Action Rules Mining

Zbigniew W. Ras
University of North Carolina, Charlotte, USA

Elzbieta Wyrzykowska
University of Information Technology and Management, Warsaw, Poland

Li-Shiang Tsay
North Carolina A&T State University, USA

INTRODUCTION

There are two aspects of interestingness of rules that have been studied in data mining literature, objective and subjective measures (Liu et al., 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 et al., 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). Interventions introduced in (Greco et al., 2006) and the concept of information changes proposed in (Skowron & Synak, 2006) are conceptually very similar to action rules. Action rules have been investigated further in (Wang at al., 2002), (Tsay & Ras, 2005, 2006), (Tzacheva & Ras, 2005), (He at al., 2005), (Ras & Dardzinska, 2006), (Dardzinska & Ras, 2006), (Ras & Wyrzykowska, 2007). 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 a 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 (Tzacheva & Ras, 2005) the notion of a cost and feasibility of an action rule was introduced. The cost is a subjective measure and feasibility is an objective measure. 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 and leading to the same reclassification goal, was proposed in (Tzacheva & Ras, 2005). This composition of rules uniquely defines a new action rule. Objects supporting the new action rule also support the initial action rule but the cost of reclassifying them is lower or even much lower for the new rule. In (Ras & Dardzinska, 2006) authors present a new algebraic-type top-down strategy for constructing action rules from single classification rules. Algorithm ARAS, proposed in (Ras & Wyrzykowska, 2007), is a bottom-up strategy generating action rules. In (He at al., 2005) authors give a strategy for discovering action rules directly from a database.

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 reclassify 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 reclassify 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, we construct new action rules which might be of more interest to business users than the initial rule. Action rules can be constructed from pairs of classification rules, from a single classification rule, and directly from a database.

MAIN THRUST OF THE CHAPTER

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

Action Rules Discovery in Information Systems

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 be changed 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 $S = (U, A_{St} \cup A_{Fl} \cup \{d\})$, 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 $A_{Fl} \cup \{d\}$ are called flexible conditions. Our goal is to change values of attributes in A_{Fl} for some objects from U so values of the attribute d for these objects may change as well. A formal expression describing such a property is called an action rule (Ras & Wieczorkowska, 2000), (Tsay & Ras, 2005).

To construct an action rule (Tsay & Ras, 2005), let us assume that two classification rules, each one referring to a different decision class, are considered. We assume here that these two rules have to be equal on their stable attributes, if they are both defined on them. We use Table 1 to clarify the process of action rule construction. Here, "St" means stable attribute and "Fl" means flexible one.

In a standard representation, these two classification rules have a form:

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

Assume now that object x supports rule r1 which means that x is classified as d1. In order to reclassify 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 (r1,r2)-action rule r defined by the expression below:

Table 1. Two classification rules extracted from S

a (St)	b (Fl)	c (St)	e (Fl)	g (St)	h (Fl)	d (Decision)
a1	b1	c1	e1			d1
a1	b2			g2	h2	d2

$r = [[a1 \wedge g2 \wedge (b, b1 \rightarrow b2) \wedge (h, \rightarrow h2)] \Rightarrow (d, d1 \rightarrow d2)]$.

The term $[a1 \wedge g2]$ is called the header of the action rule. Assume now that by Sup(t) we mean the number of tuples having property t. By the support of (r1,r2)-action rule r we mean: $Sup[a1 \wedge b1 \wedge g2 \wedge d1]$. Action rule schema associated with rule r2 is defined as:

$[[a1 \wedge g2 \wedge (b, \rightarrow b2) \wedge (h, \rightarrow h2)] \Rightarrow (d, d1 \rightarrow d2)]$.

By the confidence Conf(r) of (r1,r2)-action rule r we mean:

$[Sup[a1 \wedge b1 \wedge g2 \wedge d1]/ Sup[a1 \wedge b1 \wedge g2]] \bullet [Sup[a1 \wedge b2 \wedge c1 \wedge d2]/ Sup[a1 \wedge b2 \wedge c1]]$.

System DEAR (Tsay & Ras, 2005) discovers action rules from pairs of classification rules.

