Available online at http://www.journalijdr.com

IJDR

International Journal of DEVELOPMENT RESEARCH

ISSN: 2230-9926

International Journal of Development Research
Vol. 07, Issue, 01, pp.11032-11039, January, 2017

Full Length Research Article

A NEW APPROACH TO CLINICAL MEDICINE BY ACTION RULES

*[1,2]Osman Gürdal and [3]Agnieszka Dardzińska

[1]Suleyman Demirel University School of Medicine, 32069 Isparta, Turkey
[2]Indiana University School of Medicine, Indianapolis, 46202 IN, USA
[3]Bialystok University of Technology, Dept. of Biocybernetics and Medical Engineering, 15-351 Bialystok

ARTICLE INFO

Article History:
Received 27th October, 2016
Received in revised form
25th November, 2016
Accepted 20th December, 2016
Published online 30th January, 2017

Key Words:

Action rules,
Actionability,
Measures of interestingness,
Clinical decision system.

ABSTRACT

Action rules are based on congruent predicaments and strategies that can be triggered when possible transitions of objects occur from one stage to another. This study presents a new way of implementing actionable discoveries in regards to object-driven approaches. To manifest usefulness of this new strategy in clinics, experiments were carried out in a new algorithmic database system in support of diagnoses of patients with liver disorders. The presented algorithm utilizes and expands the realization of action rules in which medical decision processes can be properly made for a clinical decision support system. The main features of the approach are: *i)* to utilize the discovered action rules, which are parallel to a bottom-up strategy that formulates the rules with a condition in minimal length; *ii)* to generate an object-driven action rule mining by the contingent algorithm where the data is processed by the expert system, producing actionable patterns due to object-driven action rules; *iii)* to discretize the data of the selected patients and extract the highest related attributes in test values; and *iv)* to validate the results along with the patients' history and physical examination. Object-driven approach is a shortcut of the DEAR algorithm in which classification rules produced design patterns and therefore their results are limited. On the other hand, the method of object-driven of expert system, where the rules are combined and actionable patterns are shifted through in terms of breath-first traversal and redundancy is minimized. As a result, the object-driven approach is more robust and faster that means it reduces the imputing time and tautology.

INTRODUCTION

Clinical decision support systems have been relatively Clinical decision support systems have been relatively recognized and have drawn attention specifically to the healthcare domain. They contribute to decisions made regarding patient care; when well-designed and performed, a system can bring ample capacity to offer better information to physicians and patients, and assists preventive medicine in a timely fashion. Therefore, such systems improve overall efficiency and quality of healthcare and helps lower the overwhelming medical spending. Nonetheless, the decision support system in this field should not be considered a substitute for the physician; rather, it must be seen as a technology to assist the complex decision-making process.

Corresponding author:** [1,2]**Osman Gürdal
[1]Suleyman Demirel University School of Medicine, 32069 Isparta, Turkey
[2]Indiana University School of Medicine, Indianapolis, 46202 IN, USA

In the Knowledge Discovery (KD) domain, recent progress in the computer-based clinical decision support systems is contingent upon the implementation of action rules known as identifiable patterns, which are exploitable, adaptable, and beneficial methodical strategies for the system in which solutions are produced (Ras *et al*., 2005; Frawley *et al*., 1991). Like other domains, computerized decision support systems are proven to be valuable assistances and tools for physicians to undertake formidable tasks in diagnoses, therapies, and prognoses. With these highly important roles, the support of decision making plays a vital role in the medical domain. The importance of the Knowledge Discovery Database (KDD) is to convert raw data into action that finally produces a competitive advantage and benefits for users (Dardzinska, 2013; Dardzinska *et al*., 2006). Still, the discovered patterns and the formulated solutions are not matching completely and there is a division. However, this variance is overcome by manual or semi-automatic analysis (Agrawal *et al*., 1993; Silberschatz *et al*., 1996), which takes time, is not efficient in the KDD process, and is usually biased. Therefore, with these crucial roles aforementioned, the field decision support systems emerge as highly substantial tools. Possible situations include the misdiagnosis of patients from unseasoned clinicians, causing low quality of treatments. This worsens treatment of a patient, consuming resources and time. Thus, the elimination of wastefulness and human error in diagnosis and treatment can be supported by automation in computerized systems, which may result in overall improvements in the quality of medical care and costs. For the expected outcomes of action ability in real conception, the KD system systematically should determine applicable and practical plans, which is an indispensable approach for the process. It was noted in (Pawlak, 1981) that one of the key elements of the KD process is the advancement of good measures of identified patterns or foundations. With that, the inputs of the algorithm consist of data and the knowledge of the expert system, which results in predicted outcome that is achieved by a specific task. In other words, actionable patterns are mined where the user can make an action for his gain. It is, therefore, highly imperative to find a workable analytic resolution to proximate the division between the identified object-driven patterns and the context of the designed action plan.

