# Support Confidence and Utility of Action Rules Triggered by Meta-Actions

Angelina A. Tzacheva, Chandra C. Sankar, Sridharan Ramachandran, and Ramya A. Shankar
*Department of Computer Science*
*University of North Carolina at Charlotte*
*Charlotte, NC, 28223, USA*
*Email: aatzache@uncc.edu, csankar1@uncc.edu, sramac11@uncc.edu, rshanka7@uncc.edu*

***Abstract*—Action rules describe possible transitions of objects from one state to another with respect to a distinguished attribute. Early research on action rule discovery usually required the extraction of classification rules before constructing any action rule. Newest algorithms discover action rules directly from a decision system. We employ a pruning step in action rule generation, through the use of meta-actions. They are nodes of higher-level knowledge, linked with atomic action terms, which show changes triggered within classification attributes. In this paper, we propose improved measures for support and confidence of action rules, as well as we introduce a new measure - the notion of *utility* of action rules. We perform an experiment in medical domain using Mammographic Mass dataset, where action rules suggest possible ways to re-classify breast tumors from malignant to benign severity class. Results show increased support and confidence for the new proposed measures compared to the standard measures.**

***Keywords*-action rules; support; confidence; utility; mammography**

## I. 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 [19]. 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 [19]. An action rule was defined as a term $[(\omega) \wedge (\alpha \rightarrow \beta)] \Rightarrow (\phi \rightarrow \psi)$, where $\omega$ 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. How to identify an action which triggers the desired changes of flexible attributes and which is not described by values of attributes listed in the decision system is a difficult problem. In this paper, such actions are called *meta-actions*. There is a link between meta-actions and the changes they trigger within the values of flexible attributes in the decision system. Such link can be provided either by an ontology [3] or by a mapping linking meta-actions with changes of attributes values used in the decision system. 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. Ontology [3], if it is available, should help us to identify a meta-action which trigger these changes. We allow users to specify certain tresholds associate with action rule mining. In particular, the minimum values for: $support$, $confidence$, and a new measure called $utility$. The action rules, which do not meet the minimum treshold requirements are discarded. The action rules algorithm can be applied in any domain including medical, financial, industrial, and transportaion. In this study, we perform an experiment in medical domain using a Mammographic Mass Dataset. The extracted action rules suggest ways to re-classify breast tumors from malignant to benign severity class.

## II. Related Work

Action rules have been introduced in [19] and investigated further in [21], [18], [13], [22], [20], [5], and [12]. Paper [9] 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 [1]. 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 [22]. 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 [10], is based on Pawlak′s model of an information system $S$ [11]. 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 [14] 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 [24] introduced the notion of $action$ as a domain-independent way to model the domain knowledge. Given a data set about actionable features and an 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 [24] 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 [$b \uparrow$]. Domain experts may not know whether any correlations exist between classification attributes and the decision attribute. Therefore, such correlations are typically not taken into consideration. Although, it is possible for correlations to be discovered from the decision system, and presented in the form of action rules. Authors in [24] did not take into consideration stable attributes and their classification attributes are only numerical.

Ras and Gupta [16] were the first to mention the measures of support and confidence of action rules mining. Although they did not specify a formal definition of support and confidence. They said the confidence $k$ can be calculated based on the objects, which support properties of the pair of two classification rules, from which the action rule is composed.

Ras and Tsay [17] were the first to attempt to specify a definition of support and confidence for what they call extended action rules, or system DEAR. The defintion they proposed is based on the classical confidence mesaure for association rules. Authors took the confidence for the first association rule, and multipled it by the confidence of the second association rule, used in the formation of the action rule. In their paper action rules are composed of a pair of two association rules, having a decision attribute on the right side. For support of the action rule they took the support of the first association rule, or the left side of the action rule. Paper [21] uses the same support and confidence measures.

Later works, such as Tzacheva and Ras [23] employ a more advanced formula for definition of support and confidence of action rules. The support takes the minimum of cardinality of sets of left side of action rule or cardinality of sets of right side of action rule. The confidence is computed as the support of left side of action rule divided by the cardinality of set of objects supporting decision attribute of left side, multiplied with the same for right side of action rule. Same difinitions are used in more recent works such as Hajja et. al. in [8] and [7].

