support 2
$(c, c_2) \cdot (d, d_1)$ - support 1 (not frequent)
$(c, c_2) \cdot (d, d_2)$ - support 2
$(c, c_2) \cdot (d, d_2 \to d_1)$ - support 1 (not frequent)
$(c, c_2) \cdot (d, d_1 \to d_2)$ - support 1 (not frequent)

$(a, a_2) \cdot (b, b_1) \cdot (c, c_1)$ - support 2
$(a, a_2) \cdot (b, b_1) \cdot (c, c_2)$ - support 1 (not frequent)
$(a, a_2) \cdot (b, b_1) \cdot (d, d_1)$ - support 2
$(a, a_2) \cdot (b, b_1) \cdot (d, d_2)$ - support 1 (not frequent)
$(a, a_2) \cdot (b, b_1) \cdot (d, d_1 \to d_2)$ - support 1 (not frequent)
$(a, a_2) \cdot (b, b_2) \cdot (c, c_1)$ - support 2
$(a, a_2) \cdot (b, b_2) \cdot (c, c_2)$ - support 1 (not frequent)
$(a, a_2) \cdot (b, b_2) \cdot (d, d_1)$ - support 0 (not frequent)
$(a, a_2) \cdot (b, b_2) \cdot (d, d_2)$ - support 3
$(a, a_2) \cdot (b, b_2) \cdot (d, d_1 \to d_2)$ - support 0 (not frequent)
$(a, a_2) \cdot (b, b_1 \to b_2) \cdot (c, c_1)$ - support 2
$(a, a_2) \cdot (b, b_1 \to b_2) \cdot (c, c_2)$ - support 1 (not frequent)
$(a, a_2) \cdot (b, b_1 \to b_2) \cdot (d, d_1)$ - support 0 (not frequent)
$(a, a_2) \cdot (b, b_1 \to b_2) \cdot (d, d_2)$ - support 1 (not frequent)
$(a, a_2) \cdot (b, b_1 \to b_2) \cdot (d, d_1 \to d_2)$ - support 2
$(a, a_2) \cdot (b, b_2 \to b_1) \cdot (c, c_1)$ - support 2
$(a, a_2) \cdot (b, b_2 \to b_1) \cdot (c, c_2)$ - support 1 (not frequent)
$(a, a_2) \cdot (b, b_2 \to b_1) \cdot (d, d_1)$ - support 0 (not frequent)
$(a, a_2) \cdot (b, b_2 \to b_1) \cdot (d, d_2)$ - support 1 (not frequent)
$(a, a_2) \cdot (b, b_2 \to b_1) \cdot (d, d_1 \to d_2)$ - support 0 (not frequent)
....
....
....

$(a, a_2) \cdot (b, b_1 \to b_2) \cdot (c, c_1) \cdot (d, d_1 \to d_2)$ - support 2

Association action rules can be constructed from frequent action sets. For instance, we can generate association action rule

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

from the last frequent action set. Its confidence is 4/9.

5 Representative Association Action Rules

The concept of representative association rules was introduced by Kryszkiewicz [10]. They form a small subset of association rules from which the remaining association rules can be generated. Similar approach is proposed for association action rules in this paper.

By a cover C of association action rule $r = [t_1 \Rightarrow t]$ 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 association action rule. Then, $[(e, e_1 \to e_2) \cdot (b, b_1 \to b_2) \Rightarrow (c, c_1 \to c_2)] \in C(r)$.

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

Proof: From the definition of $AAR_S(\lambda_1, \lambda_2)$ we have, $sup(r) \geq \lambda_1$, and $conf(r) \geq \lambda_2$. Let $r_1 = [t_1 \cdot t_2 \rightarrow t_4]$, $r = [t_1 \rightarrow t_2 \cdot t_3 \cdot t_4]$, and $N_S(t_i) = [Y_i, Z_i]$, for $i = 1, 2, 3, 4$. Now, since $\frac{card[Y_1 \cap Y_2 \cap Y_3 \cap Y_4]}{card[Y_1]} \geq \lambda_1$, then $\frac{card[Y_1 \cap Y_2 \cap Y_4]}{card[Y_1 \cap Y_2]} \geq \lambda_1$ because $card(Y_1) \geq card(Y_1 \cap Y_2)$ and $card(Y_1 \cap Y_2 \cap Y_4) \geq card(Y_1 \cap Y_2 \cap Y_3 \cap Y_4)$.

