Action Rules

Zbigniew W. Ras*
Department of Computer Science
University of North Carolina
9201 University City Blvd.
Charlotte, NC 28223, USA
voice: +1 704-687-4567
fax: +1 704-687-3516
email: ras@uncc.edu

Angelina Tzacheva
KDD Laboratory
Department of Computer Science
University of North Carolina
Charlotte, NC 28223, USA
voice: +1 704-687-4884
email: shmugla@yahoo.com

Li-Shiang Tsay
KDD Laboratory
Department of Computer Science
University of North Carolina
Charlotte, NC 28223, USA
voice: +1 704-687-4884
email: ltsay@yahoo.com

(*Corresponding author)

Action Rules

Zbigniew W. Ras, Angelina Tzacheva, & Li-Shiang Tsay

University of North Carolina, Charlotte, USA

INTRODUCTION

There are two aspects of interestingness of rules that have been studied in data mining literature, objective and subjective measures (Liu, 1997), (Adomavicius & Tuzhilin, 1997), (Silberschatz & Tuzhilin, 1995, 1996). Objective measures are data-driven and domain-independent. Generally, they evaluate the 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 (Liu, 1997). This definition, in spite of its importance, is too vague and it leaves open door to a number of different interpretations of actionability. In order to narrow it down, a new class of rules (called action rules) constructed from certain pairs of association rules, has been proposed in (Ras & Wieczorkowska, 2000). A formal definition of an action rule was independently proposed in (Geffner & Wainer, 1998). These rules have been investigated further in (Tsay & Ras, 2004) and (Tzacheva & Ras, 2004). To give an example justifying the need of action rules, let us assume that a number of customers have closed their accounts at one of the banks. We construct, possibly the simplest, description of that group of people and next search for a new description, similar to the one we have, with a goal to identify a new group of customers from which no-one left that bank. If these descriptions have a form of rules, then they can be seen as actionable rules. Now, by comparing these two descriptions, we may find the cause why these accounts have been closed and formulate an action which if undertaken by the bank, may prevent other customers from closing their accounts. Such actions are stimulated by action rules and they are seen as precise hints for actionability of rules. For example, an action rule may say that by inviting people from a certain group of customers for a glass of wine by the bank, it is guaranteed that these customers will not close their accounts and they do not move to another bank. Sending invitations by regular mail to all these customers or inviting them personally by giving them a call are examples of an action associated with that action rule.

In paper by (Ras & Gupta, 2002), authors assume that information system is distributed and its sites are autonomous. They show that it is wise to search for action rules at remote sites when action rules extracted at the client site can not be implemented in practice (suggested actions are too expensive or too risky). The composition of two action rules, not necessary extracted at the same site, was defined in (Ras & Gupta, 2002). Authors gave assumptions guaranteeing the correctness of such a composition. One of these assumptions requires that semantics of attributes, including the interpretation of null values, have to be the same at both sites. This assumption is relaxed in (Tzacheva & Ras, 2004) since authors allow different granularities of the same attribute at involved sites. In the same paper, they introduce the notion of a cost and feasibility of an action rule. Usually, a number of action rules or chains of action rules can be applied to re-classify a certain set of objects. The cost associated with changes of values within one attribute is usually different than the cost associated with changes of values within another attribute. The strategy for replacing the initially extracted action rule by a composition of new action rules, dynamically built, was proposed in the paper by (Tzacheva &

Ras, 2004). This composition of rules uniquely defines a new action rule and it was built with a goal to lower the cost of reclassifying objects supported by the initial action rule.

BACKGROUND

In the paper by (Ras & Wieczorkowska, 2000), the notion of an action rule was introduced. The main idea was to generate, from a database, special type of rules which basically form a hint to users showing a way to re-classify objects with respect to some distinguished attribute (called a decision attribute). Clearly, each relational schema gives a list of attributes used to represent objects stored in a database. Values of some of these attributes, for a given object, can be changed and this change can be influenced and controlled by user. However, some of these changes (for instance “profit”) can not be done directly to a decision attribute. In such a case, definitions of this decision attribute in terms of other attributes (called classification attributes) have to be learned. These new definitions are used to construct action rules showing what changes in values of some attributes, for a given class of objects, are needed to re-classify objects the way users want. But, users may still be either unable or unwilling to proceed with actions leading to such changes. In all such cases, we may search for definitions of values of any classification attribute listed in an action rule. By replacing a value of such attribute by its definition extracted either locally or at remote sites (if system is distributed), we construct new action rules which might be of more interest to business users than the initial rule.

