Action Rule Mining With Predications on Semantic Representations of Data

Li-Shiang Tsay
North Carolina A&T State Univ., College of Sci. & Tech., Greensboro, USA
ltsay@ncat.edu

Abstract: Without a doubt, the World Wide Web (WWW) has already altered the way we share knowledge. The exploitation of the web is one of the biggest challenges in the area of intelligent information management. The Semantic Web was presented to ease this situation by taking the WWW into a distributed global system for knowledge representation and computing. In this work, we study action rule mining in the semantic web data and aim to assist users in uncovering previously unknown and potentially useful workable strategies. We present rule-based action rules and object-based action rules for extracting actionable patterns from a large graph dataset without the extra step of converting graph data into transaction data.

Keywords: action rules, semantic association, semantic web data

1. Introduction

Semantic web data is a vital repository for many applications, and its volume is continuously growing. It has become a rich source of useful data for developing approaches to make machines capable of reasoning from unstructured or semi-structured data and providing personalized answers. In fact, in just twenty years, we now have devices like Amazon's Alexa that utilize the Semantic web to perform tasks like booking a table at our favorite restaurants. Semantic Web technologies have made significant advancements. Information can be processed faster and at a finer level of access and semantic granularity that can be shared via web services. At the same time, new challenges arise in knowledge discovery and presentation. A new way of reasoning the meaning of content on the web is needed.

Finding actionable insights from data has always been a difficult task. Action Rule Mining (ARM) (Raś & Wieczorkowska , 2000), (Ras & Tsay , 2003) is one of the most promising technologies for extracting patterns that recommends actions leading to desirable outcomes, by giving only previously acquired data. ARM is the process of finding workable strategies for transiting objects from an undesirable state into a desirable state. Specifically, it focuses on the usability of mined patterns. An action rule is represented as $[(X,\alpha\rightarrow\beta) \Rightarrow (Y,\phi\rightarrow\psi)]$, where $(X,\alpha\rightarrow\beta)$ represents a formulated specific action, $(Y,\phi\rightarrow\psi)$ is the estimated effect obtained from the action, $\rightarrow$ means change, and $\Rightarrow$ means imply. It summarizes two populations P_1 and P_2 having properties $\{(X,\alpha), (Y,\phi)\}$ and $\{(X, \beta), (Y, \psi)\}$, respectively. P_1 can be influenced by actions $(X,\alpha\rightarrow\beta)$, the value of feature X changes from α to β, toward a higher preferable population P_2. This rule presentation is in a symbolic notation, which is very amenable to inspection and interpretation. This characteristic allows business end users and analysts to understand the underlying cause-effect in data, and takes action based upon them. ARM has been demonstrated to be of significant value in a variety of real-world data mining applications, such as customer attrition (Tarnowska & Raś, 2018), artwork price strategy (Powell, et al., 2020), reduction of hospital readmission (Mardini & Raś, 2019), etc. ARM techniques range from a post-process analysing rule manner to a process scrutinizing row data way. These techniques are automatic methods for locating important facts for influencing objects' behaviors from structured data.

In 2022, 2.5 quintillion bytes of data are created every day. About 80 percent of all generated data are unstructured data. As the forms and volume of data evolved, the task of unearthing valuable actionable knowledge for decision-making has become even more challenging. In the existing literature, various ARM approaches have been discussed, but these methods construct the action rules from either an IF-THEN rule set or a structured dataset. In this work, we study action rule mining in the semantic web data and aim to assist users in uncovering previously unknown and potentially useful cause-effect associations. The structure of this paper is organized as follows. The previous approaches to finding semantic association in the Semantic Web are discussed in Section 2. In Section 3, the theoretical foundation for semantic association is presented. The proposed semantic action rules and their interestingness measurements are introduced in Section 4. In Section 5, our conclusions are presented.

