Association Action Rules and Action Paths Triggered by Meta-Actions

Angelina A. Tzacheva
Univ. of South Carolina Upstate, Dept. of Informatics
Spartanburg, SC 29303, USA
ATzacheva@uscupstate.edu

Zbigniew W. Raś
Univ. of North Carolina, Dept. of Computer Science
Charlotte, NC, 28223, USA
and
Polish Academy of Sciences, Institute of Computer Science
01-237 Warsaw, Poland
ras@uncc.edu

Abstract

Action rules are built from atomic expressions called atomic action terms and they describe possible transitions of objects from one state to another. They involve changes of values within one decision attribute. Association action rule is similar to an action rule but it may refer to changes of values involving several attributes listed in its decision part. Action paths are defined as sequences of association action rules with the assumption that the last rule in a sequence is as action rule. Early research on action rule discovery usually required the extraction of classification rules before constructing any action rule. Newest algorithms discover action rules and association action rules directly from an information system. This paper presents a strategy for generating association action rules and action paths by incorporating the use of meta-actions and influence matrices. Action paths show the cascading effect of meta-actions leading to a desired goal.

1 Introduction

An association action rule is a rule extracted from an information system that describes a cascading effect of changes of attribute values listed on the left-hand side of a rule [12] on changes of attribute values listed on its right-hand side. Similarly to [13, 12], we assume that attributes used to describe objects are partitioned into stable and flexible. Values of flexible attributes can be changed whereas values of stable attributes have to remain the same.

Action rules mining initially required to compare profiles of two groups of targeted objects - those that are desirable and those that are undesirable [13]. 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 a hint of how values of some attributes need to be changed so the effect of moving the undesirable objects into a desirable group can be achieved.

In most cases it is not possible to change the values of flexible attributes without having them triggered by some higher-level actions. For instance, we can not lower the temperature of a patient if he does not take a drug used for that purpose. Taking aspirin is an example of a higher-level action which should trigger such a change. In this paper, such higher-level actions are called *meta-actions*. The associations between meta-actions and changes of attribute values they trigger can be modeled using either an influence matrix [19] or ontology [3].

Action rules have been introduced in [13] and investigated further in [17, 14, 10, 18, 15, 4, 9]. Paper [6] 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 is similar to Apriori [1]. Their definition of an action rule allows changes within stable attributes. Clearly, changing the value of an attribute is naturally linked with some cost [18]. In medical area, cost may refer to how expensive is the patient care but also may refer to its safety. In [6], authors used the cost measure for entirely different purpose - to rule out action rules involving changes of stable attribute values by assigning to them very high cost. However, they did not take into account correlations between attributes which may either decrease or increase the cost of rules.

Algorithm *ARED*, presented in [7], is based on Pawlak′s model of an information system S [8]. 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 [11] 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* [2].

Paper [19] introduced the notion of $action$ as a domain-independent way to model the domain knowledge. Given a data set about actionable features and utility measure, a pattern is actionable if it summarizes 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 (don't 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 [19] 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. In [16], changes of attribute values and action rules are also described using terms [$a \downarrow$], [$b \uparrow$].

Authors in [19] did not take into consideration stable attributes. Also, their attributes are only numerical. In this paper, for simplicity reason, we use only symbolic attributes. Numerical attributes, if any, are discretized before association rules or action rules are discovered.

2 Background and Objectives

In this section we introduce the notion of an information system, atomic action term, meta-action, and give examples.

By an information system [8] 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. $a : U \longrightarrow V_a$ 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 $V = \{a_1, a_2, b_1, b_2, b_3, c_1, c_2, d_1, d_2, d_3\}$.

	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

Table 1 : Information System S

An information system $S = (X, A, V)$ is called a decision system, if one of the attributes in A is distinguished and called the decision. The remaining attributes in A are classification attributes. Additionally, we assume that $A = A_{St} \cup A_{Fl} \cup \{d\}$, where attributes in A_{St} are called *stable* and in A_{Fl} *flexible*. Attribute d is the decision attribute. "Place of birth" is an example of a stable attribute. "Marital status" is an example of an attribute which can be either stable or flexible.