MAIN THRUST OF THE CHAPTER

The technology dimension will be explored to clarify the meaning of actionable rules including action rules and extended action rules.

Action rules discovery in a standalone information system

An information system is used for representing knowledge. Its definition, given here, is due to (Pawlak, 1991).

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

1. U is a nonempty, finite set of objects (object identifiers),
2. A is a nonempty, finite set of attributes i.e. $a:U \rightarrow V_a$ for $a \in A$, where V_a is called the domain of a.

Information systems can be seen as decision tables. In any decision table together with the set of attributes a partition of that set into conditions and decisions is given. Additionally, we assume that the set of conditions is partitioned into stable and flexible conditions (Ras & Wieczorkowska, 2000).

Attribute $a \in A$ is called stable for the set U if its values assigned to objects from U can not change in time. Otherwise, it is called flexible. "Date of Birth" is an example of a stable attribute. "Interest rate" on any customer account is an example of a flexible attribute. For simplicity reason, we will consider decision tables with only one decision. We adopt the following definition of a decision table:

By a decision table we mean an information system $S = (U, A_1 \cup A_2 \cup \{d\})$, where $d \notin A_1 \cup A_2$ is a distinguished attribute called decision. The elements of A_1 are called stable conditions, whereas the elements of $A_2 \cup \{d\}$ are called flexible conditions. Our goal is to change values of attributes in A_1 for some objects from U so the values of the attribute d for these objects may change as well. Certain relationships between attributes from A_1 and the attribute d will have to be discovered first.

By Dom(r) we mean all attributes listed in the IF part of a rule r extracted from S. For example, if $r = [\ (a1,3)*(a2,4) \rightarrow (d,3)]$ is a rule, then Dom(r) = {a1,a2}. By d(r) we denote the decision value of rule r. In our example d(r) = 3.

If r1, r2 are rules and $B \subseteq A_1 \cup A_2$ is a set of attributes, then r1/B = r2/B means that the conditional parts of rules r1, r2 restricted to attributes B are the same.

For example if $r1 = [(a1,3) \rightarrow (d,3)]$, then r1/{a1} = r/{a1}.

Assume also that $(a, v \rightarrow w)$ denotes the fact that the value of attribute a has been changed from v to w. Similarly, the term $(a, v \rightarrow w)(x)$ means that a(x)=v has been changed to a(x)=w. Saying another words, the property (a,v) of an object x has been changed to property (a,w). Assume now that rules r1, r2 have been extracted from S and $r1/A_1 = r2/A_1$, d(r1)=k1, d(r2)=k2 and k1< k2. Also, assume that (b1, b2,…, bp) is a list of all attributes in $Dom(r1) \cap Dom(r2) \cap A_2$ on which r1, r2 differ and r1(b1)= v1, r1(b2)= v2,…, r1(bp)= vp, r2(b1)= w1, r2(b2)= w2,…, r2(bp)= wp.

By (r1,r2)-action rule on $x \in U$ we mean a statement:

$[\ (b1, v1 \rightarrow w1) \wedge (b2, v2 \rightarrow w2) \wedge \ldots \wedge (bp, vp \rightarrow wp)](x) \Rightarrow [(d, k1 \rightarrow k2)](x).$

If the value of the rule on x is true then the rule is valid. Otherwise it is false.

Let us denote by $U^{<r1>}$ the set of all customers in U supporting the rule r1. If (r1,r2)-action rule is valid on $x \in U^{<r1>}$ then we say that the action rule supports the new profit ranking k2 for x.

A (St)	B (Fl)	C (St)	E (Fl)	G (St)	H (Fl)	D (Decision)
a1	b1	c1	e1			d1
a1	b2			g2	h2	d2

Table 1

To define an extended action rule (Ras & Tsay, 2003), let us assume that two rules are considered. We present them in Table 1 to better clarify the process of constructing extended action rules. Here, "St" means stable classification attribute and "Fl" means flexible one.

