Cooperative discovery of interesting action rules

Agnieszka Dardzińska[1] and Zbigniew W. Raś[2,3],

[1] Bialystok Technical Univ., Mathematics Dept., 15-351 Bialystok, Poland
[2] Univ. of North Carolina, Computer Science Dept., Charlotte, NC 28223, USA
[3] Polish-Japanese Institute of Information Technology, Intelligent Systems Dept., ul. Koszykowa 86, 02-008 Warsaw, Poland

Abstract. Action rules introduced in [12] and extended further to e-action rules [21] have been investigated in [22], [13], [20]. They assume that attributes in a database are divided into two groups: stable and flexible. In general, an action rule can be constructed from two rules extracted earlier from the same database. Furthermore, we assume that these two rules describe two different decision classes and our goal is to re-classify objects from one of these classes into the other one. Flexible attributes are essential in achieving that goal since they provide a tool for making hints to a user what changes within some values of flexible attributes are needed for a given set of objects to re-classify them into a new decision class. There are two aspects of interestingness of rules that have been studied in data mining literature, objective and subjective measures [8], [1], [14], [15], [23]. In this paper we focus on a cost of an action rule which was introduced in [22] as an objective measure. An action rule was called interesting if its cost is below and support higher than some user-defined threshold values. We assume that our attributes are hierarchical and we focus on solving the failing problem of interesting action rules discovery. Our process is cooperative and it has some similarities with cooperative answering of queries presented in [3], [5], [6].

1 Introduction

There are two aspects of interestingness of rules that have been studied in data mining literature, objective and subjective measures [8], [1], [14], [15]. Objective measures are data-driven and domain-independent. Generally, they evaluate rules based on their quality and similarity between them. Subjective measures, including unexpectedness, novelty, and actionability, are user-driven and domain-dependent. A rule is actionable if user can do an action to his/her advantage based on this rule [8]. A formal definition of an action rule, constructed from certain pairs of classification rules, has been proposed in [12] and investigated further in [21], [22], [13], [20]. Interventions introduced in [7] are similar to action rules. The idea behind either action rules or interventions is to construct special kind of rules showing what changes in values of attributes, for a given object, are needed in order to re-classify this object the way user wants. Assuming, for instance, that objects are customers, this re-classification may mean that a consumer not interested in a certain product, now may buy it, and therefore may shift into a group of more profitable customers.

The notion of a cost of an action rule was introduced in [22] as an objective measure. An action rule is called interesting if its cost is below some user-defined threshold value. For a given user, the cost associated with changes of values within one of his features is usually different than the cost associated with changes of values within his another feature. A heuristic strategy for replacing the initially extracted action rule by a composition of new action rules, dynamically built, was proposed in the paper by [22]. This composition of rules uniquely defines a new action rule and it is built with a goal to lower the cost of reclassifying objects supported by the initial action rule. However, in some cases the process of interesting action rules discovery may fail.

We assume, in this paper, that attributes are hierarchical and we show that failing problem of discovering interesting action rules can be treated in a similar way to the failing problem of database queries [3], [5], [6].

2 Information system and action rules

An information system is used for representing knowledge. Its definition, given here, is due to Pawlak [9].

By an information system we mean a triple $S = (U, A, V)$, where:

- U is a nonempty, finite set called the universe,
- A is a nonempty, finite set of attributes i.e. $a : U \longrightarrow V_a$ is a function for $a \in A$,
- $V = \bigcup\{V_a : a \in A\}$, where V_a is a set of values of the attribute $a \in A$.

Elements of U are called objects. In this paper, they are often seen as customers. Attributes are interpreted as features, offers made by a bank, characteristic conditions etc.

By a decision table we mean any information system where the set of attributes is partitioned into conditions and decisions. Additionally, we assume that the set of conditions is partitioned into stable and flexible. For simplicity reason, we assume that there is only one decision attribute. Date of birth is an example of a stable attribute. Interest rate on any customer account is an example of a flexible attribute (dependable on bank). We adopt the following definition of a decision table:

By a decision table we mean an information system of the form $S = (U, A_{St} \cup A_{Fl} \cup \{d\}, V)$, where $d \notin A_{St} \cup A_{Fl}$ is a distinguished attribute called decision. The elements of A_{St} are called stable conditions, whereas the elements of A_{Fl} are called flexible conditions.

