Mining of Frequent Action Rules

Agnieszka Dardzinska and Anna Romaniuk

Abstract An action rule is constructed as a series of changes, or actions, which can be made to some of the flexible characteristics of the information system that ultimately triggers a change in the targeted attribute. The existing action rules discovery methods consider the input decision system as their search domain and are limited to expensive and ambiguous strategies. In this paper, we define and propose the notion of action base as the search domain for actions, and then propose a strategy based on the FP-Growth algorithm to achieve high performance in action rules extraction. This method was initially tested on real medical diabetic database. The obtained results are quite promising.

Keywords Action rules · Association · Action base · FP-growth · Frequent action tree · Decision system · Information system · Diabetic

1 Introduction

Action rules were first introduced in [10] as a new class of rules that provides hints on possible actions a user should take to achieve a desired goal. Mining action rules is defined as the process of identifying patterns in a decision system capturing the possible changes to certain object attributes that may lead to a change in the decision value [2, 10]. Generally, action rule mining operates on a decision system [2, 7, 8] with objects having three classes of attributes: stable or semi-stable, flexible and decisions. The stable attributes are attributes that cannot be changed or, in some approaches, require a very high cost to change them [2, 5, 7]. Taking into consideration medical database with diabetic diseases, examples of stable attributes are date of birth, sex, age, SSN. Simultaneously, flexible attributes are attributes which

A. Dardzinska (✉) · A. Romaniuk
Department of Mechanics and Computer Science, Bialystok University of Technology,
Ul. Wiejska 45a, 15-351 Bialystok, Poland
e-mail: a.dardzinska@pb.edu.pl

A. Romaniuk
e-mail: a.romaniuk@doktoranci.pb.edu.pl

D. Ryżko et al. (eds.), *Machine Intelligence and Big Data in Industry*,
Studies in Big Data 19, DOI 10.1007/978-3-319-30315-4_8

values can change, such as blood glucose, blood pressure, health condition, etc. The decisions are attributes which the user would like to change, from one state to more desirable. An example would be the medical treatment kind of illness.

Existent action rules discovery methods use a decision table as their primary search domain. In our approach, the discovery of action rules is based on a domain of actions that we create from the decision system, called the action base. The main contribution of this paper is to present action rules mining problem as the association mining problem framework using the action base as the new search domain.

2 Preliminaries

We assume that $S = (X, A, V)$ is an information system, where:

- X is a nonempty, finite set of objects,
- A is a nonempty, finite set of attributes,
- $V = \{\cup V_a : a \in A\}$ is a set of all attributes values.

Additionally, $a : X \to V_a$ is a function for any $a \in A$, that returns the value of the attribute of a given object. The attributes are divided into different categories: set of stable attributes A_{St}, set of flexible attributes A_{Fl} and set od decision attributes D, such that $A = A_{St} \cup A_{Fl} \cup D$. In this paper we analyze information systems with only one decision attribute d. The example of an information system S is represented as Table 1.

Information system is represented by eight objects, one stable attribute a, two flexible attributes b, c and one decision attribute d.

Action rules, with definition presented in Sect. 3 of this paper, are very interesting and promising in medical treatment fields. They can be 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. Previous methods based on extracting action rules [2, 7, 8, 10–12] were based on an existing set of classification rules. Certain pairs of these rules were combined to assign objects

Table 1 Information system S

X	a	b	c	d
x_1	a_1	b_1	c_2	d_1
x_2	a_1	b_1	c_1	d_1
x_3	a_2	b_1	c_1	d_1
x_4	a_2	b_2	c_1	d_2
x_5	a_2	b_2	c_2	d_2
x_6	a_2	b_1	c_1	d_1
x_7	a_2	b_2	c_1	d_2
x_8	a_2	b_1	c_2	d_2

from one class to another. There is also a method which allows to explore action rules directly from the decision system [7]. In [8, 9, 12], the proposed algorithm, called Action Rules Discovery (ARD), builds rules for a given decision using an iterative marking strategy. It considers the change in attribute value as an atomic-action-term of length one, and then an action-term is a composition of atomic-action-terms. ARD starts by generating all atomic-action-terms for a given set of attribute values and assigning a mark (unmarked, positive, negative) based on standard support and confidence measures. The action-terms marked as positive are used to construct the action rules. The unmarked terms are placed into the list. Next, it generates all possible action-terms of length two by combining terms in the list. The process continues iteratively, creating terms of greater length, until the fixed point is reached. In [8] authors presented an association type of action rules and used an Apriori like strategy to find frequent action sets to induce action rules. Like ARD, the algorithm AAR (Association Action Rule) considers atomic action sets being the fine granule used to construct longer rules (similar to items and item sets in association mining). The Apriori algorithm is used with few modifications. The main changes are mostly based on modifications to the definition of support and confidence and the calculation of the measures directly from the input decision system. Although these approaches have different definitions for objective measures like support and confidence, they use the same idea of atomic-action set, action set and Standard Interpretation.

