Object–Driven Action Rules

C

Ayman Hajja
University of North Carolina, USA

Zbigniew W. Ras
University of North Carolina, USA & Warsaw University of Technology, Poland

INTRODUCTION

In recent years, data mining techniques have been extensively employed in numerous domain areas; applied to topics from fields such as syndromic surveillance, where patterns of measuring behavioral risk factors, localizing illness by geographical regions, and analyzing symptoms and medication usage have been studied (Paul & Dredze, 2011); politics, where text mining by sentiments was exploited to measure political public opinion to predict election results (Bermingham & Smeaton, 2011); and linguistics, where lexical variations across geographical regions were studied to recover coherent topics and their regional variations while identifying geographical areas of linguistic consistencies (Eisenstein, O'Connor, Smith, & Xing, 2010); and many others.

In this article, we explore concepts in the subcategory of data mining known as rule-based knowledge discovery. The high interpretability of rule-based methods has given them an intense popularity, allowing domain experts such as physicians to easily interpret the patterns discovered; hence, permitting them to gain insights, or better yet, make alterations to fit their needs. Action rule mining; proposed by Ras and Wieczorkowska (2000) in particular, has been heavily investigated in recent years, where the goal is to provide system users with actionable tasks that can be directly applied to attributes to reach a desired goal. Since its introduction in 2000, action rules have been successfully applied in many domain areas including business (Ras & Wieczorkowska, 2000), medical diagnosis and treatment (Wasyluk, Ras, & Wyrzykowska, 2008; Zhang, Ras, Jastreboff, & Thompson, 2010), and music automatic indexing and retrieval (Ras & Wieczorkowska, 2010; Ras & Dardzinska, 2011). In contrast to association rule learning (Agrawal, Imieliski, & Swami, 1993), action rule approaches mine actionable patterns that can be employed to reach a desired goal, instead of only extracting passive relations between variables.

BACKGROUND

The interpretation of object-driven action rules was first proposed by Hajja, Wieczorkowska, Ras, and Gubrynowicz (2012). The form of complex dataset that we exploit contains various instances for the same object, and a temporal aspect coupled with each instance. A common example of this structure is frequently found in medical data, where for each unique patient (representing an object), multiple visits are recorded, and where in each visit (representing an instance) a timestamp is associated with the instance.

OBJECT-DRIVEN ACTION RULES

In this section, we start by providing the reader with the necessary background to shape a complete picture behind object-driven action rules and its motivation. Classical action rules are introduced, formally defined and fully explained, to the extent of preparing the reader with the necessary background to capture its relation to object-driven action rules, by the similarities of both, and the novelty of each.

Following the background, we move our discussion to object-driven action rules; starting by clearly communicating to the reader the properties of datasets that work for action rules concepts and by fully explaining the assumptions we make for our adapted system.

DOI: 10.4018/978-1-4666-5888-2.ch114

(Classical) Action Rules

The concept of action rules was first proposed by Ras and Wieczorkowska (2000). It describes possible transition of objects from one state to another with respect to a specific attribute, called the decision attribute. The goal of action rules is to provide system users with actionable tasks that can be directly applied to objects listed in information systems to reach a desired goal.

Let $S = (X, A, V)$ denotes an information system (Pawlak, 1981), where:

1. X is a nonempty, finite set of instances,
2. A is a nonempty, finite set of attributes; $a : X \rightarrow V_a$ is a function for any $a \in A$, where V_a is called the domain of a,
3. $V = \bigcup \{V_a : a \in A\}$.

Despite the fact that elements of X are sometimes referred to as objects, in this note we will not use the two terms interchangeably, objects will possibly consist of, as will be discussed in future sections, multiple instances.

By a decision table, we mean an information system that makes a clear explicit distinction between attributes in A, and will therefore partition every attribute into either a *decision attribute*, or a non-decision attribute; called *condition attribute*. The decision attribute(s), normally but not necessarily is a single attribute, is the attribute that we are interested in the most. For system users, the eventual goal would be to change the decision attribute from less desirable, to more desirable state. For example, a company would be interested in moving clients' states of loyalty from lower to higher.

All non-decision, or condition, attributes are further partitioned into two mutually exclusive sets; first one is *stable* attributes set, and the second one being the *flexible (*or *actionable)* attributes set. By stable attribute set we refer the set that contains attributes that we have no control over; their values cannot be changed by the users of our system. An example of a *stable* attribute is the age of a patient. On the other hand, values of a *flexible* attribute can be influenced and changed; an example of a *flexible* attribute is the patient's prescribed medications. In this article, A_{St}, A_{Fl}, and $\{d\}$ will represent the set of stable attributes, the set of flexible attributes, and set of decision attribute(s), respectively. Hence, the set of attributes A can be represented as:

$$A = A_{St} \bigcup A_{Fl} \bigcup \{d\}.$$

An *atomic action set* is an expression that defines a change of state for a distinct attribute. For example,

$$(a, a_1 \rightarrow a_2)$$

is an atomic action set which defines a change of the value of attribute a from to a_2, where $a_1, a_2 \in V_a$. Clearly in this case, attribute a is a flexible attribute, since it changes its state from a_1 to a_2. In the case when there is no change, we omit the right arrow sign, so for example, (b, b_1) means that the value of attribute b remains b_1, where $b_1 \in V_b$.

Action sets are defined as the smallest collection of sets such that:

1. If t is an atomic action set, then t is an action set.
2. If t_1, t_2 are action sets and "$\wedge$" is a 2-argument functor called composition, then $t_1 \wedge t_2$ is a candidate action set.

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.

The *domain* of an action set:

$$t, \text{denoted by } Dom(t),$$

is the set of attributes of all the atomic action sets contained in t. For example,