As an example of a decision table we take $S = (\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}, \{a, c\} \cup \{b\} \cup \{d\}, V)$ represented by Table 1. The set $\{a, c\}$ lists stable attributes, b is a flexible attribute and d is a decision attribute. Also, we assume that H denotes a *high* profit and L denotes a *low* one.

Table 1. Decision System

	a	b	c	d
x_1	2	1	2	L
x_2	2	1	2	L
x_3	1	1	0	H
x_4	1	1	0	H
x_5	2	3	2	H
x_6	2	3	2	H
x_7	2	1	1	L
x_8	2	1	1	L
x_9	2	2	1	L
x_{10}	2	3	0	L
x_{11}	1	1	2	H
x_{12}	1	1	1	H

In order to induce rules in which the THEN part consists of the decision attribute d and the IF part consists of attributes belonging to $A_{St} \cup A_{Fl}$, we can use *LERS* for rules extraction [2].

In order to efficiently extract such rules, when the number of attributes is large, we use sub-tables $(U, B \cup \{d\}, V)$ of S where B is a d-reduct (see [9]) in S. The set B is called d-reduct in S if there is no proper subset C of B such that d depends on C. The concept of d-reduct in S was introduced to induce rules from S describing values of the attribute d depending on minimal subsets of $A_{St} \cup A_{Fl}$.

By $L(r)$ we mean all attributes listed in the IF part of a rule r. For example, if $r_1 = [(a_1, 2) \wedge (a_2, 1) \wedge (a_3, 4) \longrightarrow (d, 8)]$ is a rule then $L(r_1) = \{a_1, a_2, a_3\}$.

By $d(r_1)$ we denote the decision value of that rule. In our example $d(r_1) = 8$. Similarly, if $a \in A_{St} \cup A_{Fl}$, then by $a(r_1)$ we mean the value of a in the classification part of r_1. For instance, in our example, $a_1(r_1) = 2$, $a_2(r_1) = 1$.

From Table 1, we can extract the following five rules which support is greater or equal to 2:

$(b, 3) \wedge (c, 2) \longrightarrow (d, H)$, $(a, 1) \wedge (b, 1) \longrightarrow (d, L)$,
$(a, 1) \wedge (c, 1) \longrightarrow (d, L)$, $(b, 1) \wedge (c, 0) \longrightarrow (d, H)$,
$(a, 1) \longrightarrow (d, H)$

Now, let us assume that $(a, v \longrightarrow w)$ denotes the fact that the value v of attribute a is changed to w. Similarly, $(a, \longrightarrow w)$ denotes the fact that the current value of attribute a is changed to w. The term $(a, v \longrightarrow w)(x)$ means that the property (a, v) of object x is changed to (a, w). Similarly, $(a, \longrightarrow w)(x)$ denotes the fact that the current value of attribute a for the object x is changed to w.

Let $S = (U, A_{St} \cup A_{Fl} \cup \{d\}, V)$ is a decision table and classification rules r_1, r_2 are extracted from S. The notion of e-action rule constructed from r_1, r_2 (see [21]) is recalled in the section below. First, we assume that:

- $B = A_{St} \cap L(r_1) \cap L(r_2)$,
- $u_i = e_i(r_2)$, where $\{e_1, e_2, ..., e_q\} = [A_{St} \cap [L(r_2) - L(r_1)]]$ and $1 \leq i \leq q$,
- $t_i = c_i(r_2)$, where $\{c_1, c_2, ..., c_r\} = [A_{Fl} \cap [L(r_2) - L(r_1)]]$ and $1 \leq i \leq r$,
- $v_i = b_i(r_1)$ and $w_i = b_i(r_2)$, where $\{b_1, b_2, ..., b_p\} = [A_{Fl} \cap L(r_1) \cap L(r_2)]$ and $1 \leq i \leq p$.