3 Action Rules

By atomic action set we define the expression $(a, a_1 \rightarrow a_2)$ where a is an attribute in A and a_1, a_2 are values of a. If the attribute is stable or did not change its value then the atomic action set is expressed as (a, a_1). The domain of an atomic action set is its attribute.

Example 1 Consider*Diet*is a flexible attribute with values $V_D = \{1000\,\text{kcal}, 1200\,\text{kcal}, 3000\,\text{kcal}\}$. The atomic action set $(Diet, 3000\,\text{kcal} \rightarrow 1200\,\text{kcal})$ means changing the value of attribute *Diet* from 3000 to 1200 kcal.
Action sets are constructed as the conjunction of atomic actions. If t_1, t_2 are two atomic action sets with different attributes, then $t = t_1 * t_2$ is an action set. The domain of action set t is the set of attributes from all its atomic action sets $Dom(t) = Dom(t_1) \cup Dom(t_2)$.

Example 2 Let us assume that *Age* is a stable attribute with values. $V_A = \{20, 40, 60\}$ and *Weight* is a flexible attribute with values $V_W = \{overweight, equal\_weight, underweight$. An action set could be the composition $(Age, 40) * (Weight, overweight \rightarrow equalweight) * (Diet, 3000\,\text{kcal} \rightarrow 1200\,\text{kcal})$ means: for patients of Age 40, lower the body weight and change the diet. The introduction of the Standard Interpretation is the basis of measures like support and confidence [1, 2, 7]. In association mining, the support of an itemset is the count of all objects in information system. For action rules, we need to take into consideration two sets. The first set is the set of all the objects with attributes value equal to the initial state of the action. The second one is the set of all the objects having attributes values equal to the values of the final state of the action.

Example 3 The Standard interpretation of action set $Ns[(Age, 40) * (Weight, overweight \rightarrow equalweight) * (Diet, 3000\,\text{kcal} \rightarrow 1200\,\text{kcal})] = [\alpha 1, \alpha 2]$, where:
$\alpha 1 = \{x \in X : [Age(x) = 40] * [Weight(x) = overweight] * [Diet, 3000\,\text{kcal}]\}$
$\alpha 2 = \{x \in X : [Age(x) = 40] * [Weight(x) = equalweight] * [Diet, 1200\,\text{kcal}]\}$.
Action rule r is expressed as an implication $r = [t1 \rightarrow t2]$, where $t1$ and $t2$ are two action sets.
Assume that t is an action set with standard interpretation $Ns(t) = [\alpha 1, \alpha 2]$. The support of this action set t is defined as in ARD strategy:

$$Sup(t) = min\{card(\alpha 1), card(\alpha 2)\}.$$

The confidence of an action *rule* $r = [t1 \rightarrow t2]$, where $Ns(t1) = [\alpha 1, \alpha 2]$ and $Ns(t2) = [\omega 1, \omega 2]$, with $\alpha 1 \neq 0$ and $\alpha 2 \neq 0$ is defined as follows:

$$Conf(r) = \frac{card(\alpha_1 \cap \omega_1)}{card(\alpha_1)} . \frac{card(\alpha_2 \cap \omega_2)}{card(\alpha_2)}$$

Generating action rules is similar to association rule mining where frequent item sets are first extracted. The algorithm, which is based on Apriori, generates actions sets with support that exceeds specified two threshold values: minimum support λ_1, and minimum confidence λ_2. Any action set that meets this criterion is a frequent action set. An action rule is constructed and tested as following:

If $t$is a frequent action set and *t1 is* a subset of t then $r = [t - t1 \rightarrow t1]$.

$$If\, Sup(r) < \lambda_1 \text{ and } Conf(r) < \lambda_2, \text{ then } r \text{ is ruled out.}$$
$$If\, Sup(r) \geq \lambda_1 \text{ and } Conf(r) < \lambda_2, \text{ then } r \text{ is not marked.}$$
$$If\, Sup(r) \geq \lambda_1 \text{ and } Conf(r) \geq \lambda_2, \text{ then } r \text{ is a valid rule.}$$

Presented method can generate a large number of rules. However the process does not constrain what the decision attribute can be, it does not require a decision attribute to be specified at all. Moreover, if the user is interested in a particular change object from one class to another like $(Weight, overweight \rightarrow equalweight)$, there is no guarantee that we will generate the required rules. For instance, if minimum support is 10, and (*Weight, overweight*) has support equal 9, then no rules containing (*Weight, overweight*) would be generated.