Table I
INFORMATION SYSTEM $S$

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

However, these formulas are too complex for computation. Also, they are too restrictive, meaning we may obtain few action rules of the desired support and confidence for the particular decision attribute of interest. In addition, the formulas are either incorrect or undefined in the case of action rules extracted directly from the database without pair of association (classification) rules, or extracted based on a decision schema.

In this paper, we propose improved measures for support and confidence of action rules, as well as we introduce a new measure - the notion of $utility$ of action rules. We experiment with Mammographic Mass Dataset, extracting action rules which suggest ways to re-classify breast tumors from benign to malignant severity class.

## III. BACKGROUND

In this section we introduce the notion of an information system and meta-actions and give examples.

By an information system [11] 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 I. 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 $V = \{a_1, a_2, b_1, b_2, b_3, c_1, c_2, d_1, d_2, d_3\}$.

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. "Date of birth" is an example of a stable attribute. "Interest rate" for each customer account is an example of a flexible attribute.

By meta-actions associated with $S$ we mean higher level concepts medeling certain generalizations of actions introduced in [24]. Meta-actions, when executed, can trigger changes in values of some flexible attributes in $S$ described by influence matrix [24]. 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. 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 attribute, 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 [19][21][18][13][20], 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* [6] or *ERID* [2], then, construct action rules either from certain pairs of these rules or from a single classification rule. For instance, algorithm $ARAS$ [20] generates sets of terms (built from values of attributes) around classification rules and constructs action rules directly from them. In [15] authors presented a strategy for extracting action rules directly from a decision system and without using pre-existing classification rules.

In the next section, we recall the notion of action sets, action rules [15], and the notion of an influence matrix (see [24]) associated with a set of meta-actions. The values stored in an influence matrix are action sets.

## IV. ACTION RULES AND META-ACTIONS

Let $S = (X, A, V)$ is an information system, where $V = \bigcup\{V_a : a \in A\}$. First, we recall the notion of an atomic action set [14].

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* [14] 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, 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 $t$. For instance, assume that $\{(a, a_2), (b, b_1 \rightarrow b_2)\}$, $\{(a, a_2), (b, b_2 \rightarrow b_1)\}$ are two collections of atomic action sets associated with meta-actions $MA_1$, $MA_2$. It means that both $MA_1$, $MA_2$ can influence attributes $a, b$ but attribute $a$ in both cases has to remain stable. The corresponding action sets are: $(a, a_2) \cdot (b, b_1 \rightarrow b_2)$, $(a, a_2) \cdot (b, b_2 \rightarrow b_1)$.

Consider several meta-actions, denoted $M_1$, $M_2$,...,$M_n$. An action can influence the values of classification attributes in $A$. We assume here that $A - \{d\} = A_1 \cup A_2 \cup ... \cup A_m$. The influence of these meta-actions on classification attributes in $A$ is specified by the influence matrix $\{E_{i,j}\}$, $1 \leq i \leq n$, $1 \leq j \leq m$.

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 I., $a$, $c$, $d$ are flexible attributes 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 sets. Expression $(c, c_1 \rightarrow c_2)$ means that the value of attribute $c$ is changed from $c_1$ to $c_2$. Expression $(a, a_2)$ means that the value $a_2$ of attribute $a$ remains unchanged. Expression $r = [[(a, a_2) \cdot (c, c_1 \rightarrow c_2)] \Rightarrow (d, d_1 \rightarrow d_2)]$ is an example of an action rule. The rule says that if value $a_2$ remains unchanged and value $c$ will change from $c_1$ to $c_2$, 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 $\{a, c, d\}$.

## V. CANDIDATE ACTION RULES DISCOVERY

In this section we show the process of discovering candidate action rules.