Actions Rules Discovery, a New Simplified Strategy

A bottom-up strategy, called ARAS, generating action rules from single classification rules was proposed in (Ras & Wyrzykowska, 2007). We give an example describing its main steps.

Let us assume that the decision system S = (U, $A_{St} \cup A_{Fl} \cup \{d\}$), where U = {x1,x2,x3,x4,x5,x6,x7,x8}, is represented by Table 2. A number of different methods can be used to extract rules in which the THEN part consists of the decision attribute d and the IF part consists of attributes belonging to $A_{St} \cup A_{Fl}$. In our example, the set A_{St} = {a,b,c} contains stable attributes and A_{Fl} = {e,f,g} contains flexible attributes. System LERS (Grzymala-Busse, 1997) is used to extract classification rules.

We are interested in reclassifying d2-objects either to class d1 or d3. Four certain classification rules describing either d1 or d3 are discovered by LERS from the decision system S. They are given below:

$r1 = [b1 \wedge c1 \wedge f2 \wedge g1] \rightarrow d1$, $r2 = [a2 \wedge b1 \wedge e2 \wedge f2] \rightarrow d3$, $r3 = e1 \rightarrow d1$, $r4 = [b1 \wedge g2] \rightarrow d3$.

Action rule schemas associated with r1, r2, r3, r4 and the reclassification task either (d, d2 → d1) or (d, d2 → d3) are:

$r1[d2 \rightarrow d1] = [b1 \wedge c1 \wedge (f, \rightarrow f2) \wedge (g, \rightarrow g1)] \Rightarrow (d, d2 \rightarrow d1)$,
$r2[d2 \rightarrow d3] = [a2 \wedge b1 \wedge (e, \rightarrow e2) \wedge (f, \rightarrow f2)] \Rightarrow (d, d2 \rightarrow d3)$,
$r3[d2 \rightarrow d1] = [(e, \rightarrow e1)] \Rightarrow (d, d2 \rightarrow d1)$,
$r4[d2 \rightarrow d3] = [b1 \wedge (g, \rightarrow g2)] \Rightarrow (d, d2 \rightarrow d3)$.

We can show that $Sup(r1[d2 \rightarrow d1]) = \{x3, x6, x8\}$, $Sup(r2[d2 \rightarrow d3]) = \{x6, x8\}$, $Sup(r3[d2 \rightarrow d1]) = \{x3,x4,x5,x6,x7,x8\}$, $Sup(r4[d2 \rightarrow d3]) = \{x3,x4,x6,x8\}$.

Assuming that $U[r1,d2] = Sup(r1[d2 \rightarrow d1])$, $U[r2,d2] = Sup(r2[d2 \rightarrow d3])$, $U[r3,d2] = Sup(r3[d2 \rightarrow d1])$, $U[r4,d2] = Sup(r4[d2 \rightarrow d3])$ and by applying ARAS algorithm we get:

$[b1 \wedge c1 \wedge a1]^* = \{x1\} \not\subset U[r1,d2]$, $[b1 \wedge c1 \wedge a2]^* = \{x6, x8\} \subseteq U[r1,d2]$,
$[b1 \wedge c1 \wedge f3]^* = \{x6\} \subseteq U[r1,d2]$, $[b1 \wedge c1 \wedge g2]^* = \{x2, x7\} \not\subset U[r1,d2]$,
$[b1 \wedge c1 \wedge g3]^* = \{x3, x8\} \subseteq U[r1,d2]$.

ARAS will construct two action rules for the first action rule schema:

Table 2. Decision system

U	a	b	c	e	f	g	d
x1	a1	b1	c1	e1	f2	g1	d1
x2	a2	b1	c2	e2	f2	g2	d3
x3	a3	b1	c1	e2	f2	g3	d2
x4	a1	b1	c2	e2	f2	g1	d2
x5	a1	b2	c1	e3	f2	g1	d2
x6	a2	b1	c1	e2	f3	g1	d2
x7	a2	b3	c2	e2	f2	g2	d2
x8	a2	b1	c1	e3	f2	g3	d2

$[b1 \wedge c1 \wedge (f, f3 \rightarrow f2) \wedge (g, \rightarrow g1)] \Rightarrow (d, d2 \rightarrow d1)$,
$[b1 \wedge c1 \wedge (f, \rightarrow f2) \wedge (g, g3 \rightarrow g1)] \Rightarrow (d, d2 \rightarrow d1)$.