In a classical representation, these two rules will have a form:

$r1 = [\ a1 * b1 * c1 * e1 \rightarrow d1\ ]$, $r2 = [\ a1 * b2 * g2 * h2 \rightarrow d2\ ]$.

Assume now that object x supports rule r1 which means that it is classified as d1. In order to re-classify x to class d2, we need to change its value B from b1 to b2 but also we have to require that $G(x)=g2$ and that the value H for object x has to be changed to h2. This is the meaning of the extended (r1,r2)-action rule given below:

$[(B, b1 \rightarrow b2) \wedge (G = g2) \wedge (H, \rightarrow h2)](x) \Rightarrow (D, d1\rightarrow d2)(x)$.

Assume now that by Sup(t) we mean the number of tuples having property t .

By the support of the extended (r1,r2)-action rule (given above) we mean:

$$Sup[(A=a1)*(B=b1)*(G=g2)]$$

By the confidence of the extended (r1,r2)-action rule (given above) we mean:

$$[Sup[(A=a1)*(B=b1)*(G=g2)*(D=d1)]/Sup[(A=a1)*(B=b1)*(G=g2)]]\bullet$$

$$[Sup[(A=a1)*(B=b2)*(C=c1)*(D=d2)]/Sup[(A=a1)*(B=b2)*(C=c1)]].$$

To give another example of extended action rule, assume that $S=(U,A_1 \cup A_2\cup\{d\})$ is a decision table represented by Table 2. Assume that $A_1= \{c, b\}$, $A_2 = \{a\}$.

	c	a	b	d
x1	2	1	1	L
x2	1	2	2	L
x3	2	2	1	H
x4	1	1	1	L

Table 2

For instance, rules $r1=[(a,1) * (b,1) \rightarrow (d,L)]$, $r2=[(c,2) * (a,2) \rightarrow (d,H)]$ can be extracted from S, where $U^{<r1>} = \{x1, x4\}$. Extended (r1,r2)-action rule

$[(a, 1 \rightarrow 2) \wedge (c = 2)](x) \Rightarrow [(d, L\rightarrow H)](x)$

is only supported by object x1. The corresponding (r1,r2)-action rule

$[(a, 1 \rightarrow 2)](x) \Rightarrow [(d, L \rightarrow H)](x)$ is supported by x1 and x4.

The confidence of an extended action rule is higher than the confidence of the corresponding action rule because all objects making the confidence of that action rule lower have been removed from its set of support.

Actions rules discovery in distributed autonomous information system

In (Ras & Dardzinska, 2002), the notion of a Distributed Autonomous Knowledge System (DAKS) framework was introduced. DAKS is seen as a collection of knowledge systems where each knowledge system is initially defined as an information system coupled with a set of rules (called a knowledge base) extracted from that system. These rules are transferred between sites due to the requests of a query answering system associated with the client site. Each rule transferred from one site of DAKS to another remains at both sites.

Assume now that information system S represents one of DAKS sites. If rules extracted from $S = (U, A_1 \cup A_2 \cup \{d\})$, describing values of attribute d in terms of attributes from $A_1 \cup A_2$, do not lead to any useful action rules (user is not willing to undertake any actions suggested by rules), we may:

1) search for definitions of flexible attributes listed in the classification parts of these rules in terms of other local flexible attributes (local mining for rules),

2) search for definitions of flexible attributes listed in the classification parts of these rules

in terms of flexible attributes from another site (mining for rules at remote sites),

3) search for definitions of decision attributes of these rules in terms of flexible attributes from another site (mining for rules at remote sites).

Another problem which has to be taken into consideration is the semantics of attributes which are common for a client site and some of the remote sites. This semantics may easily differ from site to site. Sometime, such a difference in semantics can be repaired quite easily. For instance, if *Temperature in Celsius* is used at one site and *Temperature in Fahrenheit* at the other, a simple mapping will fix the problem. If information systems are complete and two attributes have the same name and differ only in their granularity level, a new hierarchical attribute can be formed to fix the problem. If databases are incomplete, the problem is more complex because of the number of options available to interpret incomplete values (including null vales). The problem is especially difficult in a distributed framework when chase techniques based on rules extracted at the client and at remote sites are used by a client site to impute current values by values which are less incomplete. These problems are presented and partial solutions given in (Ras & Dardzinska, 2002).