Practically, the discovered patterns in action ability comprise unbiased expressions of objective measures as the assessment is made through program-driven algorithms and approaches. However, these expressions do not seize all the involvements of the patterns in the process. On the other hand, determination of a decision by humans produce biased expressions that are subjective measures in the stage of appraisals and predictions. It has been mentioned that the objective measures (Silberschatz *et al*., 1996; Dhar *et al*., 1993; Frawley *et al.*, 1991), are not useful for all the complexities of the discovered pattern, and thus subjective measures are needed to complete the process. In view of these, there are several arguments combined with subjective measures, which consist of the structures of a pattern relying on the data associated with the discovery process, and mostly users who specifically explore the subjective interestingness: *i)* it only covers the expressed patterns; *ii)* it solely deals with domain-specific issues, for instance a healthcare-related problem; *iii)* the users may not be aware of all the facts that are related, and do not have enough knowledge about their field; and *iv)* it is usually a lengthy process and is time consuming to generate a more suitable decision. In short, objective measures neither contain domain knowledge nor acquire most intricacy of the patterns in the discovery process and the large number of the generated rules become redundant and irrelevant to the users. Thus, to construct true action ability patterns, one will need both objective and subjective measures. In the medical field, the task is to diagnose a specific disease in which all cases afflicted by it are described as components of the primary class. The rest are categorized as components of the secondary class, for instance healthy patients. In our study, the medical diagnoses are categorized by sensitivity, that is, the conditional probability of the set of correctly diagnosed cases from the primary class, and by specificity in the conditional probability of the set of correctly diagnosed cases from the secondary class.

Background and Framework

Action rules are logical terms defining knowledge for desirable actions related to the hidden objects in a database. The intent here is to concentrate on objective measures for actionability, which is defined as the extent to which a user can gain benefits from the discovered patterns, such as in the medical domain (Silberschatz *et al*., 1996; Dardzinska, 2013). Strategically, in classical terms, action rules depend upon precursory extraction of classification rules, and an evaluation is made by pairing them on such conditions that the expected outcomes are formulated with a decision feature (Silberschatz *et al*., 1996; Ras *et al.,* 2005*;* Ras *et al.,* 2006). Suppose an actionable goal of $p = [\omega \wedge (\alpha \rightarrow \beta) \rightarrow (\theta \rightarrow \psi)]$, *where* ω, α, β, θ, and ψ are descriptions of objects, for instance in the case of patients, where p is described as the satisfactions of a designed condition and the changeable measure of $(\alpha \rightarrow \beta)$ for patients who registered in a database with the expected result, $(\theta \rightarrow \psi)$. The algorithm for actionable strategy was implemented on the HEPAR clinical decision support system (Bobrowski, 1992; Wasluk *et al*., 2002) and was tested through different operating systems and modules, which are the part of a clinical database repository. Experimental procedures and decision-making processes were completed for patients with liver disorders.

There are two conceivable perspectives in terms of the strategies of actionability. One is the constituent of post-analysis at the back-end of the KD system (Silberschatz *et al*., 1996; He *et al*., 2005). This approach does not utilize the prior knowledge of the expert systems to lead the rule-generation process, which is purely subjective. The other approach is solely objective. It implements the input knowledge of the domain to control the rule generation process, which leads to determination of instrumental knowledge and comparing it with some standard beliefs. Any final outcome in regards to these beliefs will be either supportive or contradictory, both of which are viewed as interesting. Construction and implementation of this strategy can avoid unwanted and non-functional criteria, which includes reduction of time, tasks, and latency for the identified patterns. This measure eventually contributes more to the advancement of the KD process. In this paper, we concentrate on the object-driven approaches of actionability. In terms of construction methods, the discovered patterns in actionability can be grouped as rule-oriented and object-driven. In the case of rule-oriented patterns, pre-discovered rules play a significant role in the foundations in which the quality and quantity of action rules rely on selected classification methods (Ras *et al*., 2008; Dardzinska , 2013). Object-driven patterns, on the contrary, are generated straightforward from the dataset, and then implemented for the final outcome (He *et al*., 2005). This method cuts down manual utilization significantly, which means it avoids or reduces post-processing procedures.

Notations in Actionability

Considering an information system, which yields to $S = (I, A)$ represents knowledge (Pawlak, 1981), where I is a nonempty finite set of objects and A is a nonempty finite set of attributes, that is $a: I \rightarrow V_a$ is a function for any $a \in A$, where V_a is the domain of a. Constituents of I are named objects, and they are considered as patients or customers in our study. Assuming an information or decision system is described as $S = (I, A_{St} \cup A_{Fl} \cup \{d\})$, where $d \in (A_{St} \cup A_{Fl})$ is a distinctive attribute named the decision or decision system. There are two attributes split in system S: *i)* stable, $A_{St,}$, where the values cannot be changed and in some cases changes would require a high cost, e.g., inheritance or complexion of a patient; and *ii)* flexible, A_{Fl}, where the values can be changed, e.g., smoking habits or eating disorders of a patient. These attributes establish the set of conditional attributes. $d(I)=\{ k: (\exists x\in I)[d(x)=k]\}$ represents the number of elements and is called the d's rank, denoted by $p(d)$. It can be seen that categorizing the objects, $Part_s(p) = \{x_1, x_2,\ldots, x_{r(d)}\}$ in objects I are ruled by the attribute d, where $x_k= 1/d\ (\{k\})$ for $1 \le k \le p(d)$. Table 1 displays an example of an information system. In the table, there are seven objects characterized by four attributes, which are $\{b, c\}$ (flexible), a (stable), and d (the decision attribute where the analyst or physician would expect to see the results changed, e.g., recovery or healing state of a patient).

Table 1. Information system *S*

Objects	a stable	b flexible	c flexible	d decision
x_1	0	2	0	I
x_2	2	2	0	I
x_3	2	2	2	W
x_4	2	3	0	I
x_5	2	1	1	W
x_6	3	3	1	N
x_7	3	4	0	N

Action Rules

By comparing the likeness of two sets of targeted objects I, action rule mining establishes actionable and constructive procedures, which are desirable and undesirable. In order to predict precautionary models that are needed to get the aimed results, this action strategy goes beyond the learning process to forecast acceptable outcomes. Using the identified patterns becomes critical at this stage, since the management of the relationship regulates the transformation of unlikely objects that are moved into likely ones, in other words from one state to another. As an example, not only does a medical surgeon want to know the healing barriers of the patient, he also sees the importance of the recovery time of the patient. It further helps the physician to be aware of the acceleration of the prognosis and preventive actions. The concept of an action rule was more detailed in (Ras *et al.*, 2008 and references Frawley *et al.*, 1991 therein) and further examples are given in the experimental study.