2. Related work

In the late 1990s and early part of the twenty-first century, the Semantic Web was presented to take the World Wide Web (WWW) beyond keyword search, and develop it into a distributed global system for knowledge representation and computing (Berners-Lee, et al., 2001)(Berendt, et al., 2003). Such a system is built by using ontologies and globally unique identifiers to weave together related data and services. Unstructured and semi-structured data is transformed into a "Web of Data", in which both nodes and links are uniquely identified by Uniform Resource Identifiers (URI). The representation of web-information is regulated by standardized syntax, common vocabularies and knowledge, and a logical language. Basically, ontological terms are explicitly added to web data to provide logical pieces of meaning that can be automatically manipulated by machines. Meaning is expressed by Resource Description Framework (RDF), which encodes it in a set of triples (subject-predicate-object). The triples of RDF form webs of information about related things.

For Semantic Association, most studies utilized association rules to support the knowledge acquisition process by defining metadata information (Savasere, et al., 1995) (Völker & Niepert , 2011) (Jiang & Tan, 2006). In the 2011 Billion Triple Challenge, to discover common usage patterns in Linked Open Data, both positive and negative association rule mining were used. The latter compared these patterns to existing schema definitions to indicate potential modeling errors (Jimenez & Goodman, 2012). Based on a predefined mining pattern, complex semantic data can be converted into transactions before using an association rule method to learn the causal relations in the medical domain (Kochut & Janik, 2007). Closed itemset mining was utilized to extract the shared conceptualization from a folksonomy, which is the core data structure of a social resource sharing system (i.e., Flicker) (Lorey, et al., 2011). One approach was using generalized association rules to mine frequent patterns from large sets of RDF triples. Other studies use generalized association rules to analyze statistical information about the linguistic data for enriching ontology (Maedche & Staab, 2000), for detecting a context of schema/ontology matching (Ramezani, et al., 2014), and semantic annotations. In most of the cases, a generalized association analysis is utilized to facilitate engineering ontologies (Srikant & Agrawal, 1995), ontology matching (Ramezani, et al., 2014), building property axioms (Kiefer, et al., 2008) and schema analysis (Savasere, et al., 1995). Other cases use them for optimizing RDF storage, improving query processing, semantic annotations, and enhancing information for related recommendations.

Class association rules (Liu, et al., 1998) are a special subset of association rules whose consequences are restricted to predefined target labels. This makes mining associations more applicable and practical in web-scale data. Like frequent closed itemsets in the association rule mining for reducing the number of redundant discovered rules, closed factor-sets can be used to generate class association rules. Beside support and confidence, other interestingness measures *lift* and *coverage* are necessary to filter out trivial patterns. In addition, we believe that it is vital for a user to be involved in the mining process in order to extract meaningful rules from a large dataset.

3. Semantic Association

An information system is used for representing knowledge $U = \{E \cup R \cup L\}$, where E is an entity, R is a relationship, and L is a literal. An *entity* is any tangible and intangible object in the world, such as a *composer*, an *author*, a *song*, or a *novel*. A *relationship* is an association between entities, for example a particular relationship *writes*. A *literal* is any value of an entity, for example Johann Sebastian Bach. Within the RDF data model, information is expressed as a set of binary propositions and is represented by facts denoted as triples consisting of a subject, a predicate, and an object. A RDF triple s is represented as $s \in \{ E, R, (E \cup L)\}$. Its first component (the subject) stands in the relation given by the second component (the predicate) with the third component (the object), as in {*composer*, *writes*, *song*} and {*author*, *writes*, *novel*}. The terms used in an RDF tuple are relative URLs in a pair of angle brackets and literals in a pair of quotation marks. Literals are typed data values that can be used and located at the object position. Each triple basically established a link between the entity recognized by the subject with the entity identified by the object via the predicate (Auer, et al., 2011).

3.1 Semantic Association rules

Association rules are applicable in analysis of RDF stores, as every edge of the graph has a unique combination of vertex and edge labels. Analogous to traditional transaction data, the subject s of a triple tuple can be considered as a "transaction ID" and the combination of its corresponding predicate p and object o can be

seen as an "attribute-value" pair. Like an item, each pair of predicate *p* and its corresponding object *o* can be called a *Factor,* i.e., $f = \{ p\ o \}$. Let $F = \{ f_i \mid i = 1, \ldots, n \}$ is a set of distinct factors in the dataset. Any set of factors in *F* is called a factor-set. With these mappings, we can treat a RDF store *U* as a collection of *m* data cases, $U = \{c_i \mid i = 1, \ldots, m\}$. Each case c_i has a unique *subject ID* (*sid*) and contains a subset of factors in *F*, i.e., $c_i = \{sid_i, f_{i_j} \mid j = 1, \ldots, q\}$.