Methods which generate action rules mainly work in information systems in form of typical decision table, where there is a set of objects, set of attributes, and set of values of attributes [6]. As a result, forming flexible *atomic action sets* is revealed and connected with the action set generation. Hence, one would need to calculate the support of (*Weight, overweight*) and (*Weight, equal weight*) first, in order to obtain the support of $(Weight, overweight \rightarrow equalweight)$. This operation needs to be performed for all iterations. The first iteration generates frequent action sets composed of 1 element, the second iteration generates frequent action sets composed of 2 elements, the third iteration generates frequent action sets composed of 3 elements, and so on, until the fixed point is reached. In such way we obtain knowledge base. The information stored in form of action table are useful in later reclassification process, where objects move from one class to another.

Let us assume that Table 1 is the decision table and the decision attribute is $(d, d_1 \rightarrow d_2)$. It can be divided into two sub tables *Class1* and *Class2*, where:
Class1 = $\{x \in X : d(x) = d_1\}$, i.e. all the objects with decision value d_1.
Class2 = $\{x \in X : d(x) = d_2\}$, i.e. all the objects with decision value d_2.
$\forall x \in Class1, \forall y \in Class2 : (x \rightarrow y)$ is a possible transition describing a new action set *t*,
$\forall a \in A$: if a is flexible and $a(x) \neq a(y)$ then $a(t) = a(x) \rightarrow a(y)$,
$\forall a \in A$: if a is stable and $a(x) \neq a(y)$ then $a(t)$ is discarded,
$\forall a \in A$: if $a(x) = a(y)$ then $a(t) = a(x)$.
The action base, which will be formed, will contain the necessary action sets which help to move every object from *Class1* to *Class2*. The action base will contain *card*(*Class1*) * *card*(*Class2*) action sets. Each row in the action base reflects the necessary action sets for moving an object from d_1 to d_2.

Example 4 To generate action base we start from the decision Table 1 and organize it into two sub tables, as shown in Table 2, with respect to the decision attribute d. The action base includes change values as part of attribute value. The decision cardinality $card(\{d, d_1 \rightarrow d_2\})$ is equal to the number of rows in the action base. Moreover, the action base does not allow us to recover the individual counts of $\omega 1$ or $\omega 2$ nor the individual counts of $\alpha 1$ or $\alpha 2$. This information is not needed, then support for t, and confidence for r, where $t = t1 \cdot t2$ and $t2 = \{d, d_1 \rightarrow d_2\}$, are given by:
$Sup(t) = card(\alpha 1 \cap \omega 1) \cdot card(\alpha 2 \cap \omega 2), Conf(r) = \frac{Sup(r)}{Sup(t1)}$
Above description are the same support and confidence measures used in the traditional Association Mining [1, 4].

Table 2 Information system*S* divided into two separate classes

X	*a*	*c*	*c*	*d*	
x_1	a_1	b_2	c_2	d_1	*Class1*
x_2	a_1	b_1	c_1	d_1	
x_3	a_2	b_1	c_1	d_1	
x_6	a_2	b_1	c_1	d_1	
x_4	a_2	b_2	c_1	d_2	*Class2*
x_5	a_2	b_2	c_2	d_2	
x_7	a_2	b_2	c_1	d_2	
x_8	a_1	b_1	c_2	d_2	

The problem is that the *Sup(r)* can be directly calculated from the action base, but not the $Sup(t1)$. This is because the action base does not contain all occurrences of $\alpha 1$ and $\alpha 2$. However, the support for $\alpha 1$ and $\alpha 2$ can be easily calculated from the decision table. Formally, the support of an action set $t1$ with standard interpretation $Ns(t1) = [\alpha 1, \alpha 2]$ can be determined as $Sup(t1) = card(\alpha 1) \cdot card(\alpha 2)$.