Assume that $L([Y, Z]) = Y$ and $R([Y, Z]) = Z$. The algorithm *ARD* [14] for constructing candidate action rules is similar to *ERID* [2] and *LERS* [6]. Now, we will outline the strategy for assigning marks to atomic action terms and show how terms of length greater than one 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 $\lambda_1$, $\lambda_1$ 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 $conf(t_a \to t_d) < \lambda_2$ **then** $t_a$ **stays unmarked**

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

$N_S(t_{12}) = [\{x_1, x_2, x_3, x_4, x_5, x_7\}, \{x_6\}]$

For Classification Attributes in $S$:

$t_1 = (b, b_1 \to b_1)$, $t_2 = (b, b_2 \to b_2)$, $t_3 = (b, b_3 \to b_3)$, $t_4 = (a, a_1 \to a_2)$,

$t_5 = (a, a_1 \to a_1)$, $t_6 = (a, a_2 \to a_2)$, $t_7 = (a, a_2 \to a_1)$, $t_8 = (c, c_1 \to c_2)$,

$t_9 = (c, c_2 \to c_1)$, $t_{10} = (c, c_1 \to c_1)$, $t_{11} = (c, c_2 \to c_2)$, $t_{12} = (d, d_1 \to d_2)$.

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

$N_S(t_1) = [\{x_1, x_2, x_4, x_6\}, \{x_1, x_2, x_4, x_6\}]$ Not Marked /$Y_1 = Y_2$/

$N_S(t_2) = [\{x_3, x_7, x_8\}, \{x_3, x_7, x_8\}]$ Marked ”-” /$card(Y_2 \cap Z_2) = 0$/

$N_S(t_3) = [\{x_5\}, \{x_5\}]$ Marked ”-” /$card(Y_2 \cap Z_2) = 0$/

$N_S(t_4) = [\{x_1, x_6, x_7, x_8\}, \{x_2, x_3, x_4, x_5\}]$ Marked ”-” /$card(Y_2 \cap Z_2) = 0$/

$N_S(t_5) = [\{x_1, x_6, x_7, x_8\}, \{x_1, x_6, x_7, x_8\}]$ Not Marked /$Y_1 = Y_2$/

$N_S(t_6) = [\{x_2, x_3, x_4, x_5\}, \{x_2, x_3, x_4, x_5\}]$ Marked ”-” /$card(Y_2 \cap Z_2) = 0$/

$N_S(t_7) = [\{x_2, x_3, x_4, x_5\}, \{x_1, x_6, x_7, x_8\}]$ Marked ”+”
/rule $r_1 = [t_7 \Rightarrow t_{12}]$ has $conf = 1/2 \geq \lambda_2$, $sup = 2 \geq \lambda_1$/

$N_S(t_8) = [\{x_1, x_4, x_8\}, \{x_2, x_3, x_5, x_6, x_7\}]$ Not Marked
/rule $r_1 = [t_8 \Rightarrow t_{12}]$ has $conf = [2/3] \cdot [1/5] < \lambda_2$, $sup = 2 \geq \lambda_1$/

$N_S(t_9) = [\{x_2, x_3, x_5, x_6, x_7\}, \{x_1, x_4, x_8\}]$ Marked ”-” /$card(Y_2 \cap Z_2) = 0$/

$N_S(t_{10}) = [\{x_1, x_4, x_8\}, \{x_1, x_4, x_8\}]$ Marked ”-” /$card(Y_2 \cap Z_2) = 0$/

$N_S(t_{11}) = [\{x_2, x_3, x_5, x_6, x_7\}, \{x_2, x_3, x_5, x_6, x_7\}]$ Not Marked /$Y_1 = Y_2$/

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

$N_S(t_1 \cdot t_5) = [\{x_1, x_6\}, \{x_1, x_6\}]$ Not Marked /$Y_1 = Y_2$/

$N_S(t_1 \cdot t_8) = [\{x_1, x_4\}, \{x_2, x_6\}]$ Marked ”+”
/rule $r_1 = [[t_1 \cdot t_8] \Rightarrow t_{12}]$ has $conf = 1/2 \geq \lambda_2$, $sup = 2 \geq \lambda_1$/

$N_S(t_1 \cdot t_{11}) = [\{x_2, x_6\}, \{x_2, x_6\}]$ Not Marked /$Y_1 = Y_2$/

$N_S(t_5 \cdot t_8) = [\{x_1, x_8\}, \{x_6, x_7\}]$ Marked ”-”
/rule $r_1 = [[t_5 \cdot t_8] \Rightarrow t_{12}]$ has $conf = 1/2 \geq \lambda_2$, $sup = 1 < \lambda_1$/