Additionally, the following two constraints are placed on rules r_1, r_2:

- $d(r_1) = k_1$, $d(r_2) = k_2$ and $k_1 \leq k_2$, (k_2 is a higher level class than k_1),
- $(\forall a \in B)[a(r_1) = a(r_2)]$,

By (r_1, r_2)-e-action rule on $x \in U$ we mean the expression r:

$[\prod\{(a, a(r_1)) : a \in B\} \wedge (e_1, u_1) \wedge (e_2, u_2) \wedge ... \wedge (e_q, u_q) \wedge (b_1, v_1 \longrightarrow w_1) \wedge (b_2, v_2 \longrightarrow w_2) \wedge ... \wedge (b_p, v_p \longrightarrow w_p) \wedge (c_1, \longrightarrow t_1) \wedge (c_2, \longrightarrow t_2) \wedge ... \wedge (c_r, \longrightarrow t_r)](x) \Longrightarrow [(d, k_1 \longrightarrow k_2)](x)$

The term " $\Longrightarrow$ " means *"it is expected that"*.

Object $x \in U$ supports (r_1, r_2)-e-action rule r in $S = (U, A_{St} \cup A_{Fl} \cup \{d\}, V)$, if x supports rule r_1, there is an object y supporting rule r_2, and the following conditions are satisfied:

- $(\forall i \leq p)[b_i(x) = v_i] \wedge d(x) = k_1$
- $(\forall i \leq p)[b_i(y) = w_i] \wedge d(y) = k_2$
- $(\forall j \leq q)[e_j(x) = e_j(y) = u_j]$
- $(\forall a \in B)[a(x) = a(r_1)]$

By the support of (r_1, r_2)-e-action rule r in S, denoted by $Sup_S(r)$, we mean the set of all objects in U supporting r in S.

By the confidence of (r_1, r_2)-e-action rule r in S, denoted by $Conf_S(r)$, we mean

$[Sup_S(r)/Sup_S(L(r))] \times [Conf(r_2)]$

To find the confidence of (r_1, r_2)-e-action rule in S, we divide the number of objects supporting (r_1, r_2)-e-action rule in S by the number of objects supporting left hand side of (r_1, r_2)-e-action rule times the confidence of the classification rule r_2 in S.

Finally, (r_1, r_2)-e-action rule is an action rule [12], if its structure is reduced to:

$[(b_1, v_1 \longrightarrow w_1) \wedge (b_2, v_2 \longrightarrow w_2) \wedge ... \wedge (b_p, v_p \longrightarrow w_p)](x) \Longrightarrow [(d, k_1 \longrightarrow k_2)](x)$

Also, we say that r is e-action rule in S, if there are classification rules r_1, r_2 extracted from S such that r is (r_1, r_2)-e-action rule in S.

3 Cost and feasibility of action rules

Assume now that $S = (U, A_{St} \cup A_{Fl} \cup \{d\}, V)$ is a decision system, $b \in A_{Fl}$, and $b_1, b_2 \in V_b$ are its two values. By $\rho_{(S,x)}(b_1, b_2)$ we mean a number from $(0, +\infty]$ which describes the cost needed to re-classify a qualifying object $x \in U$ from b_1 to b_2. Object $x \in U$ qualifies for a change from b_1 to b_2, if $b(x) = b_1$. If the above change is not feasible in practice, then we write $\rho_{(S,x)}(b_1, b_2) = +\infty$. For instance, a user may have no clue how to relocate x, assuming that x is a person, from one place to another place. In such a case we would write $\rho_{(S,x)}(b_1, b_2) = +\infty$. The value of $\rho_{(S,x)}(b_1, b_2)$ close to *zero* is interpreted that it is quite trivial to re-classify x from b_1 to b_2 whereas any large value means that this re-classification is practically very difficult to achieve.