By atomic action term t associated with S, we mean any expression $t = (a, a_1 \rightarrow a_2)$, where $a_1, a_2 \in V_a$, $a \in A$. If $a_1 = a_2$, then we will also write (a, a_1) instead of $(a, a_1 \rightarrow a_2)$. Object $x \in X$ satisfying the description $a(x) = a_1$ belongs to the domain of t if there is a way of changing its attribute value a_1 to a_2.

By meta-actions associated with S we mean higher concepts used to model certain generalizations of actions introduced in [19]. Meta-actions, when executed, trigger changes in values of some flexible attributes in S as described by influence matrix [19] and atomic action terms. To give an example, let us assume that classification attributes in S describe teaching evaluations at some school and the decision attribute represents their overall score. *Explain difficult concepts effectively*, *Speaks English fluently*, *Stimulate student interest in the course*, *Provide sufficient feedback* are examples of classification attributes. Then, examples of meta-actions associated with S will be: *Change the content of the course*, *Change the textbook of the course*, *Post all material on the Web*. Clearly, any of these three meta-actions will not influence the attribute *Speaks English fluently* and the same its values will remain unchanged

[19]. Let us take Hepatitis as the application domain. Then *increase blood cell plague* and *decrease level of alkaline phosphatase* are examples of an atomic action term. Drugs like *Hepatil* or *Hepargen* can be seen as meta-actions triggering changes described by these two atomic action terms. It should be noted that *Hepatil* is also used to get rid of obstruction, eructation, and bleeding. However, *Hepargen* is not used to get rid of obstruction but it is used to get rid of eructation and bleeding.

Also, it should be mentioned here that an expert knowledge concerning meta-actions involves only classification attributes. Now, if some of these attributes are correlated with the decision attributes (the ones listed on the right-hand side of an association action rule), then the change of their values will cascade to the decision through the correlation. The goal of action rule discovery is to identify possibly all such correlations.

In earlier works in [13, 17, 14, 10, 15], 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* [5] or *ERID* [2] (if the data have probabilistic type), then, construct action rules either from certain pairs of these rules or from a single classification rule. For instance, algorithm $ARAS$ [15] generates sets of terms (built from values of attributes) around classification rules and constructs action rules directly from them. In [12] authors presented a strategy for extracting action rules directly from a decision system. Similarly, authors in [12] presented a strategy for discovering association action rules directly from an information system.

In the next section, we recall the notion of action terms, action rules, association rules [12], and the notion of an influence matrix (see [19]) associated with a set of meta-actions. The values stored in an influence matrix are atomic action terms. We also introduce the notion of an action path.

3 Action Rules, Action Paths, and Meta-Actions

Let $S = (X, A, V)$ be an information system, where $V = \bigcup\{V_a : a \in A\}$. First, we recall the notion of an atomic action term [11].

By an *atomic action term* 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.

By *action terms* [11] we mean a smallest collection of atomic action terms such that:

1. If t is an atomic action term, then t is an action term.
2. If t_1, t_2 are action terms, then $t_1 \cdot t_2$ is a candidate action term.
3. If t is a candidate action term and for any two atomic action terms $(a, a_1 \rightarrow a_2)$, $(b, b_1 \rightarrow b_2)$ contained in t we have $a \neq b$, then t is an action term.

By the domain of an action term t, denoted by $Dom(t)$, we mean the set of all attribute names listed in t. For instance, if $(a, a_2) \cdot (b, b_1 \rightarrow b_2)$ is an action term, then $Dom(t) = \{a, b\}$. Assume now that

$\{(a, a_2), (b, b_1 \rightarrow b_2)\}$, $\{(a, a_2), (c, c_1 \rightarrow c_2)\}$

are two collections of atomic action terms. Meta-action M_1 can trigger the changes represented by $(a, a_2) \cdot (b, b_1 \rightarrow b_2)$ and M_2 can trigger the changes represented by $(a, a_2) \cdot (c, c_1 \rightarrow c_2)$. It means that attribute a in both cases has to remain stable.