By a RDF store, we mean any information system $U = \{C, P\}$, where:

- *C is* a nonempty and finite set of cases
- *P* is a nonempty and finite set of properties, . i.e. $p : U \rightarrow O_p$ is a function for any $p \in P$ where O_p is called the domain of *p*.

Elements of *U* are called cases. In this section, for the purpose of clarity, cases are interpreted as patients. Properties *P* are interpreted as attributes such as diagnosis made by a doctor, characteristic of a tumor status, etc. As we mentioned before, a pair of property-value is called a *Factor*.

A pattern *X* is a subset of a case, $X \subseteq U$. A pattern with *k* factors is called *k*-pattern. The support of a pattern *X* is the ratio of the number of cases containing *X* to the number of all cases in *D*, denoted by *sup*(*X*). An *Association Rule* is an implication of the form $X \rightarrow Y$, where $X \neq \emptyset$, $X \subseteq F$, $Y \subseteq F$, $X \cap Y = \emptyset$. *X* is called antecedent and *Y* is a called the consequent of the rule. XY is a frequent factor-set. Strong association rules are derived from frequent factors. The support of the rule is as $sup(X \cup Y)$ and the confidence of the rule is defined as $conf(X \rightarrow Y) = sup(X \cup Y)/sup(X)$.

There is an old saying that 99% of the web information is useless to 99% of web users. Instead of searching the entire data space for every possible association rule, an alternative way of finding relevant materials from the Web is to set a restriction on the search focus according to the user's preference. Adding a constraint to limit the factors that can appear on the consequence of the rule is called a Class Association Rule (CAR). Only a small portion of data space is required for finding such rules. The computation complexity and the number of trivial rules could be considerably reduced.

3.2 Semantic Class Association rules

A decision table consists of a set of cases where each case is described by a set of properties. Properties are partitioned into premise and target. Additionally, we assume that the set of premises is portioned into stable features and flexible features. For simplicity, we assume that there is only one target property. For a medical dataset, "diagnosis" can be the target attribute. Its domain is defined as a set of literals. The target attribute classifies cases with respect to the diagnosis by a physician at a hospital. *Age* of a patient is an example of a stable attribute. The *treatment* on a patient is an example of a flexible attribute as the physician can adjust it.

Let $U = \{C, P\}$ be an information system. If there exists Ps , P_f, $P_t \subseteq P$, such that $P_s \cap P_f \cap P_t = \emptyset$ and $P_s \cup P_f \cup P_t = P$, then *U* is called a decision table. A decision table is denoted it as $U = \{C, P_s \cup P_f \cup P_t \}$, where *C* is a nonempty and finite set of cases, P_t is a distinguished property called a target class, P_s is called stable premise properties, and P_f is called flexible premise properties. The set of factors *F* in *U* can be partitioned into premise factors F_c and target factors F_t. F_t is a targeted predicate p_t with a set of its associated distinct object values, $F_t = \{ p_t, O_{t_j} \mid j = 1, \ldots, q\}$. Assume that $U = \{(c_1, c_2, c_3, c_4, c_5, c_6), (A , B) \cup (T) \}$ is a decision table represented by Table I. It consists of 6 cases $\{c_1, c_2, c_3, c_4, c_5, c_6\}$. The predicates in $\{A, B\}$ are premise one and predicate *T* is the target predicate. The minimum support $sup(r)$ is 15% and the minimum confidence $conf(r)$ for rules is 60%.