If $\rho_{(S,x)}(b_1, b_2) < \rho_{(S,x)}(b_3, b_4)$, then we say that the change of values from b_1 to b_2 is *more feasible* for x than the change from b_3 to b_4.

By $\rho_S(b_1, b_2)$, we mean the average cost of $\rho_{(S,x)}(b_1, b_2)$, for all qualifying objects $x \in U$. We assume here that values $\rho_S(b_1, b_2)$ are provided by experts. They are treated as atomic expressions and used to introduce the formal notion of the feasibility and the cost of action rules and e-action rules in S.

So, let us assume that $r = [\prod\{(a, a(r_1)) : a \in B\} \wedge (e_1, u_1) \wedge (e_2, u_2) \wedge ... \wedge (e_q, u_q) \wedge (b_1, v_1 \rightarrow w_1) \wedge (b_2, v_2 \rightarrow w_2) \wedge \wedge (b_p, v_p \rightarrow w_p)](x) \Rightarrow (d, k_1 \rightarrow k_2)(x)$ is a (r_1, r_2)-e-action rule in S. By the *cost* of r denoted by $cost(r)$ we mean the value $\sum\{\rho_S(v_k, w_k) : 1 \leq k \leq p\}$. We say that r is *feasible*, if $cost(r) < \rho_S(k_1, k_2)$.

It means that for any feasible e-action rule r, the cost of the conditional part of r is lower than the cost of its decision part and clearly $cost(r) < +\infty$.

We say that e-action rule r is interesting, if its cost is below and its support is above some user defined threshold values.

Assume now that d is a decision attribute in S, $k_1, k_2 \in V_d$, and the user would like to re-classify objects in S from the group k_1 to the group k_2. To achieve that, he may look for an appropriate action rule, possibly of the lowest cost value, to get a hint which attribute values have to be changed. To be more precise, let us assume that $R_S[(d, k_1 \to k_2)]$ denotes the set of all e-action rules in S having the term $(d, k_1 \to k_2)$ on their decision site. Now, among all e-action rules in $R_S[(d, k_1 \to k_2)]$ he may identify a rule which has the lowest cost value. But the rule he gets may not be interesting one (because its cost is still too high). Let us notice that the cost of e-action rule

$r = [\prod\{(a, a(r_1)) : a \in B\} \wedge (e_1, u_1) \wedge (e_2, u_2) \wedge ... \wedge (e_q, u_q) \wedge (b_1, v_1 \to w_1) \wedge (b_2, v_2 \to w_2) \wedge ... \wedge (b_p, v_p \to w_p)](x) \Rightarrow (d, k_1 \to k_2)(x)$

might be high only because of the high cost of one of its sub-terms in the conditional part of that rule.

Let us assume that $(b_j, v_j \to w_j)$ is that term. One option to handle this is to look for e-action rule r_1 in $R_S[(b_j, v_j \to w_j)]$ which has the smallest cost value and next take the concatenation of rules r, r_1 [22]. If the cost of the resulting rule is still too high and assuming that the decision attribute is hierarchical and k_3 is a parent of k_2, we may try to re-classify objects from k_1 to k_3. If the generalized e-action rule is interesting, we should check the lowest cost and support of the re-classification from k_1 to a child of k_3 which is the nearest to k_2.

Now, we recall the definition of concatenation of two rules. Let us assume the following scenario. The action rule $r = [(b_1, v_1 \to w_1) \wedge (b_2, v_2 \to w_2) \wedge ... \wedge (b_p, v_p \to w_p)](x) \Rightarrow (d, k_1 \to k_2)(x)$, extracted from the information system S, is not interesting because at least one of its terms, let us say $(b_j, v_j \to w_j)$ where $1 \leq j \leq p$, has too high cost $\rho_{S_i}(v_j, w_j)$ assign to it.

In this case we look for a new feasible action rule $r_1 = [(b_{j1}, v_{j1} \to w_{j1}) \wedge (b_{j2}, v_{j2} \to w_{j2}) \wedge ... \wedge (b_{jq}, v_{jq} \to w_{jq})](y) \Rightarrow (b_j, v_j \to w_j)(y)$ which concatenated with r will decrease the cost value of desired reclassification.