Consider several meta-actions, denoted M_1, M_2,...,M_n. Each one can invoke changes within values of some classification attributes in A. We assume here that $A - \{d\} = A_1 \cup A_2 \cup ... \cup A_m$. The expected changes of classification attribute values on objects from S which are triggered by these meta-actions are described by the influence matrix $\{E_{i,j}\}$, $1 \leq i \leq n$, $1 \leq j \leq m$. Table 2 gives an example of an influence matrix associated with 6 meta-actions and attributes a, b, and c.

	a	b	c
M_1		b_1	$c_2 \rightarrow c_1$
M_2	$a_2 \rightarrow a_1$	b_2	
M_3	$a_1 \rightarrow a_2$		$c_2 \rightarrow c_1$
M_4		b_1	$c_1 \rightarrow c_2$
M_5			$c_1 \rightarrow c_2$
M_6	$a_1 \rightarrow a_2$		$c_1 \rightarrow c_2$

Table 2: Influence Matrix associated with S

For instance, let us take meta-action M_2. It says that by executing M_2 on objects in S, two atomic action terms are triggered. They are: $(a_2 \rightarrow a_1)$ and $(b, b_2 \rightarrow b_2)$. It means that objects in S satisfying the description $(a, a_2) \cdot (b, b_2)$ are expected to change their description to $(a, a_1) \cdot (b, b_2)$. Also, it should be noticed that any influence matrix does not have to be associated with a decision system but only with an information system.

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

Now, we give an example of an association action rule assuming that the information system S is represented by Table 1, attributes a, c, d are flexible and b is stable. Expressions (a, a_2), (b, b_2), $(c, c_1 \rightarrow c_2)$, $(d, d_1 \rightarrow d_2)$ are examples of atomic action terms. Expression $r = [[(a, a_2) \cdot (c, c_1 \rightarrow c_2)] \Rightarrow (d, d_1 \rightarrow d_2)]$ is an example of an association action rule which is also an action rule. The rule says that if value a_2 for objects in S remains unchanged and the value of c will change from c_1 to c_2, then it is expected that value d_1 for all these objects will change to d_2. The domain $Dom(r)$ of action rule r is equal to $\{a, c, d\}$.

By an action path we mean a sequence $[t_1 \Rightarrow t_2 \Rightarrow ... \Rightarrow t_n]$, where t_i is an action term for any $1 \leq i \leq n-1$, and t_n is an atomic action term. Additionally, we assume that: $(\forall i \leq n-1)(\exists t_{i1}, t_{i2}, t_{i3}, t_{i4})[[t_i = t_{i1} \cdot t_{i2} \cdot t_{i3}] \wedge [t_{i+1} = t_{i1} \cdot t_{i4} \cdot t_{i3}] \wedge [(t_{i2} \Rightarrow t_{i4})$ is an action rule extracted from S]].

For instance:
$[(b_1, v_1 \rightarrow w_1) \wedge (b_2, v_2 \rightarrow w_2) \wedge ... \wedge (b_p, v_p \rightarrow w_p)] \Rightarrow [(b_1, v_1 \rightarrow w_1) \wedge ... \wedge [(b_{j1}, v_{j1} \rightarrow w_{j1}) \wedge (b_{j2}, v_{j2} \rightarrow w_{j2}) \wedge ... \wedge (b_{jq}, v_{jq} \rightarrow w_{jq})] \wedge ... \wedge (b_p, v_p \rightarrow w_p)] \Rightarrow (d, k_1 \rightarrow k_2)$ is an action path if $[(b_{j1}, v_{j1} \rightarrow w_{j1}) \wedge (b_{j2}, v_{j2} \rightarrow w_{j2}) \wedge ... \wedge (b_{jq}, v_{jq} \rightarrow w_{jq})] \Rightarrow (b_j, v_j \rightarrow w_j)$ is an action rule.