By object-driven action rule p in an information system S, the expression is expressed as:

$p = [[(a_1 = \omega_1) \wedge (a_2 = \omega_2) \wedge \ldots \wedge (a_q = \omega_q)]\ (b_1, \alpha_1 \rightarrow\beta_1) \wedge (b_2, \alpha_2 \rightarrow \beta_2)\ldots\wedge(b_p, \alpha_p \rightarrow\beta_p)]\rightarrow [(d, k_1 \rightarrow k_2)]$, where $\{b_1, b_2, \ldots, b_p\}$ are flexible and $\{a_1, a_2,\ldots, a_q\}$ are stable attributes in S.

Further, it is assumed that $\omega_i\in Dom(a_i)$, $i = 1,2,\ldots,q$ and $\alpha_i, \beta_i\in Dom(b_i)$, $i=1,2,\ldots,p$ when $(a_i = \omega_i)$ the value of the attribute becomes a_i and is equal to ω_i, and $(b_j, \alpha_j\rightarrow \beta_j)$, and it shows that value of the attribute b_j has been changed from α_j to β_j. That is to say, object $x\in I$ supports an action rule p in S, if there is an object $y \in I$ such that: $(\forall_i \le p)\ [[b_i(x) = \alpha_i] \wedge [b_i(y) = \beta_i]]$, $(\forall_i \le q)\ [a_i(x) = a_i(y) = \omega_i]$, $d(x) = k_1$ and $d(y) = k_2$.

The approach here is to construct action plans by comparing the profiles of two sets of identified patients. Consider the following two fundamental object-driven stages: *i)* a left-hand side rule shown by P_L; and *ii)* a right-hand side rule shown by P_R. We therefore introduce three objective measures of the rule of interestingness: *Left Support*, *Right Support*, and *Confidence*.

i) The *Left Support, P_L*, represents an action rule in object-driven status in the field in which the rule is implemented, and it discovers the related objects in the set of objects, I. It is always favorable to have larger values since there will be more interesting rules extracted for a user. By the left support, the criterion of an object-driven action rule becomes:

$p = [[(a_1 = \omega_1) \wedge (a_2 = \omega_2)\wedge\ldots\wedge (a_q = \omega_q)] \wedge (b_1,\alpha_1 \rightarrow\beta_1)\wedge(b_2, \alpha_2 \rightarrow \beta_2)\wedge\ldots\wedge (b_p, \alpha_p \rightarrow\beta_p)] \rightarrow [(d, k_1 \rightarrow k_2)]$

and it is described as the set $P_L=V_L\cup \{k_1\}$, where $V_L= \{ \omega_1, \omega_2,\ldots, \omega_q, \alpha_1, \alpha_2,\ldots, \alpha_p\}$. The field $Dom_S(V_L)$ of the left support P_L is a set of objects in S that are in complete agreement with V_L. The number of objects in the domain cardinality is expressed as $Card[Dom_S (V_L)]$ and similarly, $Card\ [Dom_S (P_L)]$ represents the number of objects in S that are completely in agreement with P_L and $Card\ [I]$, where $Card\ [I]$ is the number of objects constructed in the information system, S. In terms of left support criterion, *suppL,* of an action rule, the expression becomes:

$suppL(p) = Card[Dom_S(P_L)] /Card[I]$.

ii) The *Right Support, P_R, defines* the strength of the rule that is attested by objects in S in terms of the more desirable decision class. The reclassification effect will be stronger as the value of support is higher. The right support criterion P_R is expressed as

$P_R= V_R\{k_2\}$, where $V_R= \{\omega_1, \omega_2,\ldots, \omega_q, \beta_1, \beta_2,\ldots, \beta_p\}$.

With $Dom_S(V_R)$ it is assumed that the set of objects in the system S matching V_R. $Card[Dom_S(P_R)]$ is the number of objects that are completely in agreement with P_R. By the right support *suppR* of action rule p, we consider $suppR(r) = Card[Dom_S(P_R)]/Card[U]$. The objects where the transformation takes place from a lower desired decision class to a higher one are determined by the success of the *confidence* of rule p. What is desired at this stage is the ratio of objects that reclassified and transferred into more desirable ones. In terms of the *confidence* of an action rule p in S, shown by $Conf_S(p)$, it means that:

$Conf_S(p)=(Card[Dom_S(P_L)]/Card[Dom_S(V_L)])(Card[Dom_S(P_R)]/Card[Dom_S(V_R)])$.

MATERIALS AND METHODS

The purpose of this investigation is to analyze the actionability in an unbiased approach. Object-driven perspective induces a set of structures that are implemented mathematically to evaluate a dataset. However, it is impossible to remove some of the subjectivity completely for deciding attributes to be stable for an action. This becomes obvious since action rules depend upon domain knowledge at initial stages. That said, the splitting of attributes whether flexible or stable has to be decided by users. This process, in fact, is utterly subjective. A flexible attribute for instance bleeding, smoking, and obstruction can be manipulated and controlled by users. On the contrary, the value of a stable attribute, for instance hearing, aging, and gender cannot be changed. In addition, defining the constraints can be accomplished via stable attributes to determine the objects selected to be evaluated in the system *S*, for instance "a person is blind, cannot start seeing". By implementing objective or subjective approach with action rules, some of the chosen objects may be reclassified from one stage to another stage by modifying some of the relevant flexible attributes. It depends on the characteristics of the corresponding flexible attributes. Since the subjectivity brings hindrance in some cases for a deciding attribute, for example, if a person who is suffering from cardiac arrest is just gasping, then a bystander who knows CPR can give an aid to the patient. On the other hand, physicians would have several options to lower the body temperature of a patient who has been resuscitated following cardiac arrest, where the value of the attribute is less than what it predicted. Therefore, doctors have several options to follow as decreasing the temperature may increase the chance of survival. This is considered to be a subjective step, which indicates that the action rules cannot remove all the cases of subjectivity.