By the concatenation of action rule r_1 with action rule r we mean a new feasible action rule $r_1 \circ r$ of the form:
$[(b_1, v_1 \to w_1) \wedge ... \wedge [(b_{j1}, v_{j1} \to w_{j1}) \wedge (b_{j2}, v_{j2} \to w_{j2}) \wedge ... \wedge (b_{jq}, v_{jq} \to w_{jq})] \wedge ... \wedge (b_p, v_p \to w_p)](x) \Rightarrow (d, k_1 \to k_2)(x)$
where x is an object in U.

The strategy of handling failing discovery of interesting action rules based on concatenation of action rules has one serious drawback. Each concatenation is decreasing the confidence of the resulting rule. If its confidence in S gets too low, then such action rule is no longer interesting to the user. The cooperative strategy, presented in this paper, gives an alternate approach to interesting action rules discovery.

4 Cooperative discovery of interesting action rules

Now, we present a simple example to illustrate the problem and outline the cooperative strategy of discovering interesting action rules. Let us assume that attributes a, e are stable in $S = (U, A \cup \{d\}, V)$, attributes b, c, d are flexible in S, and additionally the decision attribute d is hierarchical. Its structure, in Lisp-like notation, is given below:

$d[d_1[d_{[1,1]}, d_{[1,2]}], d_2[d_{[2,1]}, d_{[2,2]}], d_3[d_{[3,1]}, d_{[3,2]}]]$.

The system S is represented by Table 2.

X	a	b	c	e	d
x_1	a_2	b_1	c_1	e_1	$d_{[1,2]}$
x_2	a_2	b_1	c_1	e_2	$d_{[1,2]}$
x_3	a_1	b_1	c_1	e_1	$d_{[1,1]}$
x_4	a_1	b_1	c_1	e_2	$d_{[1,1]}$
x_5	a_1	b_2	c_2	e_2	$d_{[3,1]}$
x_6	a_2	b_2	c_1	e_2	$d_{[3,2]}$
x_7	a_1	b_2	c_1	e_2	$d_{[3,2]}$
x_8	a_1	b_1	c_2	e_2	$d_{[2,1]}$
x_9	a_1	b_2	c_2	e_1	$d_{[2,1]}$

Table 2. Decision System S

We wish to reclassify objects x_3, x_4 in S, from the class described by value $d_{[1,1]}$ to the class described by $d_{[3,1]}$. The following two classification rules can be extracted from S:

$r_1 = [(a, a_1) \wedge (b, b_1) \wedge (c, c_1) \rightarrow (d, d_{[1,1]})]$,
$r_2 = [(e, e_2) \wedge (b, b_2) \wedge (c, c_2) \rightarrow (d, d_{[3,1]})]$.

From the rules r_1, r_2, the following e-action rule is constructed:

$r = [(a, a_1) \wedge (e, e_2) \wedge (c, c_1 \rightarrow c_2) \wedge (b, b_1 \rightarrow b_2)](x) \Rightarrow$
$(d, d_{[1,1]} \rightarrow d_{[3,1]})(x)$.

Rule r is supported by object x_4. Now, assume that the cost of reclassification $(c, c_1 \rightarrow c_2)$ is too high and this is the only fact which makes r not interesting for the user. In this case, we generalize $d_{[3,1]}$ to d_3. A new classification rule $r_3 = [[(b, b_2) \wedge (e, e_2)] \rightarrow (d, d_3)]$ can be extracted from S. It is supported by objects $\{x_5, x_6, x_7\}$. The classification rule r_2 was supported only by x_5. From r_1, r_3, a new e-action rule $r = [(a, a_1) \wedge (e, e_2) \wedge (b, b_1 \rightarrow b_2)](x) \Rightarrow$ $(d, d_{[1,1]} \rightarrow d_3)(x)$ can be constructed. Since it does not refer to the attribute c and no new attributes are involved, the new e-action rule can be classified as interesting for the user. It is supported by object x_4 and its confidence is 100%. We can easily check that, in this particular example, its specification

$r = [(a, a_1) \wedge (e, e_2) \wedge (b, b_1 \rightarrow b_2)](x) \Rightarrow [(d, d_{[1,1]} \rightarrow d_{[3,2]})](x)$ has the same support and confidence as the previous e-action rule and the same is classified as interesting for the user.