$N_S(t_5 \cdot t_{11}) = [\{x_6, x_7\}, \{x_6, x_7\}]$ Not Marked /$Y_1 = Y_2$/

$N_S(t_8 \cdot t_{11}) = [\emptyset, \{x_2, x_3, x_5, x_6, x_7\}]$ 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 $t_1 \cdot t_5 \cdot t_8$ can be built. It is an extension of $t_5 \cdot t_8$ which is already marked as negative. So, the algorithm *ARD* stops and two candidate action rules are constructed: $[[(b, b_1 \to b_1) \cdot (c, c_1 \to c_2)] \Rightarrow (d, d_1 \to d_2)]$, $[(a, a_2 \to a_1) \Rightarrow (d, d_1 \to d_2)]$. Following the notation used in previous papers on action rules mining (see [10], [20], [19], [13]), the first of the above two candidate action rules will be presented as $[[(b, b_1) \cdot (c, c_1 \to c_2)] \Rightarrow (d, d_1 \to d_2)]$.

## VI. Action Rules Discovery

Influence matrix associated with $S$ and a set of meta-actions is used to identify which candidate action rules extracted by the algorithm $ARD$, presented in the previous section, are valid with respect to meta-actions and hidden correlations between classification attributes and the decision attribute.

Assume that $S = (X, A \cup \{d\}, V)$ is a decision system, $A - \{d\} = A_1 \cup A_2 \cup ... \cup A_m$, $\{M_1, M_2, ..., M_n\}$ are meta-actions associated with $S$, $\{E_{i,j} : 1 \leq i \leq n, 1 \leq j \leq m\}$ is the influence matrix, and $r = [(A_{[i,1]}, a_{[i,1]} \to a_{[j,1]}) \cdot (A_{[i,2]}, a_{[i,2]} \to a_{[j,2]}) \cdot ..... \cdot (A_{[i,k]}, a_{[i,k]} \to a_{[j,k]})] \Rightarrow (d, d_i \to d_j)$ is a candidate action rule extracted from $S$.

Table II
INFLUENCE MATRIX FOR $S$

| . | a | b | c |
|---|---|---|---|
| $M_1$ | | $b_1$ | $c_2 \to c_1$ |
| $M_2$ | $a_2 \to a_1$ | $b_2$ | |
| $M_3$ | $a_1 \to a_2$ | | $c_2 \to c_1$ |
| $M_4$ | | $b_1$ | $c_1 \to c_2$ |
| $M_5$ | | | $c_1 \to c_2$ |
| $M_6$ | $a_1 \to a_2$ | | $c_1 \to c_2$ |

Also, we assume here that $A_{[i,j]}(M_i) = E_{i,j}$. Value $E_{i,j}$ is either an atomic action set or *NULL* (not defined). By meta-actions based decision system, we mean a triple consisting with $S$, 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 $(\exists p \leq k)[A_{[i,p]}(M_i)$ is defined], then
$(\forall p \leq k)[$ if $A_{[i,p]}(M_i)$ is defined, then $(A_{[i,p]}, a_{[i,p]} \to a_{[j,p]}) = (A_{[i,p]}, E_{i,p})]$

We say that $r$ is valid in $S$ with respect to meta-actions $\{M_1, M_2, ..., M_n\}$, if there is $i$, $1 \leq i \leq n$, such that $r$ is valid in $S$ with respect to meta-action $M_i$.

To give an example, assume that $S$ is a decision system represented by Table I. 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 II. Clearly, each empty slot in Table II. corresponds to *NULL* value.

In the example presented in previous section, two candidate action rules have been constructed:
$r1 = [[(b, b_1) \cdot (c, c_1 \to c_2)] \Rightarrow (d, d_1 \to d_2)]$ and
$r2 = [(a, a_2 \to a_1) \Rightarrow (d, d_1 \to d_2)]$.