4 Frequent Action Rules

In this paper we will use FP-Growth algorithm [2] to form frequent action tree, to generate the frequent action sets and then frequent action rules. Frequent action tree is obtained by order the transaction attributes values by their frequency, prune those that do not exceed minimum support given by user, and then insert action sets into a form of a tree. The result is compact data structure. The process of extracting action rules using frequent action tree is as follows:

1. Generate all the atomic action sets from set of attributes.
2. Calculate the frequency and the support of each atomic set.
3. Build and reorder the action base.
4. Build the frequent action tree from the action base.
5. Extract frequent action sets.
6. Extract action rules.

To better explain our strategy, we go through the example on the decision table described in Table 1. We use $\lambda_1 = 2$ and $\lambda_2 = 0.75$ as a minimum support and minimum confidence, respectively. Our goal is to reclassify objects from class d_1 to d_2. First we have to generate the atomic sets. These are all the possible transactions for each attribute.

$$AS = \{(a, a_1), (a, a_2), (b, b_1), (b, b_2), (b, b_1 \rightarrow b_2), (b, b_1), (c, c_1), (c, c_2), \\ (c, c_2 \rightarrow c_1)\}$$

For each atomic set we calculate its frequency. We scan the whole decision table and count the occurrence of each attribute with respect to decision (d_1, d_2).

All remain atomic action set build action base, which is in form of previous Table 3. We put all atomic set into descending order:

$$\{(a, a_2), (b, b_1 \rightarrow b_2), (c, c_1 \rightarrow c_2), (b, b_1), (c, c_1), (b, b_2), (c, c_2)\}$$

These ordered action sets form new action base.
To build the frequent action tree we insert each action set, from the action base, into the tree (Fig. 1) using the atomic action sets as the nodes. Each time a node is inserted or reused we increment its weight (Table 4).
Frequent action sets obtained from Table 5 are: $(a_2, b_1 \rightarrow b_2, c_1 \rightarrow c_2)$, $(a_2, b_1 \rightarrow b_2, c_1)$, $(a_2, b_1, c_1 \rightarrow c_2)$, $(a_2, b_1 \rightarrow b_2)$.

Table 3 Action base for decision attribute d

	a	b	c
$x_1 \to x_4$	$-$	b_2	$c_2 \to c_1$
$x_1 \to x_5$	$-$	b_2	c_2
$x_1 \to x_8$	$-$	$b_2 \to b_1$	c_2
$x_2 \to x_4$	$-$	$b_1 \to b_2$	c_1
$x_2 \to x_5$	$-$	$b_1 \to b_2$	$c_1 \to c_2$
$x_2 \to x_8$	$-$	b_1	$c_1 \to c_2$
$x_3 \to x_4$	a_2	$b_1 \to b_2$	c_1
$x_3 \to x_5$	a_2	$b_1 \to b_2$	$c_1 \to c_2$
$x_3 \to x_8$	a_2	b_1	$c_1 \to c_2$
$x_6 \to x_4$	a_2	$b_1 \to b_2$	c_1
$x_6 \to x_5$	a_2	$b_1 \to b_2$	$c_1 \to c_2$
$x_6 \to x_8$	a_2	b_1	$c_1 \to c_2$

Fig. 1 Frequent action tree

$$\begin{tikzcd}
 & \boxed{a_2} \arrow[dl] \arrow[dr] & \\
\boxed{b_1 \to b_2} \arrow[d] \arrow[dr] & & \boxed{b_1} \arrow[d] \\
\boxed{c_1 \to c_2} & \boxed{c_1} & \boxed{c_1 \to c_2}
\end{tikzcd}$$

Table 4 Support of atomic action set

Atomic action set	Support of atomic action set
(a, a_1)	0 not marked
(a, a_2)	6
(b, b_1)	3
(b, b_2)	2
$(b, b_1 \to b_2)$	6
$(b, b_2 \to b_1)$	1 not marked
(c, c_1)	3
(c, c_2)	2
$(c, c_1 \to c_1)$	6
$(c, c_2 \to c_1)$	1 not marked

Table 5 New ordered action base for decision attribute d

	a	b	c
$x_3 \to x_4$	a_2	$b_1 \to b_2$	c_1
$x_3 \to x_5$	a_2	$b_1 \to b_2$	$c_1 \to c_2$
$x_3 \to x_8$	a_2	b_1	$c_1 \to c_2$
$x_6 \to x_4$	a_2	$b_1 \to b_2$	c_1
$x_6 \to x_5$	a_2	$b_1 \to b_2$	$c_1 \to c_2$
$x_6 \to x_8$	a_2	b_1	$c_1 \to c_2$