To justify the purpose of introducing the notion of an action path, we take as an example the medical domain. The medical state of a patient is represented by values of attributes including results of medical tests. Action terms represent possible changes within values of some of these attributes which can be triggered by drugs, consultations with doctors or surgeries. At the same time, the description of a medical state of a patient based on these attributes may not be sufficient to identify any disease and the same what set of drugs should be prescribed to the patient. In such cases, the doctor still may decide to prescribe some drugs with a goal to move the patient to a new medical state which hopefully will be less complex in terms of stating the medical diagnosis. The purpose of introducing the notion of an action path is to model this kind of reasoning which also describes the steps of a medical treatment.

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

1. If $(a, a_1 \rightarrow a_2)$ is an atomic action term, 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 association action rule 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 association action rule, where $N_S(t_1) = [Y_1, Y_2]$, $N_S(t_2) = [Z_1, Z_2]$. Support and confidence of r is 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 requires that $card(Y_1) \neq 0$ and $card(Y_2) \neq 0$. Otherwise, the confidence of an association action rule is undefined.

Coming back to the example of S given in Table 1, we can find a number of association action rules associated with S. Let us take $r = [[(b, b_1) \cdot (c, c_1 \rightarrow c_2)] \Rightarrow (d, d_1 \rightarrow d_2)]$ as an example of association action rule. Then,

$N_S((b, b_1)) = [\{x_1, x_2, x_4, x_6\}, \{x_1, x_2, x_4, x_6\}]$,
$N_S((c, c_1 \rightarrow c_2)) = [\{x_1, x_4, x_8\}, \{x_2, x_3, x_5, x_6, x_7\}]$,
$N_S((d, d_1 \rightarrow d_2)) = [\{x_1, x_2, x_3, x_4, x_5, x_7\}, \{x_6\}]$,
$N_S((b, b_1) \cdot (c, c_1 \rightarrow c_2)) = [\{x_1, x_4\}, \{x_2, x_6\}]$.

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

4 Candidate Action Rules Discovery

The strategy for discovering candidate action rules is similar to the one presented in [11], to the algorithm *ERID* [2], and algorithm *LERS* [5]. It is outlined below.

Assume that $L([Y, Z]) = Y$ and $R([Y, Z]) = Z$. Now, we recall the strategy for assigning marks to atomic action terms and show how action terms are built. Only positive marks yield candidate action rules. Action terms of length k are built from unmarked action terms of length $k-1$ and unmarked atomic action terms of length one. Marking strategy for terms of any length is the same as for action terms of length one.

Assume that $S = (X, A \cup \{d\}, V)$ is a decision system and λ_1, λ_1 denote minimum support and confidence, respectively. Each $a \in A$ uniquely defines the set $C_S(a) = \{N_S(t_a) : t_a$ *is an atomic action term built from elements in* $V_a\}$. By t_d we mean an atomic action term built from elements in V_d.

Marking strategy for atomic action terms

For each $N_S(t_a) \in C_S(a)$ **do**

if $L(N_S(t_a)) = \emptyset$ or $R(N_S(t_a)) = \emptyset$ or $L(N_S(t_a \cdot t_d)) = \emptyset$ or $R(N_S(t_a \cdot t_d)) = \emptyset$, **then** t_a is **marked negative**.

if $L(N_S(t_a)) = R(N_S(t_a))$ **then** t_a **stays unmarked**

if $card(L(N_S(t_a \cdot t_d))) < \lambda_1$ **then** t_a is **marked negative**

if $card(L(N_S(t_a \cdot t_d))) \geq \lambda_1$ and $conf(t_a \rightarrow t_d) < \lambda_2$ **then** t_a **stays unmarked**

if $card(L(N_S(t_a \cdot t_d))) \geq \lambda_1$ and $conf(t_a \rightarrow t_d) \geq \lambda_2$ **then** t_a is **marked positive** and the action rule $[t_a \rightarrow t_d]$ is printed.

Now, to clarify *ARD* (Action Rules Discovery) strategy for constructing candidate action rules, we go back to our example with S defined by Table 1 and with $A_{St} = \{b\}$, $A_{Fl} = \{a, c, d\}$. We are interested in candidate action rules which may reclassify objects from the decision class d_1 to d_2. Also, we assume that $\lambda_1 = 2$, $\lambda_2 = 1/4$.