Clearly $r1$ is valid in $S$ with respect to $M_4$ and $M_5$. Also, $r_2$ is valid in $S$ with respect to $M_1$, $M_4$, $M_5$ because there is no overlap between the domain of action rule $r_2$ and the set of attributes influenced by any of these meta-actions. However, we can not say that $r2$ is valid in $S$ with respect to $M_2$ since $b_2$ is not listed in the classification part of $r_2$.

Assume assume that $S = (X, A \cup \{d\}, V)$ is a decision system with meta-actions $\{M_1, M_2, ..., M_n\}$ associated with $S$. Any candidate action rule extracted from $S$ which is valid in a meta-actions based decision system is called action rule. So, the process of action rules discovery is simplified to simple checking the validity of candidate action rules.

Since the rule $r_1$ is valid and applicable to $x_1$ and $x_4$, then it 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 Table III. is of type $\lambda$ (see [2]) and it is given below:

New candidate action rules can be extracted from $S_1$, using algorithm *ERID* [2], and next verified by meta-actions and the corresponding influence matrix associated with $S_1$. Now, if any new 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).

Table III
INFORMATION SYSTEM $S_1$

| . | a | b | c | d |
|---|---|---|---|---|
| $x_1$ | $a_1$ | $b_1$ | $c_1$ | $d_1$ |
| $x_2$ | $a_2$ | $b_1$ | $c_2$ | $d_1$ |
| $x_3$ | $a_2$ | $b_2$ | $c_2$ | $d_1$ |
| $x_4$ | $a_2$ | $b_1$ | $c_1$ | $d_1$ |
| $x_5$ | $a_2$ | $b_3$ | $c_2$ | $d_1$ |
| $x_6$ | $a_1$ | $b_1$ | $c_2$ | $d_2$ |
| $x_7$ | $a_1$ | $b_2$ | $c_2$ | $d_1$ |
| $x_8$ | $a_1$ | $b_2$ | $c_1$ | $d_3$ |
| $y_1$ | $a_1$ | $b_1$ | $c_2$ | $(d_2, 1/2)$ |
| $y_4$ | $a_2$ | $b_1$ | $c_2$ | $(d_2, 1/2)$ |

## VII. SUPPORT, CONFIDENCE, AND UTILITY OF ACTION RULES

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 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 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:

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

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

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

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

$N_S((b, b_1) \cdot (c, c_1 \to c_2)) = [\{x_1, x_4\}, \{x_2, x_6\}]$.

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

Essentially, the support takes the minimum of cardinality of sets of left side of action rule or cardinality of sets of right side of action rule. The confidence is computed as the support of left side of action rule divided by the cardinality of set of objects supporting decision attribute of left side, multiplied with the same for right side of action rule. Works, such as Tzacheva and Ras [23], as well as Hajja et. al. in [8] and [7] use these definitions.

However, these formulas are too complex for computation. Also, they are too restrictive, meaning we may obtain few action rules of the desired support and confidence for the particular decision attribute of interest. In addition, the formulas are undefined in the case of action rules extracted directly from the database without pair of association (classification) rules, or extracted based on a decision schema, where left side may be any attribute value.

For example, the action rule $r1$ has both the left side of each atomic action term specified, as well as the right side. While in rule $r2$ the left side is not specified, because it can be any value. It says that $a$ needs to bechanged to value $a_1$ no matter what the current value of $a$ is. Same for the decision attribute $d$ - if the actions specified by the action rule are undertaken, then the value of the decision attribute is expected to change to $d_2$ (the desired value), no matter what the current value is.

$r1 = [[(a, a_1 \to a_2) \wedge (b, b_1 \to b_2)] \Rightarrow (d, d_1 \to d_2)]$

$r2 = [[(a, \to a_2) \wedge (b, \to b_2)] \Rightarrow (d, \to d_2)]$.

Most of the recent algorithms for actino rule extraction take objects directly from the database, instead of using pair of two classification rules. Therefore, most recent algorithms end up producing rules of the type of $r_2$, where attribute values on the left side may not be specified. In that case, some of the sets in the forumals for computing support and confidence given above, are not defined. In particular, the set $Y_1$ which takes number of objects supporting the values of the left side of the rule. Also, the set $Z_1$ which takes the number of objects support the left value of the decision attribute.