Now, let us assume that the action rule

$$r = [(b_1, v_1 \rightarrow w_1) \wedge (b_2, v_2 \rightarrow w_2) \wedge \ldots \wedge (b_p, v_p \rightarrow w_p)](x) \Rightarrow (d, k_1 \rightarrow k_2)(x),$$

extracted from system S, does not provide any useful hint to a user for its actionability.

In this case we may look for a new action rule (extracted either from S or from some of its remote sites)

$$r_1 = [(b_{j1}, v_{j1} \rightarrow w_{j1}) \wedge (b_{j2}, v_{j2} \rightarrow w_{j2}) \wedge \ldots \wedge (b_{jq}, v_{jq} \rightarrow w_{jq})](y) \Rightarrow (b_j, v_j \rightarrow w_j)(y)$$

which concatenated with r may provide better hint for its actionability. For simplicity reason, we assume that the semantics and the granularity levels of all attributes listed in both information systems are the same.

Concatenation $[r_1 \circ r]$ is a new action rule (called global), defined as:

$$[(b_1, v_1 \rightarrow w_1) \wedge \ldots \wedge [(b_{1j}, v_{1j} \rightarrow w_{1j}) \wedge (b_{j2}, v_{j2} \rightarrow w_{j2}) \wedge \ldots \wedge (b_{jq}, v_{jq} \rightarrow w_{jq})]$$

$$\wedge \ldots \wedge (b_p, v_p \rightarrow w_p)](x) \Rightarrow (d, k_1 \rightarrow k_2)(x)$$

where x is an object in $S = (X,A,V)$.

Some of the attributes in $\{b_{j1}, b_{j2},\ldots, b_{jq}\}$ may not belong to A. Also, the support of r_1 is calculated in the information system from which r_1 was extracted. Let us denote that system by $S_m = (X_m, A_m, V_m)$ and the set of objects in X_m supporting r_1 by $Sup_{S_m}(r_1)$. Assume that $Sup_S(r)$ is the set of objects in S supporting rule r. The domain of $[r_1 \circ r]$ is the same as the domain of r which is equal to $Sup_S(r)$. Before we define the notion of a similarity between two objects belonging to two different information systems, we assume that $A=\{b_1, b_2, b_3, b_4\}$, $A_m=\{b_1, b_2, b_3, b_5, b_6\}$, and objects $x \in X$, $y \in X_m$ are defined by Table 3 given below.

The similarity $\rho(x, y)$ between x and y is defined as: $[1+0+0+1/2+1/2+1/2]=5/12$.

	b_1	b_2	b_3	b_4	b_5	b_6
x	v_1	v_2	v_3	v_4		
y	v_1	w_2	w_3		w_5	w_6

Table 3

To give more formal definition of similarity, we assume that:

$\rho(x,y) = [\sum\{\rho(b_i(x), b_i(y)) : b_i \in (A \cup A_m)\}] / card(A \cup A_m)$, where:

- $\rho(b_i(x), b_i(y))$, if $b_i(x) \neq b_i(y)$

- $\rho(b_i(x), b_i(y))$, if $b_i(x)=b_i(y)$
- $\rho(b_i(x), b_i(y))$, if either $b_i(x)$ or $b_i(y)$ is undefined.

Also, assume that

$\rho(x, Sup_{Sm}(r_1)) = \max\{\rho(x, y) : y \in Sup_{Sm}(r_1)\}$, for each $x \in Sup_S(r)$

By the confidence of the action rule $[r_1 \circ r]$ we mean

$[\sum\{\rho(x, Sup_{Sm}(r_1)) : x \in Sup_S(r)\} / card(Sup_S(r))] \cdot Conf(r_1) \cdot Conf(r)$

where Conf(r) is the confidence of the rule r in S and $Conf(r_1)$ is the confidence of the rule r_1 in S_m.

If we allow to concatenate action rules extracted from S with action rules extracted at other sites of DAKS, we are increasing the total number of generated action rules and the same our chance to find more suitable action rules for reclassifying objects in S is also increased.

