G-ACTIONS ONTOLOGY POST-PROCESSING WITH ACTION RULES

Angelina A. Tzacheva

University of South Carolina Upstate
Department of Informatics
800 University Way, Spartanburg, SC
USA
atzacheva@uscupstate.edu

Abstract. *Actionable knowledge is a highly desired discovery with data mining research. Action rules describe possible transitions of objects in an information system - from one state to another more desirable state, with respect to a distinguished attribute. In this paper we propose a new method for post-processing action rules through G-Actions ontology. It contains nodes of generalized actions knowledge, which are linked with individual terms at the lower levels. This system recommends actions a user can undertake to his/advantage based on the data. Application areas include financial and medical domain.*

Keywords: *actionable knowledge, action rules, post-processing, ontology.*

1 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 [13]. 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 [13]. An action rule was defined as a term $[(\omega) \wedge (\alpha \rightarrow \beta)] \Rightarrow (\varphi \rightarrow \psi)$, where ω is a conjunction of fixed condition features shared by both groups, $(\alpha \rightarrow \beta)$ represents proposed changes in values of flexible features, and $(\varphi \rightarrow \psi)$ 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 will trigger 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 located in an ontology [3] layer, which we therefore call actions ontology.

Clearly, it has to be a link between the 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 actions with changes of attribute 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 will tell us what changes within flexible attributes are needed to achieve that goal.

2 Previous Work

Action rules have been introduced in [13] and investigated further in [16][14][10][17] [15][4][9]. Paper [6] was probably the first attempt towards formally introducing the problem of mining action rules without pre-existing classification rules. Authors explicitly formulated it as a search problem in a support-confidence-cost framework. The proposed algorithm 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 [17]. 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 [7], is based on Pawlak's model of an information system S [8]. The goal was to identify certain relationships between granules defined by the indiscernibility relation on its objects. Some of these relationships uniquely define action rules for S. Paper [11] presents a strategy for discovering action rules directly from the decision system. Action rules are built from atomic expressions following a strategy similar to ERID [2]. Paper [18] introduced the notion of action as a domain-independent way to model the domain knowledge. Given a dataset about actionable features and a 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 triggers changes of attribute values described by terms [a ↓], [b ↑], 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 [18] 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* ↓], then [*b* ↑]. In this paper, such link is contained in the ontology layer. Clearly, expert does not know correlations between classification attributes and the decision attribute. Such correlations can be described as action rules and they have to be discovered from the decision system. Authors in [18] did not take into consideration stable attributes and their classification attributes are only numerical. In this paper, for simplicity reason, we use only symbolic attributes. Numerical attributes, if any, are discretized before action rules are discovered.

3 Information System and G-Actions

In this section we introduce the notion of an information system and G-actions.

By an information system [8] we mean a triple S = (X, At, V), where:

1. X is a nonempty, finite set of objects
2. At is a nonempty, finite set of attributes, i.e. $a : U \rightarrow V_a$, where V_a is called the domain of $a \in At$
3. $V = \cup\{ V_a: a \in A \}$.

For example, Table 1 shows an information system S with a set of objects $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$, set of attributes At = {a, b, c, d}, and a set of their values $V = \{a_1, a_2, b_1, b_2, b_3, c_1, c_2, d_1, d_2, d_3\}$.

Table 1. 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

An information system S = (X, At, V) is called a decision system, if one of the attributes in At is distinguished and called the decision. The remaining attributes in At are classification attributes. Additionally, we assume that $At = A_{St} \cup A_{Fl} \cup \{d\}$, 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 G-actions associated with S we mean higher level concepts modeling certain generalizations of actions introduced in [18]. G-actions, when executed, can influence or trigger changes in values of some flexible attributes in S. They are specified by expert. 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 G-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 actions will not influence the attribute Speaks English fluently and therefore its values will remain unchanged. However other values of attributes can be influenced easily. It should be mentioned here that an expert knowledge concerning G-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 attribute through the correlation. The goal of action rule discovery is to identify possibly all such correlations.

4 Action Rules

In earlier works in [13][16][14][10][15], action rules are constructed from classification rules. This means that we use pre-existing classification rules or generate them using a rule discovery algorithm, such as LERS [5] or ERID [2], then, construct action rules either from certain pairs of these rules or from a single classification rule. For instance, algorithm ARAS [15] generates sets of terms (built from values of attributes) around classification rules and constructs action rules directly from them. In [12] authors presented a strategy for extracting action rules directly from a decision system and without using pre-existing classification rules.

Let $S = (X, At, V)$ be an information system, where $V = \cup \{V_a : a \in At\}$. First, we recall the notion of an atomic action set [11]. 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 $(a, a_1 \rightarrow a_2)$, we often write (a, a_1) for any $a_1 \in V_a$.

By Action Sets [11] we mean a smallest collection of sets such that:
1. If t is atomic action set, then t is an action set.
2. If t_1, t_2 are action sets, then $t_1 \wedge t_2$ 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 actions A_1, A_2. It means that both A_1, A_2 can influence attributes a, b but attribute a in both cases has to remain stable. The corresponding action sets are: $(a, a_2) \wedge (b, b_1 \rightarrow b_2)$, $(a, a_2) \wedge (b, b_2 \rightarrow b_1)$.