To resolve this issue, we propose the following formulas for support and confidence of action rules:

$sup(r) = card(Y_2 \cap Z_2)$

We recall tha the set $Z_2$ gives us the objects, which have property the desired value of the decision attribute (or the right side of the decision attribute in the action rule). While the set $Y_2$ gives us the number of objects, which have attribute values on the right side of action rule for all other attributes.

In other words, the $support$ of the action rule gives the number of objects, which are already in the desired state or class, and already have the properties or the attribute values to which we are suggesting to change the rest of the qualifying objects. In this sense, the higher the support is, the stronger the action rule suggestion is.

$conf(r) = [\frac{card(Y_2 \cap Z_2)}{card(Y_2)}]$

The $confidence$ is the support of the action rule, divided by number of objects in set $Y_2$ i.e. objects which have attribute values on the right side of action rule for all attributes, except the decision.

We believe the proposed new formulas for support and confidence more accuratly represent the support and confidence of aciton rules. Also, they are less computationally expensive, and less restrictive, which would yield action rules of higher confidence.

Along with the support and the confidence of action rules, we propose computing a new measure which we call - the action rule $utility$. We define the utility as:

$util(r) = card(Y_1 \cap Z_1)$

The $utility$ gives us the objects, which have the $potential$ to be acted upon, and changed into the desired class or the desired state of the decision attribute. The higher the utility, the more useful or usable the action rule is.

Since we utilize the above metioned sets $Y_1$ and $Z_1$, we provide a definition for them in the case of rules of the type of $r_2$ above, where the left side is not specified, because it can be any value. In the case of $r_2$:

$r2 = [[(a, \to a_2) \wedge (b, \to b_2)] \Rightarrow (d, \to d_2)]$.

$Y_1 = [(\forall x \in X : a, \neg a_2) \cap (\forall x \in X : b, \neg b_2) \cap \ldots \cap (\forall x \in X : n, \neg n_2)]$

$Z_1 = [(\forall x \in X : d, \neg d_2)]$

Clearly, in the sets $Y_1$ and $Z_1$ are included only objects with flexible attributes specified by the action rule.

## VIII. Experiment with Mammographic Mass Dataset

We conduct an experiment using a Mammographic Mass Dataset, donated by Prof. Dr. Rdiger Schulz-Wendtland from the Institute of Radiology at the University Erlangen-Nuremberg, Germany [4]. This dataset is used to predict the severity (benign or malignant) of a mammographic mass lesion from BI-RADS attributes and the patient's age. It contains a BI-RADS assessment, the patient's age and three BI-RADS attributes together with the ground truth (the severity field) for 516 benign and 445 malignant masses that have been identified on full field digital mammograms collected at the University Erlangen-Nuremberg. The dataset contains 961 instances, and has 6 attributes (1 goal field, 1

Table IV
ACTION RULES EXTRACTED FROM MAMMOGRAPHIC MASS DATASET

| . | Action Rule |
|---|---|
| r1 | $(Marg, 3 \to 1) \Rightarrow (Sev, 1 \to 0)$ |
| r2 | $(Marg, 3 \to 1) \cdot (Shape, 4 \to 2) \Rightarrow (Sev, 1 \to 0)$ |
| r3 | $(BI-RADS, 4 \to 4) \cdot (Shape, 4 \to 2) \Rightarrow (Sev, 1 \to 0)$ |
| r4 | $(BI-RADS, 5 \to 4) \cdot (Dens, 3 \to 3) \Rightarrow (Sev, 1 \to 0)$ |
| r5 | $(Shape, 4 \to 2) \cdot (Dens, 3 \to 3) \Rightarrow (Sev, 1 \to 0)$ |
| r6 | $(Shape, 4 \to 1) \cdot (BI-RADS, 5 \to 4) \Rightarrow (Sev, 1 \to 0)$ |
| r7 | $(Marg, 3 \to 4) \cdot (BI-RADS, 5 \to 4) \Rightarrow (Sev, 1 \to 0)$ |
| r8 | $(Dens, 3 \to 3) \cdot (Shape, 4 \to 2) \Rightarrow (Sev, 1 \to 0)$ |
| r9 | $(Marg, 5 \to 1) \cdot (Shape, 4 \to 1) \Rightarrow (Sev, 1 \to 0)$ |
| r10 | $(Dens, 3 \to 3) \cdot (Marg, 3 \to 1) \Rightarrow (Sev, 1 \to 0)$ |