In a similar way we show that $\frac{card[Z_1 \cap Z_2 \cap Z_4]}{card[Z_1 \cap Z_2]} \geq \lambda_1$. The same, $sup(r_1) \geq \lambda_1$.

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 association action rules, with minimum support λ_1 and minimum confidence λ_2 we mean

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

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

Proof: Let us assume that $r \in RAAR_S(\lambda_1, \lambda_2)$ and there exists $r_1 = [t_1 \Rightarrow t] \in AAR_S(\lambda_1, \lambda_2)$ such that $r_1 \neq r$ and $r \in C(r_1)$. Now, since $r \in C(r_1)$, then r is not in $RAAR_S(\lambda_1, \lambda_2)$.

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

Proof: Assume that $r = [t \Rightarrow s] \in AAR_S(\lambda_1, \lambda_2)$ and $t = t_1 \cdot t_2 \cdot ... \cdot t_k$, where t_i is an atomic action set for $1 \leq i \leq k$. It means that $conf(r) \geq \lambda_2$ and $sup(r) \geq \lambda_1$. Let $r_i(t) = [[t - t_i] \Rightarrow s \cdot t_i]$ for any atomic action set t_i in t. Clearly, $sup(r_i(t)) = sup(r)$ and $conf(r_i(t)) \leq conf(r)$.

Now, we show how to construct representative association action rule from which r can be generated. First, we follow 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.

Procedure II will extend the decision part of the rule generated by Procedure I. 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:

(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.

Now, the resulting association action rule is a representative rule from which the initial rule r can be generated.

6 Simple Association Action Rules

In this section we introduce the notion of a simple association action rule, the cost of association action rule, and give a strategy to construct simple association action rules of lowest cost.

Let $(a, a_1 \rightarrow a_2)$ is an atomic action set. We assume that the cost of changing attribute a from a_1 to a_2 is denoted by $cost_S((a, a_1 \rightarrow a_2))$ [27]. For simplicity reason, the subscript S will be omitted if this does not lead to a confusion. Let $t_1 = (a, a_1 \rightarrow a_2)$, $t_2 = (b, b_1 \rightarrow b_2)$ be two atomic action sets. We say that t_1, t_2 are positively correlated if change t_1 supports change t_2. Saying another words, change t_1 implies change t_2.

Now, assume that action set t is constructed from atomic action sets $T = \{t_1, t_2, ..., t_m\}$. We introduce a binary relation $\simeq$ on T defined as: $t_i \simeq t_j$ iff t_i and t_j are positively correlated.

Relation $\simeq$ is an equivalence relation and it partitions T into m equivalence classes ($T = T_1 \cup T_2 \cup ... \cup T_m$), for some m. Now, in each equivalence class T_i, an atomic action set $a(T_i)$ of the lowest cost is identified. The cost of t is defined as: $cost(t) = \sum\{cost(a(T_i)) : 1 \leq i \leq m\}$.

Now, assume that $r = [t_1 \Rightarrow t]$ is an association action rule. We say that r is simple if $cost(t_1 \cdot t) = cost(t_1)$. The cost of r is defined as $cost(t_1)$.

We assume that user gives three threshold values, λ_1 - minimum support, λ_2 - minimum confidence, λ_3 - maximum cost. Let t be a frequent action set in S and t_1 is its subset. Any association action rule $r \in AAR_S(\lambda_1, \lambda_2)$ is called association action rule of acceptable cost if $cost(r) \leq \lambda_3$. Similarly, frequent action set t is called a frequent action set of acceptable cost if $cost(t) \leq \lambda_3$.