Consider several actions, denoted $A_1, A_2, \ldots, A_n$. An action can influence the values of classification attributes in At. We assume here that $At - \{d\} = At_1 \cup At_2 \cup \ldots \cup At_m$. The influence of these actions on classification attributes in At is specified by the actions ontology.

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 $Dom(t_1) \cup Dom(t_2) \in At$ and $Dom(t_1) \cap Dom(t_2) = \emptyset$. 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 1. 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 $r = [[(a, a_2) \wedge (c, c_1 \rightarrow c_2)]) \Rightarrow (d, d_1 \rightarrow d_2)]$ is an example of an action rule. The rule says that if value a_2 remains unchanged and value c changes from c_1 to c_1 , then it is expected that the value d will change from d_1 to d_2. We recall that d is the distinguished (decision) attribute, which the user is interested in. The domain Dom(r) of action rule r is equal to {a, c, d}.

We extract candidate action rules by using algorithm ARD [11].

5 G-Actions Ontology Post-Processing Step

An ontology [3], which is a system of fundamental concepts, that is, a system of background knowledge of any knowledge base, explicates the conceptualization of the target world and provides us with a solid foundation on which we can build sharable knowledge bases for wider usability than that of a conventional knowledge base. From knowledge-based systems point of view, it is defined as “a theory (system) of concepts/ vocabulary used as building blocks of an information processing system” by Mizoguchi [3]. Ontologies are agreements about shared conceptualizations. A very simple case would be a type hierarchy, specifying classes and their subsumption relationships.

Actions ontology associated with S is used to identify which candidate action rules, extracted by the algorithm ARD, are valid with respect to our G-actions and hidden correlations between classification attributes and the decision attribute. We build the actions ontology through expert knowledge of the domain. In this case, expert knowledge about the actions, and what changes within classification attributes those actions would trigger. In other words, what consequences happen when an action is undertaken.

Assume that $S = \{X, At \cup \{d\}, V\}$ is an information system, where $At - \{d\} =\{At_1, At_2,\ldots, At_k\}$. Assume that $\{G_1, G_2, \ldots, G_n\}$ is a set of G-actions associated with S, $O(\{G_1, G_2, \ldots, G_n\} \cup \{r_j : 1 \leq j \leq m\}, \leq)$ is the ontology associated with S, where r_j is an atomic action set for any $j \in \{1,2,\ldots,m\}$ and $\leq$ shows the influence of G-actions on atomic action sets. We also assume that $G_i \leq r_j$, if G-action G_i triggers atomic action set r_j. Clearly, an atomic action set is uniquely linked with an attribute in S. For instance, if $r_i = (At_i, At_{[i,j]} \rightarrow At_{[i,k]})$ is an atomic action set, then At_i is uniquely linked with r_i. Now, the fact that G-action G_i triggers atomic action set r_j will be stated as $At_j(G_i)=r_j$.

By ontology based information system, we mean a couple consisting of the information system S and the ontology O.

We say that $r=[\prod_j(At_j, At_{[j,i]} \rightarrow At_{[j,k]}) \Rightarrow (d, d_1 \rightarrow d_2)]$ is valid in S with respect to G-action G_i, if for every j the following condition holds:

$$\text{if } [At_j(G_i) \text{ is defined] then } (At_j, At_{[j,i]} \rightarrow At_{[j,k]}) = (At_j, r_j).$$

We say that r is valid with respect to actions ontology O, if there is i, $1 \leq i \leq n$, such that r is valid in S with respect to at least one G-action G_i specified in O.

To give an example, assume that S is an information system represented by Table 1 and $\{G_1, G_2, \ldots, G_n\}$ is the set of G-actions assigned to S with an ontology O shown in Figure 1. Assume two candidate action rules have been constructed by the algorithm ARD.

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

r_1 is valid in S with respect to $G_4 = \{(b,b_1),(c,c_1 \rightarrow c_2)\}$ and $G_5 = \{(c,c_1 \rightarrow c_2)\}$. However, we cannot say that r_2 is valid in S with respect to $G_2 = \{(b,b_2),(a,a_2 \rightarrow a_1)\}$ since b_2 is not listed in the classification part of r_2.

Assume that S is an information system with actions ontology O. Any candidate action rule extracted from S, which is valid in the ontology based information system is called action rule. In this way, the process of action rules discovery is simplified to checking the validity of candidate action rules.