Table V
SUPPORT, CONFIDENCE, AND UTILITY OF ACTION RULES

| . | OSup | OConf | NSup | NConf | Util |
|---|---|---|---|---|---|
| r1 | 73 | 55.7 | 316 | 72.4 | 73 |
| r2 | 59 | 71 | 131 | 90.3 | 59 |
| r3 | 68 | 50.2 | 156 | 90.1 | 68 |
| r4 | 274 | 68 | 341 | 76.6 | 274 |
| r5 | 284 | 63.11 | 128 | 80 | 284 |
| r6 | 236 | 82.4 | 167 | 90.8 | 236 |
| r7 | 55 | 50.7 | 61 | 53.5 | 55 |
| r8 | 284 | 63.11 | 128 | 80 | 284 |
| r9 | 93 | 72.9 | 163 | 88.6 | 93 |
| r10 | 65 | 56 | 247 | 87.9 | 65 |

non-predictive, 4 predictive attributes). We designate 'BI-RADS', 'Shape', 'Margin', and 'Density' as the flexible attributes, assuming that we have control over changing the values of these lesion properties. We designate 'Age' as the stable attribute because we are unable to change the age of a patient. And finally, we designate 'Severity' as our decision (class) attribute. This attribute layout allows the action rules we extracted to suggest changes in flexible attributes, in order to re-classify a mammographic mass lesion from class: malignant to class: benign. We extracted 700 action rules from the Mammographic Mass Dataset while running the dataset through both the standard (Old) and the proposed (New) Formulas for calculating Support and Confidence. We use a Support and Confidence threshold of 45/50 respectively when mining for these action rules. Selected action rules extracted are shown in Table IV. The respective Old Support (OSup), New Support (NSup), Old Confidence (OConf), New Confidence (NConf), and Utility (Util) are shown in Table V.

Let us consider the second rule $r2$ in Table IV.:

$$r2 : (Marg, 3 \to 1) \cdot (Shape, 4 \to 2) \Rightarrow (Sev, 1 \to 0)$$

it means that if the Margin is changed from 3 to 1, and the Shape is changed from 4 to 2, then the Severity of the tumor is expected to change from 1 to 0, where 1 is Malignant and 0 is Benign. The suggested desired changes can be triggered by Meta-Actions described in Section 4. A possible Meta-Action for example could be: 'doctor prescribes specific medication', or 'doctor performs a specific medical procedure'. In Table V., we can see that the standard (Old) Support measure for the rule $r2$ is 59. The standard (Old) Confidence for this rule is 71%. The proposed (New) Support for this rule is 131. The proposed (New) Confidence for this rule is 90%. As we can see the proposed new measures allow for increased Support and Confidence. In addition the proposed measure of Utility, which is equal to 59, means that there are 59 objects that have the $potential$ to be acted upon, and changed into the desired class or the desired state of the decision attribute. The higher the utility, the more useful or usable the action rule is.

Table VI
AVERAGE INCREASE IN SUPPORT IN CONFIDENCE OF ACTION RULES USING NEW FORMULAS COMPARED TO OLD FORMULAS

| . | OldSup | OldConf | NewSup | NewConf | Util |
|---|---|---|---|---|---|
| Sum | 65,796 | 50,238 | 106,890 | 57,845 | 63,932 |
| Avg | 94.81 | 72.39 | 159.54 | 86.33 | 95.42 |

After the action rules are generated using both sets of Support and Confidence formulas (Old and New), we analyze the same action rules against the Support and Confidence results from each formula and create a summary spreadsheet to cross-reference the results. We discovered that using the proposed New Support and Confidence formulas on average led to a +64 increase in Support and +14 in Confidence levels per action rule, as shown in Table VI. This increase, coupled with the proposed Utility formula which shows the actions with the highest influence to change the decision attribute, allows action rules to more efficiently and precisely suggest ways to re-classify tumors from class: Malignant to class: Benign.