Now, in order to construct simple association action rules of a lowest cost, we built frequent action sets of acceptable cost following the strategy presented in Section 4 enhanced by additional constraint which requires to verify the cost of frequent action sets being produced. Any frequent action set which cost is higher than λ_3, is removed. Now, if t is a frequent action set of acceptable cost and $\{a(T_i) : i \leq m\}$ is a collection of atomic action sets constructed by following the strategy presented in this section, then $\Pi\{a(T_i) : i \leq m\} \Rightarrow [t - \{a(T_i) : i \leq m\}]$ is a simple association action rule of acceptable cost assuming that its confidence is not greater than λ_2.

7. Application Domain - *HEPAR* Database

As the application domain for action rules mining, we have chosen the *HEPAR* system built in collaboration between the Institute of Biocybernetics and Biomedical Engineering of the Polish Academy of Sciences and physicians at the Medical Center of Postgraduate Education in Warsaw, Poland [2], [25], [15]. *HEPAR* was designed for gathering and processing clinical data about patients with liver disorders. It contains information on 758 patients described by 106 attributes (including 31 laboratory tests with values discretized to: below normal, normal, above normal). It has 14 stable attributes. Two laboratory tests are invasive: HBsAg [in tissue] and HBcAg [in tissue]. The decision attribute has 7 values: *I* (acute hepatitis), *IIa* (subacute hepatitis [types B and C]), *IIb* (subacute hepatitis [alcohol-abuse]), *IIIa* (chronic hepatitis [curable]), *IIIb* (chronic hepatitis [non-curable]), *IV* (cirrhosis-hepatitis), *V* (liver-cancer).

The diagnosis of liver disease depends on a combination of patient's history, physical examinations, laboratory tests, radiological tests, and frequently a liver biopsy. Blood tests play an important role in the diagnosis of liver diseases. However, their results should be analyzed along with the patient's history and physical examination. The most common radiological examinations used in the assessment of liver diseases are ultrasound and sonography. Ultrasound is a good test for the detection of liver masses, assessment of bile ducts, and detection of gallstones presence. However, it does not detect the degree of inflammation or fibrosis of the liver. Ultrasound is a noninvasive procedure and there are no risks associated with it. Liver biopsy enables the doctor to examine how much inflammation and how much scarring has occurred. Liver biopsy is an example of invasive procedure that carries certain risks to the patient. Therefore, despite of the importance of its results to the diagnosis, clinicians try to avoid biopsy as often as possible. However, liver biopsy is often the only way to establish correct diagnosis in patients with chronic liver disorders.

A medical treatment is naturally associated with reclassification of patients from one decision class into another one. In this research we were mainly interested in the re-classification of patients from the class *IIb* into class *I* and from the class IIIa into class I but without referring to any invasive tests results in action rules.

Database HEPAR has many missing values. Following the approach proposed in [22], we removed all its attributes with more than 90% of null values assuming that these attributes are not related to invasive tests. Also, subjective attributes (like *history of alcohol abuse*) have been removed. Next, we used one of the classical null value imputation techniques (provided in Rough Sets Exploration System (RSES)) to make the resulting database complete.

For the testing purpose, we have chosen the same d-reduct *R* = {*m, n, q, u, y, aa, ah, ai, am, an, aw, bb, bg, bm, by, cj, cm*} as in [22] because it does not contain any invasive tests. By d-reduct we mean a minimal subset of attributes which by itself can fully characterize the knowledge about attribute d in the database. The description of attributes (tests) listed in R is given below:

- m - Bleeding
- n - Subjaundice symptoms
- q - Eructation
- u - Obstruction
- y - Weight loss
- aa - Smoking
- ah - History of viral hepatitis (stable)
- ai - Surgeries in the past (stable)
- am - History of hospitalization (stable)
- an - Jaundice in pregnancy
- aw - Erythematous dermatitis
- bb - Cysts
- bg - Sharp liver edge (stable)
- bm - Blood cell plaque
- by - Alkaline phosphatase
- cj - Prothrombin index
- cm - Total cholesterol
- dd - Decision attribute

Many action rules have been discovered. Three of them had a very high confidence (close to 100) and they are given below:

$$[(am, 2) \cdot (ah, 2) \cdot (bg, 2)] \cdot (q, 2 \rightarrow 1) \cdot (cm, 2 \rightarrow 1) \Rightarrow (dd, IIIA \rightarrow I)$$