From frequent action sets we get frequent action rules:
$r1 = (a, a2) \cdot (b, b_1 \to b_2) \cdot (c, c_1 \to c_2) \to (d, d_1 \to d_2)$ with $conf(r1) = 0.33$
$r2 = (a, a2) \cdot (b, b_1 \to b_2) \cdot (c, c_1) \to (d, d_1 \to d_2)$ with $conf(r2) = 0.66$
$r3 = (a, a2) \cdot (b, b_1) \cdot (c, c_1 \to c_2) \to (d, d_1 \to d_2)$ with $conf(r4) = 1$
$r4 = (a, a2) \cdot (b, b_1 \to b_2) \to (d, d_1 \to d_2)$ with $conf(r5) = 0.66$
For each extracted rule we calculate its confidence. In our example only $r3$ exceeds the minimum confidence for a valid association action rule extracted from information system S.

5 Experiments

Our dataset contains clinical data of 150 patients affected by diabetes. Patients are characterized by 15 attributes (5 stable and 10 flexible), and classified into three groups: diabetes, pre-diabetes and healthy. The goal was to find rules which help to reclassify patients to group of healthy persons.:
$[class, diabetes \to healthy]$, $[class, pre\text{-}diabetes \to healthy]$
We assumed $\lambda_1 = 30\,\%$, $\lambda_2 = 70\,\%$, and we obtained several action rules. Some of them are given below:

- If patient with type 2 diabetes is overweight and lose weight and begin regular physical activity, then his blood glucose returns to normal (sup = 70 %. Conf = 84 %)
- If patient will parasite cleanser, then his blood glucose will return to normal (sup = 35 %. Conf = 72 %)
- If patient will stop all hormone-effecting medications during the diabetes curing process, then her glucose tolerance will increase (sup = 90 %. Conf = 95 %)
- If patient will eliminate all processed foods, white foods, artificial sweeteners, tap water, artificial additives, then his blood glucose will remain normal (sup = 80 %. Conf = 85 %).

6 Conclusions and Future Work

In this paper, we propose the action base for frequent action rules mining. The action base transforms the complex problem of finding action rules from a decision table, into finding action rules from an action base. Applying this method to real medical databases we can notice that it is quite useful method, giving promising results. In the future direction, we will try to modify the method to compact frequent action set. We will also work on information systems with several decision attributes.

References

1. Agrawal, R., Srikant, R.: Fast algorithm for mining assocation rules. Int. Conf. Very Larg. Data Bases pp. 487–499 (1993)
2. Dardzinska, A.: Action Rules Mining. Springer, Berlin (2013)
3. Deogun, J., Raghavan, V., Sever, H.: Rough set based classification methods and extended decision tables. In: International Workshop on Rough Sets and Soft Computing. pp. 302–309 (1994)
4. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation, ACM SIGMOD. In: International Conference on Management of Data. pp. 1–12 (2000)
5. Im, S., Ras, Z.: Action Rule Extraction from a Decision Table: ARED. In: International Syposium on Methodologies for Intelligent Systems pp. 160–168 (2008)
6. Pawlak, Z.: Information systems—theoretical foundations. Inf. Syst. J. Elsevier **6**, 205–218 (1981)
7. Ras, Z., Dardzinska, A.: Action Rules Discovery without Pre-existing Classification Rules. In: The Sixth International Conference on Rough Sets and Current Trends in Computing pp. 181–190 (2008)
8. Ras, Z., Dardzinska, A., Tsay, L., Wasyluk, H.: Association Action Rules. In: IEEE International Conference on Data Mining Workshops pp. 283–290 (2008)
9. Ras, Z., Tsay, L.: Discovering Extended Action Rules (System DEAR). In: International IIS IIPWM'03 Conference. pp. 293–300 (2003)
10. Ras, Z., Wieczorkowska, A.: Action-Rules: How to Increase Profit of a Company, The Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases. pp. 587–592
11. Ras, Z., Wyrzykowska, E., Wasyluk, H.: ARAS: Action Rules Discovery Based on Agglomerative Strategy. In: Third International Workshop on Mining Complex Data. pp. 196–208 (2007)
12. Tsay, L., Ras, Z.: Action rules discovery: system DEAR2, method and experiment. J. Exp. Theor. Artif. Intell. pp. 119–128 (2005)