Table 1: Example of a decision RDF store

S	P	O
c_1	*B*	b_1
c_1	*A*	a_3
c_1	*T*	t_2
c_2	*A*	a_2
c_2	*A*	a_3
c_2	*T*	t_1
c_2	*B*	b_3

S	P	O
c_3	*B*	b_3
c_3	*A*	a_3
c_3	*T*	t_1
c_4	*A*	a_3
c_4	*T*	t_1
c_5	*B*	b_1
c_5	*T*	t_2
c_6	*B*	b_3
c_6	*T*	t_1

A decision system *U* classifies a set of cases so that for each object there exists a class label assigned to it. A *class association rule r* in *U* can be expressed as: $r = X \rightarrow Y$, where $X \neq \emptyset$, $X \subseteq F_C$, $Y \in F_t$, $X \cap Y = \emptyset$. The antecedent *X* of the rule is a set of premise *factors* and the consequent *Y* is the target used to characterize interesting segments of the populations and must be specified by a user. Referring back to the example, four strong class association rules are constructed. These rules and their support and confidence are presented in Table 2.

Table 2: A set of Class Association rules with their interestingness measures.

Class Association Rule	*sup*(*r*)	*conf*(*r*)
(B, b_3) → (T, t_1)	50.0%	75.0%
(A, a_3)→ (T, t_1)	50.0%	75.0%
(B, b_1) → (T, t_2)	33.3%	100.0%
(B, b_3) (A, a_3) → (T, t_1)	33.3%	100%

4. Semantic action rules

The basic principle of action rule mining is a process of learning a function that maps one class of objects into another class by changing values of some conditional features describing them. The conditional features are divided into stable and flexible. The goal of the learning process is to create a transition model, for objects in a decision table, which suggests possible changes that can be made within values of some flexible attributes to influence these objects the way user wants. A decision system *S* classifies a set of objects so that for each object there exists a class label assigned to it. Action rule mining is the process of showing what changes in values of some of the flexible attributes for a given class of objects are needed in order to shift them from one decision class into another more desired one.

There are two possible approaches for building action rules, object-based and rule-based. In object-based action rules, action rules are directly constructed from the decision table by taking all possible pair combinations of flexible items. In Rule-based action rules, action rules are built from certain pairs of class association rules extracted earlier from the same decision table. Several definitions related to action rules, support and confidence the rules from structured data are proposed in [Tsay and Ras 2005]. They have been extended for an RDF store and listed in the sections below.

4.1 Object-based action rules

By *action rule r* in *R* we mean an expression

$$r = [[(P_s1 = \omega_1) \wedge (P_{s2} = \omega_2) \wedge \ldots \wedge (P_{sm} = \omega_m)] \wedge (P_{f1}, \alpha_1 \rightarrow \beta_1) \wedge (P_{f2}, \alpha_2 \rightarrow \beta_2) \wedge \ldots \wedge (P_{fn}, \alpha_n \rightarrow \beta_n)] \Rightarrow [(P_t, k_1 \rightarrow k_2)],$$

where $\{P_{f1}, P_{f2}, \ldots, P_{fn}\}$ are flexible premise and $\{P_{s1}, P_{s2}, \ldots, P_{sm}\}$ are stable in *R*.

Additionally, we assume that $\omega_i \in Dom(P_{si})$, $i=1,2,\ldots,m$ and $\alpha_i, \beta_i \in Dom(P_{fi})$, $i=1,2,\ldots,n$. The term $(P_{si} = \omega_i)$ states that the value of the attribute P_{si} is equal to ω_i, and $(P_{fj}, \alpha_j \rightarrow \beta_j)$ means that value of the attribute P_{fj} has been changed from α_j to β_j.

We say that object $x \in U$ supports an action rule r in R, if there is an object $y \in U$ such that: $(\forall j \leq n)[[P_{fj}(x) = \alpha_j] \wedge [P_{fj}(y) = \beta_j]]$, $(\forall i \leq m)[P_{si}(x) = P_{si}(y) = \omega_i]$, $P_t(x) = k_1$ and $P_t(y) = k_2$.

An action rule is meaningful only if it contains at least one flexible feature. If we apply the left hand side of an action rule to object *x,* then the rule basically says: the values ω_i of stable attributes P_{si} (i=1,2,…,*m*) have to remain unchanged in *x*; if we change the value of attribute P_{fj} in *x* from α_j to β_j, for *j*=1,2,…, *n*, then the object *x* which is in the class k_1 is expected to transition to class k_2.