Framework

The purpose of this study is to improve and utilize action rules for which an algorithm is proposed to implement the set of highly compact action rules. The algorithm is established on the object-driven action rules that a final decision of a medical diagnosis may be improved in regards to the clinical decision support system. A breadth-first type strategy is used to construct the algorithm and it rules out prior extraction of classification methods. All the action rules identified in the process are expressed to be compact without any redundancy. In terms of design, there are two cases for the algorithm: *i)* split the decision table into sub tables; and *ii)* construct the actionable patterns.

Splitting the reference table

The sub-tables, shown below in Table2, of Ts_1, Ts_2,…,Ts_p are grouped by dividing the decision table *S* by its attributes. These sub-tables are chosen due to the procedures of reclassification. This process presents viable strategies and often two sub-tables are selected. In fact, not only should data mining be adaptive and applicable for a specific condition, in this case the healthcare domain, but also it has to be precautionary to reduce high risk cases, such as monitoring hepatic encephalopathy for a patient diagnosed with cirrhosis. In the following case, the decision attributes of patients are categorized as healthy (*H*), have diseases (*D*),or are incurable (*N*). The domain of the attributes is expressed as *{D, H, N}*, and the reclassification is pointed from *D* to *H*.

Table 2. Decision system table *S*

Objects	A	B	C	D
x_1	0	2	0	I
x_2	2	2	0	I
x_3	2	2	2	W
x_4	2	3	0	I
x_5	2	1	1	W
x_6	3	3	1	N
x_7	3	4	0	N

The decision system table *S* in Table 2is split into Ts_1 *and* Ts_2 sub-tables as displayed in Table 3 in regards to the decision attributes $d \approx D$ and $d \approx H$. Since the incurable cases are excluded, the patients with decision value *N* are not taken into account and d ≈ *N* becomes omitted.

Table 3. Actionable methods built by sub-tables of Ts_1 and Ts_2 contain the selected objects

	Objects	*a*	*b*	*c*		*Objects*	*a*	*b*	*c*
Ts_1	x_1	0	2	0	Ts_2	x_3	2	1	2
	x_2	2	2	0		x_5	2	1	1
	x_4	2	3	0					

Constructing the actionable patterns

An action rule is in the form of $[(A_{St}, \omega) \wedge (A_{Fl}, \gamma \rightarrow \sigma)] \rightarrow (d, \theta \rightarrow \kappa)$ where (A_{St}, ω) is a premise-type stable, $(A_{Fl,}\gamma \rightarrow \sigma)$ is a premise-type flexible and $(d,\theta \rightarrow \kappa)$ is a decision-type atom. The following steps determine the construction: *i)* an atom is set for individual attribute and a unique candidate is picked, provided that it endorses minimum support. In order to qualify the identified atoms, the anti-monotonic condition is set. If the support condition of the atom is lower than the onset value, then a negative sign is given and thus it is removed from the list. *ii)* a series of integrations are performed on the candidates of flexible and decision type atoms through the assumptions made during the construction of an action rule. When the *Left Support*, *Right Support*, and *Confidence* satisfy the onset values of γ_1, γ_2, γ_3, then a rule is built and a positive sign is given on the atom with a high confidence. As this happens, the identified and eligible atom is also dropped from the list for the selection, which assures that the action rules identified are the most compact ones. At this stage, the object-driven term is introduced; namely, premise-type atoms join in as an input and associate with decision-type atoms concurrently to determine the rule in case it is approved. This recursive checking continues until there is no unsigned atom left. The stable atomic components are not taken into account since they are not the only ones building the object-driven action rules in the process. In addition, the algorithm sets up one more unsigned atomic element on the condition that it meets the onset value, otherwise, it is signed negatively and discarded.

The action rule is built with regard to associating atoms where they are freshly picked with high confidence of the atomic component $(d,\theta \rightarrow \kappa)$ and that it will either be granted or rejected. The recursive call continues processing the unsigned candidates and adding new atomic components to them concurrently until there is no input available. Reclassifying the algorithm means placing the objects from an undesirable to a more desirable status. The objects L_S* and R_S*, which have the properties of P_L and P_R in the decision system *S*, are called granules. The first loop of the process is formed as atomic stages of premise type $(d,\theta \rightarrow \kappa)$ and decision type $(d,\theta \rightarrow \kappa)$ begin in *S.* Premise types are divided into *stable* and *flexible*, and minimally an action rule has to have at least one flexible premise type. While the stable atomic components are part of the algorithm to enhance the confidence level, they are not the only ones from which action rules are built. In this case, a stable atom (*a*, *2*) is formed among the valid candidates. To validate atomic components of a flexible attribute, for example, the two sub-tables are checked through the domain. Atomic components of $(b, 2 \rightarrow 1)$ and $(b, 3 \rightarrow 1)$ means that "*2*", "*3*", and "*1*" are the values of attribute *b* in sub-tables $Ts_1$and Ts_2, respectively. This means that the values *2* or *3* of attribute *b* are replaced by the value *1*. The same operation can be applied for attribute c and its atomic components to extract the following action rules:

One-component set

// Values of a decision attribute in granular form Decision type atomic component: (d, $D\rightarrow H$), Granules: $L^* = \{x_1, x_2, x_4\}$, $R^*=\{x_3, x_5\}$

// Values of decision attributes

Expressions of premise type stable atoms

1. $(a,0)$, $L^*=\{x_1\}$, $R^* = \varphi$ Signed *"-"*.
2. $(a,2)$, $L^* = \{x_2, x_4\}$, $R^*= \{x_3, x_5\}$.