In a similar way we construct action rules from the remaining three action rule schemas.

ARAS consists of two main modules. To explain them in a better way, we use another example which has no connection with Table 2. The first module of ARAS extracts all classification rules from S following LERS strategy. Assuming that d is the decision attribute and user is interested in reclassifying objects from its value d1 to d2, we treat the rules defining d1 as seeds and build clusters around them. For instance, if A_{St} ={a, b, g} and A_{Fl} = {c, e, h} are attributes in S = (U, $A_{St} \cup A_{Fl} \cup \{d\}$), and r =[[a1 ∧ b1 ∧ c1 ∧ e1] → d1] is a classification rule in S, where Va={a1,a2,a3}, Vb={b1,b2,b3}, Vc={c1,c2,c3}, Ve={e1,e2,e3}, Vg={g1,g2,g3}, Vh={h1,h2,h3}, then we remove from S all tuples containing values a2, a3, b2, b3, c1, e1 and we use again LERS to extract rules from the obtained subsystem.

Each rule defining d2 is used jointly with r to construct an action rule. The validation step of each of the set-inclusion relations, in the second module of ARAS, is replaced by checking if the corresponding term was marked by LERS in the first module of ARAS.

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 either in an autonomous information system or a distributed one, based on ontologies, should be investigated.

(He et al., 2005) proposed a strategy for discovering action rules directly from a database. More research needs to be done also in that area.

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.

Dardzinska, A., Ras, Z. (2006). Cooperative discovery of interesting action rules, *Proceedings of FQAS 2006 Conference*, Milano, Italy, (Eds. H.L. Larsen et al.), Springer, LNAI 4027, 489-497.

Greco, S., Matarazzo, B., Pappalardo, N., Slowinski, R. (2005). Measuring expected effects of interventions based on decision rules, *Journal of Experimental and Theoretical Artificial Intelligence 17 (1-2)*, Taylor and Francis.

Grzymala-Busse, J. (1997). A new version of the rule induction system LERS, *Fundamenta Informaticae 31 (1),* 27-39.

He, Z., Xu, X., Deng, S., Ma, R. (2005). Mining action rules from scratch, in *Expert Systems with Applications* 29 (3), Elsevier, 691-699.

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. (2006). Action Rules Discovery, a new simplified strategy, in *Foundations of Intelligent Systems, Proceedings of ISMIS'06,* F. Esposito et al. (Eds.), Bari, Italy, LNAI 4203, Springer, 445-453.

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 1910, Springer, 587-592.

Ras, Z., Wyrzykowska, E. (2007). ARAS: Action rules discovery based on agglomerative strategy, in *Mining Complex Data, Post-Proceedings of the ECML/PKDD'07 Third International Workshop, MCD 2007*, LNAI, Springer, will appear.

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).*

Skowron, A., Synak, P. (2006). Planning Based on Reasoning about Information Changes, in *Rough Sets and Current Trends in Computing*, LNCS 4259, Springer, 165-173.

Tsay, L.-S., Ras, Z. (2005). Action Rules Discovery System DEAR, Method and Experiments, *Journal of Experimental and Theoretical Artificial Intelligence 17 (1-2)*, Taylor and Francis, 119-128.

Tsay, L.-S., Ras, Z. (2006). Action Rules Discovery System DEAR3, in *Foundations of Intelligent Systems, Proceedings of ISMIS'06*, F. Esposito et al. (Eds.), Bari, Italy, LNAI 4203, Springer, 483-492.

Tzacheva, A., Ras, Z. (2005). Action rules mining, *International Journal of Intelligent Systems 20 (7),* Wiley, 719-736.

Wang, K., Zhou, S., Han, J. (2002). Profit mining: From patterns to actions, in *Proceedings of EDBT'02*, 70-87.

KEY TERMS

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.

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