The first rule is applicable to patients with a history of hospitalization, history of viral hepatitis, and with a sharp liver edge which is not normal. It says that if we get rid of eructation and decrease the cholesterol level to normal, then we should be able to reclassify such patients from the category *IIIA* to *I*.

$$[(am, 2) \cdot (bg, 2) \cdot (ai, 1)] \cdot (u, 2 \rightarrow 1) \cdot (y, 2 \rightarrow 1) \Rightarrow (dd, IIIA \rightarrow I)$$

The second rule is applicable to patients with a history of hospitalization, with a sharp liver edge which is not normal, and with no surgeries in the past. It says that if we get rid of obstruction and change the weight loss level to normal, then we should be able to reclassify such patients from the category *IIIA* to *I*.

$[(am, 2) \cdot (bg, 2) \cdot (ai, 1)] \cdot (q, 2 \rightarrow 1) \cdot (u, 2 \rightarrow 1) \cdot (n, 2 \rightarrow 1) \Rightarrow (dd, IIIA \rightarrow I)$

The third rule is applicable to patients with a history of hospitalization, with a sharp liver edge which is not normal, and with no surgeries in the past. It says that if we get rid of eructation, get rid of obstruction, and change the subjaundice symptoms to normal, then we should be able to reclassify such patients from the category *IIIA* to *I*.

8. Conclusion and Future Work

Action rules mining can be successfully applied in many other areas including Business and Music Information Retrieval (MIR). In [23], authors present the system *MIRAI* for automatic indexing of music by instruments and emotions. When *MIRAI* receives a musical waveform, it divides that waveform into segments of equal size and then its classifiers identify the most dominating musical instruments and emotions associated with each segment and also with the musical waveform. In [12], [13] authors follow another approach and present a Basic Score Classification Database (*BSCD*) which describes associations between different scales, regions, genres, and jumps. This database is used to automatically index a piece of music by emotions. In [19], it was shown how to use action rules extracted from *BSCD* to modify the retrieved piece of music when we need to change the emotions it invokes in people. Action rules show how it can be achieved with a minimal change in the associated music score. By a score, in MIR area, we mean a written form of a musical composition. Representative association action rules form the smallest subset of association action rules which can be extracted from *BSCD* database from which the remaining association action rules can be generated by cover operator. The same, they can be interpreted as compact representations of classes of action rules.

9. Acknowledgment

This material is based in part upon work supported by the Ministry of Science and Higher Education in Poland under Grant *N N519 404734*, supported by the Bialystok Technical University under Grant Numbers *W/WI/15/07, S/WI/1/08*, and partially supported by the National Science Foundation under Grant Number *IIS-0414815*. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] R. Agrawal, R. Srikant. Fast algorithm for mining association rules. *Proceeding of the Twentieth International Conference on VLDB*, 1994, 487-499.

[2] L. Bobrowski. HEPAR: Computer system for diagnosis support and data analysis. *Prace IBIB 31*, Institute of Biocybernetics and Biomedical Engeneering, Polish Academy of Sciences, Warsaw, Poland, 1992.

[3] A. Dardzińska, Z. Raś. Extracting rules from incomplete decision systems. *Foundations and Novel Approaches in Data Mining*, Studies in Computational Intelligence, Springer, Vol. 9, 2006, 1994, 143-154.

[4] D. Fensel. *Ontologies: a silver bullet for knowledge management and electronic commerce*. Springer, 1998.

[5] H. Geffner, J. Wainer. Modeling action, knowledge and control. *ECAI*, 1998, 532-536.

[6] S. Greco, B. Matarazzo, N. Pappalardo, R. Słowiński. Measuring expected effects of interventions based on decision rules. *J. Exp. Theor. Artif. Intell.*, Taylor and Francis, Vol. 17, No. 1-2, 2005, 103-118.

[7] J. Grzymala-Busse. A new version of the rule induction system LERS. *Fundamenta Informaticae*, IOS-Press, Vol. 31, No. 1, 1997, 27-39.