Expressions of the related premise type flexible atoms

1. $(b, 2\rightarrow 1)$, $L^* = \{x_1, x_2\}$, $suppL(p) = 2/7$; $R^*= \{x_3, x_5\}$, $suppR(p) = 2/7$; $Conf(p) = (2/2)\text{x}(2/2) = 100\%$ Signed *"+"*
2. $(b, 3\rightarrow 1)$, $L^* = \{x_4\}$, $suppL(p) = 1/7$; $R^* = \{ x_3, x_5\}$, $suppR(p) = 2/7$; $Conf(p) = (1/3)\text{x}(2/2) = 33\%$.
3. $(c, 0\rightarrow 2)$, $L^*= \{ x_1, x_2, x_4\}$, $suppL(r) = 3/7$; $R^* = \{x_3\}$, $suppR(p) = 1/7$; $Conf(p) = (3/3)\text{x}(1/1) = 100\%$ Signed *"+"*.
4. $(c, 0\rightarrow 1)$, $L^*= \{ x_1,x_2,x_4\}$, $suppL(p) = 3/7$; $R^* = \{x_5\}$, $suppR(p) = 1/7$; $Conf(p) = (3/3)\text{x}(1/2) = 50\%$.

The object-driven action rule, p, links each atom that is premise and decision types, which are favorable as long as the criteria of supporting conditions for the *supply(p), suppR(p),* and *Conf(p)* are suitable for user-specified set values. It is the idea that the algorithm prunes the atomic candidates that are unqualified and then it ties with the anti-monotonic procuring in the foundation. In our case, this condition is completed by a negative sign "-" as there is no confidence (satisfactory support) for an atomic action. In the one-frame set above, the atomic component *(a, 0)* does not qualify to meet the minimum requirement, and therefore it is not taken into account for the later steps. It is tagged with a minus sign. Another consideration of the algorithm is to get the shortest patterns. If there are any of the atomic action set combined with a decision type, it builds an actionable pattern with the condition that the atomic component is the shortest one, otherwise it is not considered to be picked. For two component stages, the rule can be processed as two unsigned premise-type atomic action sets with different attributes constructed by merging. For instance, in the expression, the atomic component $(b, 2\rightarrow 1)$ with the component $(d, D\rightarrow H)$ confirms all three onset values and, therefore, the action rule becomes $(b, 2\rightarrow 1)\Rightarrow (d, H\rightarrow D)$ and is identified further and, accordingly, a positive sign "+" is placed for the atomic component $(b, 2\rightarrow 1)$. Therefore, this is regarded as a good candidate. The following is the sequence of two atomic action stages in which three action rules are discovered.

Two-component set:

1. $(a, 2) \wedge (b, 3\rightarrow 1)$, $L^*=\{x_2\}$, $suppL(p)=1/7$; $R^*=\{x_3, x_5\}$, $suppR(p)=2/7$; $Conf(p) = (1/1)\text{x}(2/2) =100\%$ Signed *"+"*

2. $(a, 2) \wedge (c, 0\rightarrow 1)$, $L^*=\{x_2, x_4\}$, $suppL(p)=2/7$; $R^*=\{x_5\}$, $suppR(p)=1/7$;$Conf(p) = (2/2)\text{x}(1/1) = 100\%$ Signed *"+"*

3. $(b, 3\rightarrow 1) \wedge (c, 0\rightarrow 1)$, $L^*=\{x4\}$, $suppL(p)=1/7$; $R^*=\{x5\}$, $suppR(p)=1/7$; $Conf(p)=(1/1)\text{x}(1/1)=100\%$ Signed *"+"*

As noticed above in the set, all the predicate type atoms are tagged with a "+" sign, then the procedure is ended. In this process, five action rules are discovered as shown with their confidence levels below:

1. $(b, 2\rightarrow 1)\Rightarrow(d, D\rightarrow H)$, $suppL(p)=2/7$, $suppR(p)=2/7$, $Conf(p)=100\%$
2. $(c, 0\rightarrow 2)\Rightarrow(d, D\rightarrow H)$, $suppL(p)=3/7$, $supR(p)=1/7$, $Conf(p)=75\%$
3. $[(a,2)\wedge (b, 3\rightarrow 1)]\Rightarrow(d, D\rightarrow H)$, $suppL(p)=1/7$, $supR(p)=2/7$, $Conf(p)=100\%$
4. $[(a,2)\wedge (c, 0 \rightarrow 1)]\Rightarrow(d, D\rightarrow H)$, $suppL(p)=2/7$, $suppR(p)=1/7$, $Conf(p)=100\%$
5. $[(b, 3\rightarrow 1)\wedge (c, 0\rightarrow 1)]\Rightarrow(d, D\rightarrow H)$, $suppL(p)=1/7$, $suppR(p)=1/7$, $Conf(p)=100\%$