All atomic action terms for S are listed below:

For decision attribute in S: $(d, d_1 \rightarrow d_2)$

For classification attributes in S:

$(b, b_1 \rightarrow b_1)$, $(b, b_2 \rightarrow b_2)$, $(b, b_3 \rightarrow b_3)$, $(a, a_1 \rightarrow a_2)$, $(a, a_1 \rightarrow a_1)$, $(a, a_2 \rightarrow a_2)$,

$(a, a_2 \rightarrow a_1)$, $(c, c_1 \rightarrow c_2)$, $(c, c_2 \rightarrow c_1)$, $(c, c_1 \rightarrow c_1)$, $(c, c_2 \rightarrow c_2)$, $(d, d_1 \rightarrow d_2)$.

Following the first loop of *ARD* algorithm we get:

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

$N_S((b, b_1 \rightarrow b_1)) = [\{x_1, x_2, x_4, x_6\}, \{x_1, x_2, x_4, x_6\}]$
Not Marked /$Y_1 = Y_2$/

$N_S((b, b_2 \rightarrow b_2)) = [\{x_3, x_7, x_8\}, \{x_3, x_7, x_8\}]$
Marked "-" /$card(Y_2 \cap Z_2) = 0$/

$N_S((b, b_3 \rightarrow b_3)) = [\{x_5\}, \{x_5\}]$
Marked "-" /$card(Y_2 \cap Z_2) = 0$/

$N_S((a, a_1 \rightarrow a_2)) =$
$[\{x_1, x_6, x_7, x_8\}, \{x_2, x_3, x_4, x_5\}]$
Marked "-" /$card(Y_2 \cap Z_2) = 0$/

$N_S((a, a_1 \rightarrow a_1)) =$
$[\{x_1, x_6, x_7, x_8\}, \{x_1, x_6, x_7, x_8\}]$
Not Marked /$Y_1 = Y_2$/

$N_S((a, a_2 \rightarrow a_2)) =$
$[\{x_2, x_3, x_4, x_5\}, \{x_2, x_3, x_4, x_5\}]$
Marked "-" /$card(Y_2 \cap Z_2) = 0$/

$N_S((a, a_2 \rightarrow a_1)) =$
$[\{x_2, x_3, x_4, x_5\}, \{x_1, x_6, x_7, x_8\}]$
Marked "+"
/rule $r_1 = [(a, a_2 \rightarrow a_1) \Rightarrow (d, d_1 \rightarrow d_2)]$ has $conf = 1/2 \geq \lambda_2$, $sup = 2 \geq \lambda_1$/

$N_S((c, c_1 \rightarrow c_2)) = [\{x_1, x_4, x_8\}, \{x_2, x_3, x_5, x_6, x_7\}]$
Not Marked
/rule $r_1 = [(c, c_1 \rightarrow c_2) \Rightarrow (d, d_1 \rightarrow d_2)]$ has $conf = [2/3] \cdot [1/5] < \lambda_2$, $sup = 2 \geq \lambda_1$/

$N_S((c, c_2 \rightarrow c_1)) = [\{x_2, x_3, x_5, x_6, x_7\}, \{x_1, x_4, x_8\}]$
Marked "-" /$card(Y_2 \cap Z_2) = 0$/

$N_S((c, c_1 \rightarrow c_1)) = [\{x_1, x_4, x_8\}, \{x_1, x_4, x_8\}]$
Marked "-" /$card(Y_2 \cap Z_2) = 0$/

$N_S((c, c_2 \rightarrow c_2)) =$
$[\{x_2, x_3, x_5, x_6, x_7\}, \{x_2, x_3, x_5, x_6, x_7\}]$
Not Marked /$Y_1 = Y_2$/

We build action terms of length two from unmarked action terms of length one.