FUTURE TRENDS

Business user may be either unable or unwilling to proceed with actions leading to desired reclassifications of objects. Undertaking the actions may be trivial, feasible to an acceptable degree, or may be practically very difficult. Therefore, the notion of a cost of an action rule is of very great importance. New strategies for discovering action rules of the lowest cost in DAKS, based on ontologies, will be investigated.

CONCLUSION

Attributes are divided into two groups: stable and flexible. By stable attributes we mean attributes which values can not be changed (for instance, age or maiden name). On the other hand attributes (like percentage rate or loan approval to buy a house) which values can be changed are called flexible. Rules are extracted from a decision table, using standard KD methods, with preference given to flexible attributes - so mainly they are listed in a classification part of rules. Most of these rules can be seen as actionable rules and the same used to construct action-rules.

REFERENCES

Adomavicius, G., Tuzhilin, A. (1997). Discovery of actionable patterns in databases: the action hierarchy approach, Proceedings of KDD97 Conference, Newport Beach, CA, AAAI Press.

Geffner, H., Wainer, J. (1998). Modeling action, knowledge and control, ECAI 98, 13th European Conference on AI, (Ed. H. Prade), John Wiley & Sons, 532-536.

Liu, B., Hsu, W., Chen, S. (1997). Using general impressions to analyze discovered classification rules, *Proceedings of KDD97 Conference*, Newport Beach, CA, AAAI Press.

Pawlak, Z., (1991). Rough Sets: Theoretical aspects of reasoning about data, Kluwer.

Ras, Z., Dardzinska, A. (2002). Handling semantic inconsistencies in query answering based on distributed knowledge mining, *Foundations of Intelligent Systems, Proceedings of ISMIS'02 Symposium*, LNAI, No. 2366, Springer-Verlag, 66-74.

Ras, Z., Gupta, S. (2002). Global action rules in distributed knowledge systems, *Fundamenta Informaticae Journal 51 (1-2)*, IOS Press, 175-184.

Ras, 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, LNAI, No. 1910, Springer-Verlag, 587-592.

Silberschatz, A., Tuzhilin, A., (1995). On subjective measures of interestingness in knowledge discovery, *Proceedings of KDD'95 Conference*, AAAI Press.

Silberschatz, A., Tuzhilin, A., (1996). What makes patterns interesting in knowledge discovery systems, *IEEE Transactions on Knowledge and Data Engineering 5 (6).*

Tsay, L.-S., Ras, Z.W. (2004). Action Rules Discovery System DEAR, Method and Experiments, in the Special Issue on Knowledge Discovery, (Ed. Z.W. Ras), *Journal of Experimental and Theoretical Artificial Intelligence*, Taylor and Francis, will appear.

Tzacheva, A., Ras, Z.W. (2004). Action rules mining, in the Special Issue on Knowledge Discovery, (Ed. Z.W. Ras), International Journal of Intelligent Systems, Wiley, will appear.

Tzacheva, A., Ras, Z.W. (2003). Discovering non-standard semantics of semi-stable attributes, *Proceedings of FLAIRS-2003*, St. Augustine, Florida, (Eds. I. Russell, S. Haller), AAAI Press, 330-334.

TERMS AND THEIR DEFINITION

Actionable rule: A rule is actionable if user can do an action to his/her advantage based on this rule.

Autonomous information system: Information system existing as an independent entity.

Domain of rule: Attributes listed in the IF part of a rule.

Flexible attribute: Attribute is called flexible if its value can be changed in time.

Knowledge base: A collection of rules defined as expressions written in predicate calculus. These rules have a form of associations between conjuncts of values of attributes.

Ontology: An explicit formal specification of how to represent objects, concepts and other entities that are assumed to exist in some area of interest and relationships holding among them. Systems that share the same ontology are able to communicate about domain of discourse without necessarily operating on a globally shared theory. System commits to ontology if its observable actions are consistent with definitions in the ontology.

Semantics: The meaning of expressions written in some language, as opposed to their syntax which describes how symbols may be combined independently of their meaning.

Stable attribute: Attribute is called stable for the set U if its values assigned to objects from U can not change in time.