[8] Z. He, X. Xu, S. Deng, R. Ma. Mining action rules from scratch. *Expert Systems with Applications*, Elsevier, Vol. 29, No. 3, 2005, 691-699.

[9] S. Hettich, C.L. Blake, C.J. Merz. UCI Repository of machine learning databases. *http://www.ics.uci.edu/∼mlearn/MLRepository.html*, University of California, Irvine, 1998.

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

[11] S. Im, Z.W. Raś. Action rule extraction from a decision table: ARED. *Foundations of Intelligent Systems, Proceedings of ISMIS'08, A. An et al. (Eds.)*, Springer, LNCS, Vol. 4994, 2008, 160-168.

[12] R. Lewis, Z.W. Raś. Rules for processing and manipulating scalar music theory. *Proceedings of the International Conference on Multimedia and Ubiquitous Engineering (MUE 2007) in Seoul, South Korea*, IEEE Computer Society, 2007, 819-824.

[13] R. Lewis, W. Jiang, Z.W. Raś. Mining scalar representations in a non-tagged music database. *Foundations of Intelligent Systems, Proceedings of ISMIS'08, A. An et al. (Eds.)*, Springer, LNAI, Vol. 4994, 2008, 445-454.

[14] A. Øhrn, J. Komorowski. ROSETTA: A Rough Set Toolkit for Analysis of Data. *Proc. Third International Joint Conference on Information Sciences*, 1997, 403-407.

[15] A. Oniśko, M. Druzdzel, H. Wasyluk. Extension of the HEPAR II model to multiple-disorder diagnosis. *Intelligent Information Systems*, Advances in Soft Computing, Springer, 2000, 303-313.

[16] Z. Pawlak. Information systems - theoretical foundations. *Information Systems Journal*, Elsevier, Vol. 6, 1981, 205-218.

[17] Y. Qiao, K. Zhong, H.-A. Wang and X. Li. Developing event-condition-action rules in real-time active database. *Proceedings of the 2007 ACM Symposium on Applied Computing*, ACM, 2007, 511-516.

[18] Z.W. Raś, A. Dardzińska. Action rules discovery, a new simplified strategy. *Foundations of Intelligent Systems, Proceedings of ISMIS'06 Symposium*, Springer, LNAI, Vol. 4203, 2006, 445-453.

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

[20] Z.W. Raś, A. Tzacheva, L.-S. Tsay, O. Gürdal. Mining for interesting action rules. *Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2005)*, IEEE Computer Society, 2005, 187-193.

[21] Z.W. Raś, A. Wieczorkowska. Action-Rules: How to increase profit of a company. *Proceedings of PKDD 2000*, Springer, LNAI, Vol. 1910, 2000, 587-592.

[22] Z. Raś, E. Wyrzykowska, H. Wasyluk. ARAS: Action rules discovery based on agglomerative strategy. *Mining Complex Data, Post-Proceedings of 2007 ECML/PKDD Third International Workshop (MCD 2007)*, Springer, LNAI, Vol. 4944, 2008, 196-208.

[23] Z.W. Ras, X. Zhang, R. Lewis. MIRAI: Multi-hierarchical, FS-tree based music information retrieval system. *Proceedings of RSEISP 2007*, Springer, LNAI, Vol. 4585, 2007, 80-89.

[24] A. Skowron. Rough sets and boolean reasoning. *Granular Computing: an Emerging Paradigm*, Physica-Verlag, 2001, 95-124.

[25] H. Wasyluk, Z.W. Raś, E. Wyrzykowska. Application of Action Rules to HEPAR Clinical Decision Support System. *Experimental and Clinical Hepatology*, Vol. 4, No. 2, 2008, 46-48.

[26] L.-S. Tsay, Z.W. Raś. Action rules discovery system DEAR3. *Foundations of Intelligent Systems, Proceedings of ISMIS 2006*, Springer, LNAI, Vol. 4203, 2006, 483-492.

[27] A. Tzacheva, Z.W. Raś. Constraint based action rule discovery with single classification rules. *Proceedings of the Joint Rough Sets Symposium (JRS07)*, Springer, LNAI, Vol. 4482, 2007, 322-329.