$N_S((b, b_1 \rightarrow b_1) \cdot (a, a_1 \rightarrow a_1)) = [\{x_1, x_6\}, \{x_1, x_6\}]$
Not Marked /$Y_1 = Y_2$/

$N_S((b, b_1 \rightarrow b_1) \cdot (c, c_1 \rightarrow c_2)) = [\{x_1, x_4\}, \{x_2, x_6\}]$
Marked "+"
/rule $r_1 = [[(b, b_1 \rightarrow b_1) \cdot (c, c_1 \rightarrow c_2)] \Rightarrow (d, d_1 \rightarrow d_2)]$ has $conf = 1/2 \geq \lambda_2$, $sup = 2 \geq \lambda_1$/

$N_S((b, b_1 \rightarrow b_1) \cdot (c, c_2 \rightarrow c_2)) = [\{x_2, x_6\}, \{x_2, x_6\}]$
Not Marked /$Y_1 = Y_2$/

$N_S((a, a_1 \rightarrow a_1) \cdot (c, c_1 \rightarrow c_2)) = [\{x_1, x_8\}, \{x_6, x_7\}]$
Marked "-"
/rule $r_1 = [[(a, a_1 \rightarrow a_1) \cdot (c, c_1 \rightarrow c_2)] \Rightarrow (d, d_1 \rightarrow d_2)]$ has $conf = 1/2 \geq \lambda_2$, $sup = 1 < \lambda_1$/

$N_S((a, a_1 \rightarrow a_1) \cdot (c, c_2 \rightarrow c_2)) = [\{x_6, x_7\}, \{x_6, x_7\}]$
Not Marked /$Y_1 = Y_2$/

$N_S((c, c_1 \rightarrow c_2) \cdot (c, c_2 \rightarrow c_2))$
$= [\emptyset, \{x_2, x_3, x_5, x_6, x_7\}]$
Marked "-" /$card(Y_1) = 0$/

Finally (there are only 3 classification attributes in S), we build action terms of length three from unmarked action terms of length one and length two.

Only, the term $(b, b_1 \rightarrow b_1) \cdot (a, a_1 \rightarrow a_1) \cdot (c, c_1 \rightarrow c_2)$ can be built. It is an extension of $(a, a_1 \rightarrow a_1) \cdot (c, c_1 \rightarrow c_2)$ which is already marked as negative. So, the algorithm *ARD* stops and two candidate action rules are constructed:

$[[(b, b_1 \rightarrow b_1) \cdot (c, c_1 \rightarrow c_2)] \Rightarrow (d, d_1 \rightarrow d_2)]$,
$[(a, a_2 \rightarrow a_1) \Rightarrow (d, d_1 \rightarrow d_2)]$.

Following the notation used in previous papers on action rules mining (see [7], [15], [13], [10]), the first of the above two candidate action rules will be presented as $[[(b, b_1) \cdot (c, c_1 \rightarrow c_2)] \Rightarrow (d, d_1 \rightarrow d_2)]$.

The strategy for discovering candidate association action rules is similar to [12].

5 Discovering Action Rules and Action Paths

Influence matrix associated with S is used to identify which candidate association action rules and which action paths are valid with respect to meta-actions and hidden correlations between classification attributes and decision attributes.

Assume that $S = (X, A, V)$ is an information system, $A = \{A_1, A_2, ..., A_m\}$, $\{M_1, M_2, ..., M_n\}$ are meta-actions associated with S, $\{E_{i,j} : 1 \le i \le n, 1 \le j \le m\}$ is an influence matrix, and $r = [(A_{[i,1]}, a_{[i,1]} \rightarrow a_{[j,1]}) \cdot (A_{[i,2]}, a_{[i,2]} \rightarrow a_{[j,2]}) \cdot ..... \cdot (A_{[i,k]}, a_{[i,k]} \rightarrow a_{[j,k]})] \Rightarrow [(A_{[i,k+1]}, a_{[i,k+1]} \rightarrow a_{[j,k+1]}) \cdot (A_{[i,k+2]}, a_{[i,k+2]} \rightarrow a_{[j,k+2]}) \cdot ..... \cdot (A_{[i,p]}, a_{[i,p]} \rightarrow a_{[j,p]})]$ is a candidate association action rule extracted from S, where $A_{[i,1]}, A_{[i,2]}, ..., A_{[i,k]}, A_{[i,k+1]}, A_{[i,k+2]}, ..., A_{[i,p]} \in A$. We assume that $A_{[i,j]}(M_i) = E_{i,j}$ for any $1 \le i \le n, 1 \le j \le m$. Value $E_{i,j}$ is either an atomic action term or *NULL* (not defined). By meta-actions based information system, we mean a triple consisting with S, the set of meta-actions associated with S, and the influence matrix linking them.