The primary idea of the object-driven algorithm for action rules lies in the anti-monotonic property of the support. The goal of this algorithm is to find the shortest action rules due to pruning for the qualification of the candidates. This is achieved by assigning a "+" sign when there is confidence or satisfactory support, and assigning a "-" sign where there is not enough confidence. Some of the previous studies of the test results and their implementations are available on public domains are given in Ref. (Ras *et al.,* 2008; He *et al.*, 2005)

EXPERIMENTAL STUDY

Action rules were presented in (Adomavicius *et al.,* 1997) and were formulated further in terms of certain pairs of classification methods (Ras *et al*., 2008). In the early stages of development, the action rules were too expensive and had some redundancies in the construction due to implementation of some satisfied pairs of classification rules. It was later demonstrated that action rules can be constructed via single classification methods to meet the final outcomes. In (*Ras et al.,* 2008) they suggested and formulated a simple rule of Learning from Examples Bases on Rough Sets (LERS)-type algorithm to build action rules of a single classification rule, which is named Action Rules Discovery Based on Agglomerative Strategy (ARAS). ARAS algorithm is formed on LERS that is a bottom-up approach in a breath-first manner to form all frequent item sets with a qualified part of length k, before forming those qualified part of length $k+1$. Although the ARAS algorithm is based on the related and non-related decision values of atomic components to produce clusters which lead to decision rules, it is not an object-driven algorithm that lies on constructing the shortest action rules to reach the final outcomes. More information on the application domain of an experiment of ARAS is available in (Dardzinska, 2013).

Experimental Design I: DEAR Database

Since object-driven action rules are based on the actionability of the DEAR database (Ras *et al*., 2005; Bobrowski, 1992), and therefore we made a comparison between the two algorithms. In the case of the DEAR algorithm, where the classical action rule extraction methods are implemented, there are four action rules generated, which are shown below:

1. $(b,2\to 1)\Rightarrow(d,\ D\to H)$, $suppL(p) = 2/7$, $suppR(p) = 2/7$, $Conf\,(p) = 100\%$
2. $(c,0\to 2)\Rightarrow(d,\ D\to H)$, $suppL(p) = 3/7$, $suppR(p) = 1/7$, $Conf\,(p) = 75\%$
3. $[(a,2)\ \wedge(c,0\to 1)]\Rightarrow(d,\ D\to H)$, $suppL(p) = 2/7$ *and* $suppR(p) = 1/7$, $Con\,f\,(p) = 100\%$
4. $(b,3\to 1)\Rightarrow(d,\ D\to H)$, $suppL(p) = 1/7$, $suppR(p) = 1/7$, $Con\,f\,(p) = 100\%$.

As presented above, for the object-driven action rule algorithm for which an example of a decision table is given, there are nine classification rules that were extracted as shown below:

1. $(a,0) \to (d,\ D)$	2. $(b,2) \to (d,\ D)$	3. $(c,0) \to (d,\ D)$
4. $(a,3) \to (d,N)$	5. $(b,1) \to (d,H)$	6. $(c,2) \to (d,H)$
7. $(a,2)\ \wedge(b,3) \to (d,\ D)$	8. $(b,3)\ \wedge(c,1)\to(d,\ N)$	9. $(a,2)\wedge(c,1)\to(d,H)$

Without the object-driven rule-extraction approach, the DEAR algorithm can only induce the actionable patterns from satisfied pairs of classification rule extraction, and thus their results are limited as compared to the object-driven new approach (Hajja *et al.*, 2014) and the illustration between object-driven and classification rules is depicted in Figure1 below.

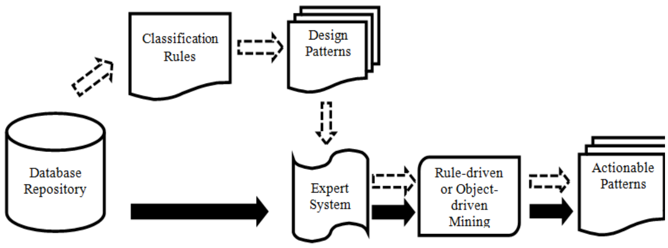

Figure1. Object-driven (solid arrows) and classical rule (dotted arrows) extraction approaches are depicted

Constructing an object-driven algorithm for Action Rules

Input: Object-driven algorithm (AlgOD) for discovering action rules can be built assuming the information (decision) table $S = (I,\ A,\ V)$ and where I is a set of objects, A is a set of attributes and V is a set of values of attributes, respectively. In addition, attribute A is split into stable (A_{St}), flexible (A_{Fl}), and decision *(d)* attributes, respectively. Further, *AlgOD* $[S,\ \omega_1,\ \omega_2,\omega_3,R(p)]$ is constructed where $\omega_1,\ \omega_2,\ \omega_3$ are for minimum set values of right support (*suppR*), left support (*suppL*), and confidence (*confd*), and *R(p)* depicts a set of rules produced by *AlgOD* through the decision system, $S = (I,\ A_{St}\cup A_{Fl}\ \cup\{d\},\ V)$.

```
//Function of Action Rule Instantiation()
  for (i = 0; i < maxAtoms)
    pi = combine premise-type with decision-type atoms and
      compute its suppL, suppR, and confd
      If (suppL >= ω1) Then
        If (suppR >= ω2) Then
    If (confd >= ω3) Then
          R(p) = add Rules Produced by AlgOD(pi)
     Discard the atom from the list
    Else
          Discard the premise atoms from left cluster of the
     list
       End If
     Else
    Discard the premise atoms from right cluster of the list
      End If
Else
  End for loop
End Function
//Main Method
Function Main()
//Split the table
If S has more than 1 decision value, then
  K= SplitS()
  Ks = Chose split sub-tables (Ks)
  Show metadata of AlgOD
//Enter the inputs of user to designate ASt, AFl, d, and make the
reclassification
  getInput() //ASt, AFl, d types atoms entered
//Instantiation of terms
 n=InstantiatedConditions()
 flag=true
 While(Flag)
    If n.size > 0 Then
      ProduceActionRules(n)
      ProduceUnifiedComponents()
    else
      flag=false
    End if else
   End While
End Main()
```