## IX. CONCLUSION

We employ a meta-action based decision 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 the influence matrix linking them. Meta-actions jointly with the influence matrix are used as a postprocessing tool in action rules discovery. Influence matrix shows the correlations among classification attributes triggered off by meta-actions. If the candidate actions rules are not on par with them, then they are not classified as action rules. However, if the influence matrix does not show all the interactions between classification attributes, then still some of the resulting action rules may fail when tested on real data.

We have introduced improved measures for support and confidence of action rules. The $support$ of the action rule gives the number of objects, which are already in the desired state or class, and already have the properties or the attribute values to which we are suggesting to change the rest of the qualifying objects. In this sense, the higher the support is, the stronger the action rule suggestion is. The $confidence$ is the support of the action rule, divided by number of objects in set $Y_2$ i.e. objects which have attribute values on the right side of action rule for all attributes, except the decision.

We have proposed a new measure for action rules which we call - the action rule $utility$. The $utility$ gives us the objects, which have the $potential$ to be acted upon, and changed into the desired class or the desired state of the decision attribute. The higher the utility, the more useful or usable the action rule is.

We conducted an experiment in medical domain with a Mammographic Mass Dataset. We extracted 700 action rules from this dataset, which suggest ways to re-classify breast tumors from malignant to benight severity class. We discovered that using the proposed New Support and Confidence formulas on average led to a +64 increase in Support and +14 in Confidence levels per action rule.

We believe the proposed new formulas for support and confidence more accuratly represent the support and confidence of aciton rules. Also, they are less computationally expensive, and less restrictive, which yields action rules of higher confidence. This increase in confidence, coupled with the proposed $utility$ formula, allows for the discovery of action rules, which more efficiently and precisely define ways to re-classify objects the the desired state. Therefore, we provide stronger and more accurate suggestions for the user of how to accomplish their desired goal.

## 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. Dardzinska, Z. Ras (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] A. Frank, A. Asuncion, (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

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

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

[7] A. Hajja, Z.W. Ras, A. Wieczorkowska (2014), Hierarchical Object-driven Action Rules, in Special issue on Mining Complex Patterns, A. Appice et al (Eds), Journal of Intelligent Information Systems, Springer, Vol. 42, No. 2, 207-232

[8] A. Hajja, A. Wieczorkowska, Z.W. Ras, R. Gubrynowicz (2013), Pair-based Object-driven Action Rules, in New Frontiers in Mining Complex Patterns, NFMCP 2012, ECML/PKDD Workshop, Bristol, United Kingdom, September, 2012, LNAI 7765, Springer, 2013, 79-93

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

[10] S. Im, Z.W. Ras (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

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

[12] 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

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

[14] Z.W. Ras, A. Dardzinska (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

[15] Z.W. Ras, A. Dardzinska, L.-S. Tsay, H. Wasyluk (2008), Association Action Rules, IEEE/ICDM Workshop on Mining Complex Data (MCD 2008), in Pisa, Italy, Proceedings, IEEE Computer Society

[16] Z.W. Ras, S. Gupta (2002), Action rules, re-classification of objects and e-commerce, in Soft Computing , Proceedings of the IIS'2002 Symposium, Sopot, Poland, Physica-Verlag, Springer, 101-108

[17] Z.W. Ras, L.S. Tsay (2003), Discovering extended action-rules, System DEAR, in Intelligent Information Systems 2003, Advances in Soft Computing, Proceedings of the IIS'2003 Symposium, Zakopane, Poland, Springer, 293-300

[18] Z.W. Ras, A. Tzacheva, L.S. Tsay, O. Gurdal (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

[19] Z.W. Ras, 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

[20] Z. Ras, 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

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

[22] A.A. Tzacheva, Z.W. Ras (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

[23] A.A. Tzacheva, Z.W. Ras (2010), Association Action Rules and Action Paths Triggered by Meta-Actions, in Proceedings of 2010 IEEE Conference on Granular Computing, Silicon Valley, CA, IEEE Computer Society, 772-776

[24] 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