We say that r is valid in S with respect to meta-action M_i, if the following condition holds:

if $(\exists p \le k)[A_{[i,p]}(M_i)$ is defined], then
$(\forall p \le k)[$ if $A_{[i,p]}(M_i)$ is defined, then
$(A_{[i,p]}, a_{[i,p]} \rightarrow a_{[j,p]}) = (A_{[i,p]}, E_{i,p})]$

We say that r is valid in S with respect to the set of meta-actions M, where $M \subseteq \{M_1, M_2, ..., M_n\}$, if r is valid in S with respect to each meta-action in M.

We say that the set of meta-actions M covers association action rule $r = [(A_{[i,1]}, a_{[i,1]} \rightarrow a_{[j,1]}) \cdot (A_{[i,2]}, a_{[i,2]} \rightarrow a_{[j,2]}) \cdot ..... \cdot (A_{[i,k]}, a_{[i,k]} \rightarrow a_{[j,k]})] \Rightarrow [(A_{[i,k+1]}, a_{[i,k+1]} \rightarrow a_{[j,k+1]}) \cdot (A_{[i,k+2]}, a_{[i,k+2]} \rightarrow a_{[j,k+2]}) \cdot ..... \cdot (A_{[i,p]}, a_{[i,p]} \rightarrow a_{[j,p]})]$ in S if r is valid in S with respect to the set of meta-actions M and

$(\forall p \le k)[(\exists M_i \in M)[M_i(A_{[i,p]}) = a_{[i,p]} \wedge r$ is valid in S with respect to M_i.

To give an example, assume that S is an information system represented by Table 1 and $\{M_1, M_2, M_3, M_4, M_5, M_6\}$ is the set of meta-actions assigned to S with an influence matrix shown in Table 3. Any empty slot in Table 3 represents *NULL* value.

	a	b	c
M_1		b_1	$c_2 \rightarrow c_1$
M_2	$a_2 \rightarrow a_1$	b_2	
M_3	$a_1 \rightarrow a_2$		$c_2 \rightarrow c_1$
M_4		b_1	$c_1 \rightarrow c_2$
M_5			$c_1 \rightarrow c_2$
M_6	$a_1 \rightarrow a_2$	b_1	

Table 3: Influence Matrix associated with S

In the previous section we explained the strategy for discovering candidate association action rules. Two such rules have been constructed:

$r_1 = [[(b, b_1) \cdot (c, c_1 \rightarrow c_2)] \Rightarrow (d, d_1 \rightarrow d_2)]$ and
$r_2 = [(a, a_2 \rightarrow a_1) \Rightarrow (d, d_1 \rightarrow d_2)]$.

Clearly r_1 is valid in S with respect to M_4, M_5, and M_6. Action rule r_2 is valid in S with respect to M_1, M_2, M_4, and M_5. The meta-action M_4 covers rule r_1. Also the set of meta-actions $\{M_5, M_6\}$ covers that rule.

Assume that $S = (X, A \cup \{d\}, V)$ is an information system with meta-actions $M = \{M_1, M_2, ..., M_n\}$ associated with S. Any candidate association action rule extracted from S which is covered by a subset of meta-actions in M is called association action rule. In such a case we will also say that the validation process for a candidate association action rule was positive.

Assume now that R is the set of candidate association action rules generated by the algorithm presented in the previous section. The last step in the process of association action rules discovery is used to identify which rules in R are covered by some meta-actions from M.