Output: Set of concise object-driven action rules displayed below

Figure 2. AlgOD object-driven algorithm

Experimental Design II: HEPAR Database

As described above, the objected-driven approach was also empirically tested on the computer-based HEPAR database for implementing and processing medical data sets of patients who had liver disorders. This clinical tool was established to reduce hepatic biopsies. The experimental procedure was conducted on several medical databases available in the public domains (Dardzinska *et al*., 2006; Wasluk *et al.*, 2002). The HEPAR database contains 758 records of patients along with 106 attributes, as well as 31 laboratory exams with values discretized as: *"normal*", "*below normal*", and "*above normal*". Among those 14 stable attributes, the two medical tests were invasive that related to hepatitis B surface (HBsAg) and core (HBcAg) antigens.

The following denotations were for values of the attributes that made the determination on the tests: *I* for acute hepatitis, *IIa* for sub-acute hepatitis of types B and C, *IIb* for sub-acute hepatitis with alcohol-abuse, *IIIa*for curable chronic hepatitis, *IIIb* non-curable chronic hepatitis, *IV* for cirrhosis hepatitis, and *V* for liver cancer. It is worth noting that a complete diagnosis in liver disease relies on physical examinations, radiological and laboratory tests, background (history), and, if needed, biopsies (invasive) from a patient. While radiographic examinations may detect the presence of liver diseases, no tests can be treated as a diagnostic standard. However, ultrasonography is usually the first radiographic examination used in the assessment of liver diseases. Ultrasounds are noninvasive, commonly available, and there is no risk of radiation or intravenous contrast with potential for nephrotoxity (Wasluk *et al*., 2006; Heidelbaugh *et al.*, 2006). On the other hand, there can be some risks involved in liver biopsy, such as prolonged bleeding, infection near biopsy site, and injury to nearby organs. Nevertheless, liver biopsy becomes only option to establish an accurate diagnosis and treatment of chronic liver disease as gold standard.

In action rules, a medical treatment of a patient's record is generally reclassified from one stage to another. The HEPAR database contains missing values in which more than 90% percent of all the attributes were removed with the assumption that there was no linkage to any invasive exams, such as biopsy (Wasluk *et al*., 2006; Ras *et al.*, 2003). In this study, the main concentration is to reclassify the conditions of the patients who were diagnosed with liver disease from class *IIB* to class *I* and from class *IIIA* to class *I,* with the exception of invasive examinations in action rules. In addition, subjective attributes such as history of alcohol abuse were eliminated by implementing the classical null value imputation techniques.

In the testing stage, we used the RSES software to discover d-reducts as follows:

Given that the *set R = {m, n, q, u, y, aa, ah, ai, am, an, aw, bb, bg, bm, by, cj, cm}* and does not include any invasive exams.The following definitions show the values of the attributes:

m (bleeding), q (eructation), n (Subjaundice symptoms), u (obstruction), y (weight loss), aa (smoking),ah (history of viral hepatitis –stable),ai (surgeries in the past –stable), am (history of hospitalization – stable), an (jaundice in pregnancy), aw (erythematous dermatitis), bb (cysts), bg (sharp liver edge – stable) bm (blood cell plaque), by (alkaline phosphatase), cj (prothrombin index), cm (total cholesterol), dd (decision attribute).

The medical datasets were also tested using the Action Rules Discovery Based on Agglomerative Strategy (ARAS) algorithm, which constructs action rules from the values of attributes, which is detailed in (Ras *et al*., 2008; Dardzinska, 2013). In their study, there were two action rules identified by implementing ARAS through the database confined to eliminated-reduct R. The confidence level of the rules was not mentioned. The object-driven algorithm implemented to generate eight action rules and among three are with the very high confidence level as shown below:

1. $[(am, 2) \wedge (ah, 2) \wedge (bg, 2)] \wedge (q, 2 \rightarrow 1) \wedge (cm, 2 \rightarrow 1) \Rightarrow (dd, IIIA \rightarrow I)$

This rule with the highest confidence relates to a patient that has a history of hospitalization with viral hepatitis with condition of a sharp liver edge in abnormality. It states that if the eructation is eliminated and the level of cholesterol is reduced to *normal*, then the condition of the patient is reclassified from class *IIIA* to class *I.*

2. $[(am, 2) \wedge (bg, 2) \wedge (ai, 1)] \wedge (u, 2 \rightarrow 1) \wedge (y, 2 \rightarrow 1) \Rightarrow (dd, IIIA \rightarrow I)$

The second rule with highest confidence corresponds to a patient who has a history of hospitalization with a sharp liver edge in abnormal condition without any past surgeries. It states that if the obstruction is eliminated and weight loss is improved to *normal*, then the condition of the patient is reclassified from class *IIIA* to class *I*.

3. $[(am, 2) \wedge (bg, 2) \wedge (ai, 1)] \wedge (q, 2 \rightarrow 1) \wedge (u, 2 \rightarrow 1) \wedge (n, 2 \rightarrow 1) \Rightarrow (dd, IIIA \rightarrow I)$

The third rule with highest confidence describes that a patient who has a history of hospitalization with a sharp liver edge in abnormal condition without any surgeries in the past. It states that if the eructation and obstructions are eliminated and the subjaundice improves to *normal*, then the patient is reclassified from class *IIIA* to class *I*.