From the point of transition, we are not targeting all possible cases on the decisional part of changes. Since some states are more preferable than other states, we should basically ask users to specify in what direction they prefer to see the changes. On the conditional part of action rules, we have no information to verify if the rule is applicable. If the domain expert can supply prior knowledge of a given domain, then some of the rules cannot be applied. For example, the size of a tumor's growth can not increase when the status of a patient has transitioned from sick to becoming cured. Therefore, some combinations can be ruled out automatically just by having an expert who is involved in the application domain.

Since an action rule is constructed by comparing the profiles of two sets of targeted objects, we can assume that there are two patterns associated with each action rule, a left hand side pattern r_L and a right hand side pattern r_R. There are three objective measures of rule interestingness including *Left Support*, *Right Support*, and *confidence*.

The ***Left Support*** defines the domain of an action rule which identifies objects in *U* on which the rule can be applied. The larger its value is, the more interesting the rule will be for a user. The left hand side pattern of action rule

$$r = [[(P_{s1} = \omega_1) \wedge (P_{s2} = \omega_2) \wedge \ldots \wedge (P_{sm} = \omega_m)] \wedge (P_{f1}, \alpha_1 \rightarrow \beta_1) \wedge (P_{f2}, \alpha_2 \rightarrow \beta_2) \wedge \ldots \wedge (P_{fn}, \alpha_n \rightarrow \beta_n)] \Rightarrow [(P_T, k_1 \rightarrow k_2)],$$

is defined as the set $r_L = V_L \cup \{k_1\}$, where $V_L = \{\omega_1, \omega_2, \ldots, \omega_m, \alpha_1, \alpha_2, \ldots, \alpha_n\}$. The domain $Dom(V_L)$ of the left pattern r_L is a set of objects in *U* that exactly match V_L. Card[$Dom(V_L)$] is the number of objects in that domain. Card[$Dom(r_L)$] is the number of objects in *U* that exactly match r_L and Card[*U*] is the total number of objects in the decision system *U*. By the left support *supL* of an action rule *r*, we mean $supL(r)$ = Card[$Dom(R_L)$] /Card[*U*].

The ***Right Support*** shows how well the rule is supported by objects in *U* from the preferable decision class. The higher its value is, the stronger case of the transition effect will be. The pattern r_R of an action rule *r* is defined as $r_R = V_R \cup \{k_2\}$, where $V_R = \{\omega_1, \omega_2, \ldots, \omega_m, \beta_1, \beta_2, \ldots, \beta_n\}$.

By domain $Dom(V_R)$ we mean a set of objects matching V_R. Card[$Dom(r_R)$] is the number of objects that exactly match r_R. By the right support *supR* of action rule *r*, we mean $supR(r)$ = Card[$Dom(r_R)$] /Card[*U*].

The ***confidence*** of **action rule *r*** shows the success measure in transforming objects from a lower preference decision class to a higher one. The *support* of action rule *r* in *R,* denoted by *Sup*(r), is the same as the left support $supL(r)$ of action rule *r*. This is the percentage of objects that need to be changed into a more preferable class. By the *confidence* of the action rule *r in R*, denoted by $Conf(r)$, we mean

$Conf(r)$=(Card[$Dom(r_L)$]/Card[$Dom(V_L)$])∗(Card[$Dom(r_R)$]/Card[$Dom(V_R)$]).

Referring back to the example, we assume that attribute *A* is stable, attribute *B* is flexible, and *T* is the target attribute. The values of attribute *T* are "t_1" and "t_2" in *U*. The preferable change is from t_1 to t_2. The minimum support for both *supR* and *supL* is 15%, and the minimum confidence for rules is 60%. Two action rules have been constructed. These rules and their support and confidence are presented in Table 3.

Table 3. A set of action rules with their interestingness measures.