In our example, r_1 is an association action rule because the meta-action M_4 covers it. Rule r_1 is also applicable to x_1 and x_4. So, the meta-action M_4 will trigger r_1 which in turn will generate two new tuples: y_1 as the result of its application to x_1 and y_2 as the result of its application to x_4. The resulting information system is of type λ (see [2]) and it is given below:

	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
y_1	a_1	b_1	c_2	$(d_2, 1/2)$
y_4	a_2	b_1	c_2	$(d_2, 1/2)$

Table 4 : Information System S_1.

New candidate association action rules can be extracted from S_1, using algorithm similar to the one presented in [12], and next their validity is verified by meta-actions and the corresponding influence matrix associated with S_1. Now, if any new association action rules are extracted, then S_1 will be updated again and the process will continue till the fix point is reached (information system is not changed).

The validation process for action paths with respect to meta-actions and hidden correlations between classification attributes and the decision attribute is similar to the validation process for association action rules. Just, we have to verify if action rules used to define each path are valid with respect to meta-actions and correlations between classification and decision attributes.

6 Conclusion

We have introduced a meta-action based information system which is as a triple $(S, \{M_i : i \leq n\}, \{E_{i,j} : i \leq n, j \leq m\})$, where M_i are meta-actions associated with S, and $\{E_{i,j} : i \leq n, j \leq m\}$ is an influence matrix linking them. Meta-actions jointly with the influence matrix are used as a postprocessing tool in association action rules discovery. Influence matrix shows the correlations among attributes triggered off by meta-actions. If the candidate association actions rules are not on par with them, then they are not classified as association action rules. However, if the influence matrix does not show all the interactions between classification attributes, then still some of the resulting association action rules may fail when tested on real data.

7 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* and by AMVIS Program CONTACT in Czech Republic under Grant *ME 913*.

References

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

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

[3] D. Fensel (1998), Ontologies: a silver bullet for knowledge management and electronic commerce, Springer-Verlag

[4] S. Greco, B. Matarazzo, N. Pappalardo, R. Slowiński (2005), Measuring expected effects of interventions based on decision rules, J. Exp. Theor. Artif. Intell., Vol. 17, No. 1-2, 103-118

[5] J. Grzymala-Busse (1997), A new version of the rule induction system LERS, Fundamenta Informaticae, Vol. 31, No. 1, 27-39

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

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

[8] Z. Pawlak (1981) Information systems - theoretical foundations, Information Systems Journal, Vol. 6, 205-218

[9] Y. Qiao, K. Zhong, H.-A. Wang and X. Li (2007), Developing event-condition-action rules in real-time active database, Proceedings of the 2007 ACM symposium on Applied computing, ACM, New York, 511-516

[10] Z.W. Raś, A. Dardzińska (2006), Action rules discovery, a new simplified strategy, Foundations of Intelligent Systems, LNAI, No. 4203, Springer, 445-453

[11] Z.W. Raś, A. Dardzińska (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 5306, Springer, 181-190

[12] Z.W. Raś, A. Dardzińska, L.-S. Tsay, H. Wasyluk (2008), Association Action Rules, IEEE/ICDM Workshop on Mining Complex Data (MCD 2008), Pisa, Italy, ICDM Workshops Proceedings, IEEE Computer Society, 2008, 283-290

[13] Z.W. Raś, A. Wieczorkowska (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, No. 1910, Springer, 587-592

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

[15] Z. Raś, E. Wyrzykowska, H. Wasyluk (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, Vol. 4944, Springer, 2008, 196-208

[16] J. Rauch (2010), Considerations on Logical Calculi for Dealing with Knowledge in Data Mining, Advances in Data Management, Z. Ras, A. Dardzinska (Eds.), Studies in Computational Intelligence, Vol. 223, Springer, 2009, 177-199

[17] L.-S. Tsay, Z.W. Raś (2006), Action rules discovery system DEAR3, in Foundations of Intelligent Systems, Proceedings of ISMIS 2006, Bari, Italy, LNAI, No. 4203, Springer, 483-492

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

[19] K. Wang, Y. Jiang, A. Tuzhilin (2006), Mining Actionable Patterns by Role Models, in Proceedings of the 22nd International Conference on Data Engineering, IEEE Computer Society, 16-25