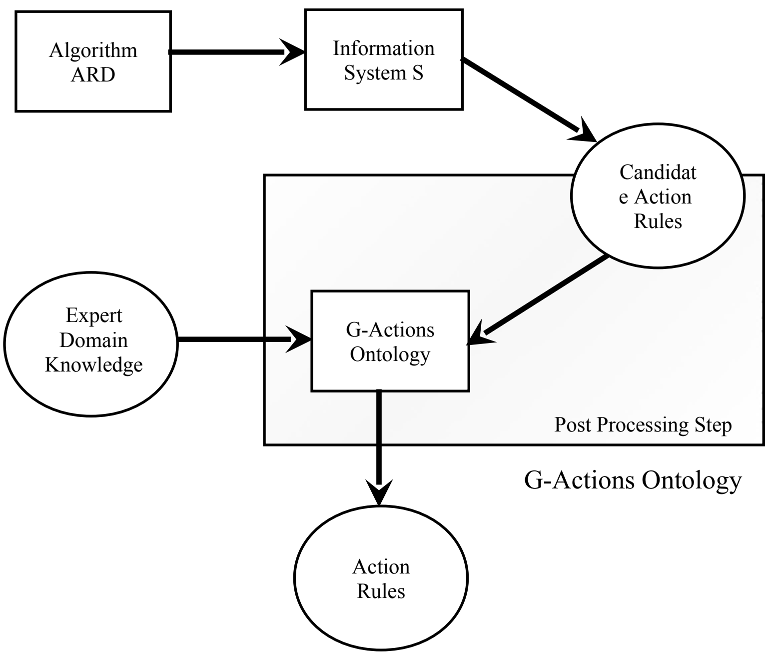

Fig. 1. Action Rule Discovery with G-Actions Ontology Post Processing

6 Conclusions

We introduce an ontology as a post-processing step in action rules discovery. An ontology based information system is a pair consisting of the information system S and the ontology O. The ontology shows the relationships between G-actions and atomic action sets they trigger. Actions ontology is used as a post-processing tool in action rules discovery. If the candidate action rules are not in agreement with G-actions, then they are not classified as action rules. However, if the ontology O does not contain all the interactions between G-actions and the atomic action sets, then still some of the resulting action rules may fail when tested on real data.

References:

[1] R. Agrawal and R. Srikant, Fast algorithm for mining association rules, Proceeding of the Twentieth International C onference on VLDB, 1994, pp. 487-499.

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

[3] R. Mizoguchi, Tutorial on ontological engineering - Part 1: Introduction to Ontological Engineering. New Generation Computing, OhmSha&Springer, Vol.21, No.4, 2003, pp. 365-384.
[4] S. Greco, B. Matarazzo, N. Pappalardo, R. Slowinski, Measuring expected effects of interventions based on decision rules, Journal of Experimental Theoretical Artificial Intelligence, Vol. 17, No. 1-2, 2005, pp. 103-118.
[5] J. Grzymala-Busse, A new version of the rule induction system LERS, Fundamenta Informaticae Journal, Vol. 31, No. 1, 1997, pp. 27-39.
[6] Z. He, X. Xu, S. Deng, R. Ma, Mining action rules from scratch, Expert Systems with Applications, Vol. 29, No. 3, 2005, pp. 691-699.
[7] S. Im and Z.W. Ras, 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, 2008, pp. 160-168.
[8] Z. Pawlak, Information systems - theoretical foundations, Information Systems Journal, Vol. 6, 1981, pp. 205-218.
[9] Y. Qiao, K. Zhong, H.-A. Wang and X. Li, Developing event-condition-action rules in real-time active database, *Proceedings of the 2007 ACM symposium on Applied computing*, ACM, New York, 2007, pp. 511-516.
[10] Z.W. Ras and A. Dardzinska, Action rules discovery, a new simplified strategy, Foundations of Intelligent Systems, LNAI, No. 4203, Springer, 2006, pp. 445-453.
[11] Z.W. Ras and A. Dardzinska, 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, 2008, pp. 181-190.
[12] Z.W. Ras, A. Dardzinska, L.-S. Tsay, H. Wasyluk, Association Action Rules, IEEE/ICDM Workshop on Mining Complex Data (MCD 2008), in Pisa, Italy, Proceedings, IEEE Computer Society, 2008, pp. 310-319.
[13] Z.W. Ras and A. Wieczorkowska, 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, 2000, pp. 587-592.
[14] Z.W. Ras, A.A. Tzacheva, L.-S. Tsay, O. Gurdal, Mining for interesting action rules, Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2005), Compiegne University of Technology, France, 2005, pp. 187-193.
[15] Z. Ras, E. Wyrzykowska, H. Wasyluk, 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, 2007, 196-208.
[16] L.-S. Tsay and Z.W. Ras, Action rules discovery system DEAR3, in Foundations of Intelligent Systems, Proceedings of ISMIS 2006, Bari, Italy, LNAI, No. 4203, Springer, 2006, pp. 483-492.
[17] A.A. Tzacheva and Z.W. Ras, Constraint based action rule discovery with single classification rules, in Proceedings of the Joint Rough Sets Symposium (JRS07), Toronto, Canada, LNAI, Vol. 4482, Springer, 2007, pp. 322-329.
[18] K. Wang, Y. Jiang, A. Tuzhilin, Mining Actionable Patterns by Role Models, in Proceedings of the 22nd International Conference on Data Engineering, IEEE Computer Society, 2006, pp. 16-25.