Now, let us assume that $r = [[(b_1, v_1 \longrightarrow w_1) \wedge (b_2, v_2 \longrightarrow w_2) \wedge ... \wedge (b_p, v_p \longrightarrow w_p)] \Longrightarrow (d, z_1 \longrightarrow z_2)]$ is an action rule which cost is above some user specified threshold value. Rule r is produced from two classification rules r_1, r_2, where:

$r_1 = [(b_1, v_1) \wedge (b_2, v_2) \wedge ... \wedge (b_p, v_p)] \Longrightarrow (d, z_1)$,
$r_2 = [(b_1, w_1) \wedge (b_2, w_2) \wedge ... \wedge (b_p, w_p)] \Longrightarrow (d, z_2)$.

We also assume that attribute d is hierarchical and by $h(z_2)$ we denote its value which is a parent of z_2 in the tree representing the domain of d.

Now, we construct a new classification rule $G_d(r_2)$ by generalizing the decision value z_2 in r_2 to $h(z_2)$. In the next step, we check if by dropping any of the terms (b_i, w_i) listed in $L(G_d(r_2))$ we get a new rule r_2' which has the confidence above some user defined threshold value. If this is the case, then we drop that term in r_2 and produce a new action rule from r_1, r_2' and check its cost. If the cost is still too high we continue the process. If the cost is below the threshold value we stop the process.

In general, e-action rules show what changes within some values of flexible attributes are needed for a given set of objects to re-classify them from their current class d_1 into a new decision class d_2. If a discovered e-action rule has too high cost, we can try to replace class d_2 by its generalization. If attribute d is hierarchical and the notion of a distance between its values is given, then d_2 after its generalization can be specialized to the value d_3 which is the closest to d_2. Also, generalizations usually increase the support of rules involved.

5 Conclusion

For mining interesting e-action rules, we assume that user provides three threshold values: maximum cost λ_1 , minimum support λ_2, and minimum confidence λ_3. They are called interesting if their cost is lower than λ_1, support is higher than λ_2, and confidence is also higher than λ_3. Discovered e-action rules which satisfy the threshold requirement for confidence and support usually have too high cost. The strategy proposed in [22] shows how to decrease the cost of e-action rules by concatenating them with rules which are cheaper. But, the concatenation of two e-action rules has support usually lower than the support of the rules involved. So, we can easily fall down below the threshold λ_2. If this strategy does not work, then the cooperative process proposed in this paper may still produce interesting e-action rules which are satisfactory for a user.