CONCLUSION AND FUTURE WORK

This paper discusses and presents a new approach in which action rules can be extracted in terms of object-driven perspectives that are related to liver diseases in the healthcare domain of hepatology. It identifies action rules, described as knowledge-based logical terms linked to objects, where they are identified and extracted from an information system that makes the decision. In other words, the diagnostics of individual patients are harmonized and paired to a computerized knowledge-based system and algorithms that construct patient-specific suggestions for possible treatments. By controlling onset values of the rule extraction, one can combine and sift through extracted patterns into the global actionable rules by means of object-driven avenue of action ability for decision makers and patient relationships from the data about patients, which promotes robust outcomes as compared to classical methods.

The obligation on this decision support system has to focus on conformity of the operational expert system due to its instruction and accuracy of the knowledge. For future work, how the value of an attribute effects the system in regard to the support, confidence, and time cost is laid upon should be explored. In addition, the full optimization of the object-driven algorithm can be tested rigorously for much lower computational complexity for clinical conformity, especially accuracy of diagnoses, in other areas of the medical domain. In addition, rigorous error analyses need to be studied. These include missing a decision scenario or incorrect logic, along with coding errors, such as bugs in the developed software.

Acknowledgements

We are indebted to Prof. Dr. Z W Ras whose invaluable advice and suggestions.

REFERENCES

Adomavicius, G., and Tuzhilin, A. 1997. Discovery of actionable patterns in databases: The action hierarchy approach. Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, The AAAI Press. Aug. 14-17; pp. 111-114.

Agrawal, R., and Srikant, R. 1994. Fast algorithm for mining association rules. Proceeding of the 20th International Conference on very large database. Sep. 12-15; pp. 487-499.

Agrawal, R., Imielinski, T., and Swami, A. 1993. Mining association rules between sets of items in large database. Proceedings of ACM SIGMOD International Conference on Management of Data. May 25-28; pp. 207-216.

Bobrowski, L. 1992 HEPAR: Computer system for diagnosis support and data analysis. Prace IBIB 31, Institute of Biocybernetics and Biomedical Engineering, Polish Academy of Sciences, Warsaw, Poland.

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

Dardzinska, A. 2013 Action rules mining. Studies in Computational Intelligence, Springer Publication, Springer-Verlag, Germany.

Dhar, V., and Tuzhilin, A. 1993. Abstract-driven pattern discovery in databases. IEEE Tran06s Knowledge and Data Engineering. 5(6), 926-938.

Frawley, W. J., Piatetsky-Shapiro, G., and Matheus, C. J. 1991. Knowledge discovery in databases: An Overview, Knowledge Discovery in Databases, G. Piatetsky-Shapiro and W.J. Frawley, eds.. AAAI/MIT Press. pp. 1-27.

Hajja, A., Ras, Z. W., and Wieczorkowska, A. 2014. Hierarchical object-driven action rules. *J. Intell. Inf.Syst.* 42 (2), pp. 207-232.

He, Z., Xu, X., Deng, S., and Ma, R. 2005. Mining action rules from scratch. Expert Systems with Applications 29(3), pp. 691–699.

Heidelbaugh, J. J., and Bruderly, M. 2006. Cirrhosis and chronic liver failure: part I diagnosis and evaluation. Am Fam. Physician. 74(5), pp. 756-762.

Pawlak, Z. 1981. Information systems - theoretical foundations.*Information Systems Journal.* 6, 205-218.

Ras, Z. W., and Dardzinska, A. 2006. Action rules discovery- a new simplified strategy. Foundations of Intelligent Systems, Esposito F. *et al*. (Eds.), LNAI, No. 4203, Springer. pp. 445-453.

Ras, Z. W., and Dardzinska, A. 2008. Action Rules Discovery without pre-existing classification rule. Proceedings of RSCTC 2008 Conference, in Akron, Ohio, LNAI 5306, Springer. Oct. 23-25; pp. 181-190.

Ras, Z. W., and Tsay, L. S. 2003. Discovering extended action-rules (System DEAR). Intelligent Information Systems, Proceedings of the IIS' 2003 Symposium, Advances in Soft Computing, Springer. Nov. 6-8; pp. 293-300.

Ras, Z. W., Tzacheva, A., Tsay, L. S., and Gurdal, O. 2005. Mining for interesting action rules. Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology, Compiegne University of Technology, France. Sep. 19-22; pp. 187-193.

Ras, Z. W., Wyrzykowska, E., and Wasyluk, H. 2008. ARAS: Action rules discovery based on agglomerative strategy. Mining Complex Data, Post-Proceedings of 2007 ECML/PKDD Third International Workshop (MCD 2007), LNAI., Springer. Sept. 15-19; 4944, pp. 196-208.

Silberschatz, A., and Tuzhilin, A. 1996. What makes patterns interesting in Knowledge Discovery Systems. IEEE Transactions on Knowledge and Data Engineering. 5, pp. 970-974.

Wasyluk, H. 2002. Second Evaluation of the Diagnostic Computer System HEPAR. Proceedings 6th World Multiconference of Systemic, Cybernetics and Informatics Orlando, USA, ed. N. Callaos. 2, July 14-18; pp. 242-246.

Wasyluk, H., Onisko, A., and Druzdzel, M. 2006 Support of diagnosis of liver disorders based on a causal Bayesian network model. Med. Sci. Monitor No 7, suppl. 1, pp. 327-332.

****