References

1. Adomavicius, G., Tuzhilin, A. (1997) Discovery of actionable patterns in databases: the action hierarchy approach, in *Proceedings of KDD97 Conference*, Newport Beach, CA, AAAI Press.
2. Chmielewski M. R., Grzymala-Busse J. W., Peterson N. W., Than S., (1993), The Rule Induction System LERS - a version for personal computers in *Foundations of Computing and Decision Sciences*, Vol. 18, No. 3-4, Institute of Computing Science, Technical University of Poznan, Poland, 181-212
3. Chu, W., Yang, H., Chiang, K., Minock, M., Chow, G., Larson, C. (1996) Cobase: A scalable and extensible cooperative information system, in *Journal of Intelligent Information Systems*, Vol. 6, No. 2/3, 223-259
4. Fensel, D., (1998), *Ontologies: a silver bullet for knowledge management and electronic commerce*, Springer-Verlag
5. Gaasterland, T. (1997) Cooperative answering through controlled query relaxation, in *IEEE Expert*, Vol. 12, No. 5, 48-59
6. Godfrey, P. (1993) Minimization in cooperative response to failing database queries, in *International Journal of Cooperative Information Systems*, Vol. 6, No. 2, 95-149
7. Greco S, Matarazzo B, Pappalardo N, Slowinski R (2005) Measuring expected effects of interventions based on decision rules, Special Issue on Knowledge Discovery, (Ed. Z.W. Raś), *Journal of Experimental and Theoretical Artificial Intelligence*, Taylor and Francis, Vol. 17, No. 1-2, 103-118
8. Liu, B., Hsu, W., Chen, S. (1997) Using general impressions to analyze discovered classification rules, in *Proceedings of KDD97 Conference*, Newport Beach, CA, AAAI Press
9. Pawlak Z., (1985), *Rough Ssets and decision tables*, in Lecture Notesin Computer Science 208, Springer-Verlag, 186-196.
10. Pawlak, Z., (1991), *Rough Sets: Theoretical aspects of reasoning about data*, Kluwer Academic Publisher.
11. Raś, Z., Dardzińska, A., (2002) Handling semantic inconsistencies in query answering based on distributed knowledge mining, in *Foundations of Intelligent Systems*, Proceedings of ISMIS'02 Symposium, LNCS/LNAI, No. 2366, Springer-Verlag, 66-74
12. Raś, Z., Wieczorkowska, A., (2000), Action Rules: how to increase profit of a company, in *Principles of Data Mining and Knowledge Discovery*, (Eds. D.A. Zighed, J. Komorowski, J. Zytkow), Proceedings of PKDD'00, Lyon, France, LNCS/LNAI, No. 1910, Springer-Verlag, 587-592
13. Raś, Z.W., Tzacheva, A., Tsay, L.-S., (2005) Action rules, in *Encyclopedia of Data Warehousing and Mining*, (Ed. J. Wang), Idea Group Inc., 1-5
14. Silberschatz, A., Tuzhilin, A., (1995) On subjective measures of interestingness in knowledge discovery, in *Proceedings of KDD'95 Conference*, AAAI Press.
15. Silberschatz, A., Tuzhilin, A., (1996) What makes patterns interesting in knowledge discovery systems, in *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No. 6
16. Raś, Z., Gupta, S., (2002) Global action rules in distributed knowledge systems, in *Fundamenta Informaticae Journal*, IOS Press, Vol. 51, No. 1-2, 175-184
17. Skowron A., Grzymala-Busse J., (1991) From the Rough Set Theory to the Evidence Theory, in *ICS Research Reports*, 8/91, Warsaw University of Technology, October

18. Sowa, J.F., (2000) Ontology, Metadata and Semiotics, in *Conceptual Structures: Logical, Linguistic, and Computational Issues*, B. Ganter, G.W. Mineau (Eds), LNAI 1867, Springer-Verlag, 55-81
19. Suzuki, E., Kodratoff, Y., (1998) Discovery of surprising exception rules based on intensity of implication, in *Proc. of the Second Pacific-Asia Conference on Knowledge Discovery and Data mining (PAKDD)*
20. Tsay, L.-S., Raś, Z.W., Dardzińska, A. (2005) Mining E-Action Rules", in *Mining Complex Data*, Proceedings of 2005 IEEE ICDM Workshop in Houston, Texas, Published by Math. Dept., Saint Mary's Univ., Nova Scotia, Canada, 85-90
21. Tsay, L.-S., Raś, Z.W., (2005) Action rules discovery: System DEAR2, method and experiments, in *Special Issue on Knowledge Discovery*, Journal of Experimental and Theoretical Artificial Intelligence, Taylor and Francis, Vol. 17, No. 1-2, 119-128
22. Tzacheva, A., Raś, Z.W., (2005) Action rules mining, in *Special Issue on Knowledge Discovery*, International Journal of Intelligent Systems, Wiley, Vol. 20, No. 7, 719-736
23. Zbidi, N., Faiz, S., Limam, M.,(2006) On mining summaries by objective measures of interestingness, in Machine Learning, Springer, Vol. 62, No. 3, 175-198