Action Rule	*supL*(*r*)	*supR*(*r*)	*conf*(*r*)
$(B, b_3 \rightarrow b_1) \Rightarrow (T, t_1 \rightarrow t_2)$	50.0%	33.3%	75.0%
$(A=a_3) \wedge (B, b_3 \rightarrow b_1) \Rightarrow (T, t_1 \rightarrow t_2)$	33.3%	33.3%	100.0%

4.2 Rule-based action rules mining

Let us assume that objects are described by *I* premises and $L(r) = \{ P_{f1}, P_{f2}, \ldots, P_{fn} \}$. We say that objects *x* supports the action rule *r* in *U*, if there are objects y in *U* and two class association rules *r*1, *r*2 are extracted from *U* such that:

- $P_t(r_1)=k_1$, $P_t(r_2)=k_2$ and $k_1 \leq k_2$
- $(\forall a \in [P_s \cap L(r_1) \cap L(r_2)])\ [a(r_1) = a(r_2)]$
- $(\forall i \leq m)\{\forall e_i \in [P_s \cap [\ L(r_2) - L(r_1)]]\}\ [e_i(r_2) = u_i]$
- $(\forall i \leq n)\{\forall b_i \in [P_f \cap L(r_1) \cap L(r_2)]\}\ [b_i(r_1) = v_i\ ]\ \&\ [b_i\ (r_2) = w_i]$
- $(\forall i \leq o)\{\forall f_i \in [P_f \cap [L(r_2) - L(r_1)]]\}\ [f_i\ (r_2) = c_i]$
- objects x supports rule r1
- objects y supports rule r2

Let $P_s \cap L(r1) \cap L(r2) = A$. By rule-based action rule on $x \in U$ we mean a statement:

$$\{\prod [a = a(r_1): a \in A] \wedge (e_1 = u_1) \wedge (e_2 = u_2) \wedge \ldots \wedge (e_m = u_m) \wedge (b_1, v_1 \rightarrow w_1) \wedge (b_2, v_2 \rightarrow w_2) \wedge \ldots \wedge (b_n, v_n \rightarrow w_n) \wedge (f_1, \rightarrow c_1) \wedge (f_2, \rightarrow c_2) \wedge \ldots \wedge (f_r, \rightarrow c_o)\}(x) \Rightarrow [(P_t, k_1 \rightarrow k_2)](x).$$

where $a = a(r_1)$ denotes that the values of the common stable attributes for both class association rules (r_1 and r_2) are the same, $(e_i = u_i)$ denotes the value of the i^{th} stable attribute in the rule r_2 not in the rule r_1 is equal to u_i, $(b_j, v_j \rightarrow w_j)$ means that the value of the j^{th} flexible premise *b* has been changed from v_j to w_j, and $(f_l, \rightarrow c_l)$ indicates that the value of the l^{th} flexible feature has to be changed from an arbitrary value to *c* for handling correlated flexible features. When attributes are correlated, the change of one attribute value will influence the change of another value.

If we apply the left hand side of the rule on object *x* and this rule basically says: if we change the value of attribute P_{f1} from v_1 to w_1, change the value of attribute P_{f2} from v_2 to w_2, …, and change the value of attribute P_{fn} from v_n to w_n for objects *x*, then, the object which is in class k_1 is supposed to transition to class k_2. Clearly, to have this kind of action rules makes sense. A pair of class association rules extracted from the RDF store should support an action rule. One class association rule basically links the property of $v_1, v_2, \ldots, v_n$ and the property of k_1. Then we have another class association rule that links the property of $\beta_1, \beta_2, \ldots, \beta_p$ and the property of k_2. By definition, an action rule must have this kind of support to be valid.

By the ***support*** of action rule *r*, denoted by *Sup*(*r*), we mean the set of all objects in *U* supporting *r*. In other words, this set of all objects in *U* supporting *r* has the property $(a_1 = u_1) \wedge (a_2 = u_2) \wedge \ldots \wedge (a_q = u_q) \wedge (b_1 = v_1) \wedge (b_2 = v_2) \wedge \ldots \wedge (b_p = v_p) \wedge (d = k_1)$.

By the ***confidence*** of *R* in *S*, denoted by *Conf*(*r*), we mean

$$[Sup(r)/Sup(L(r))] \times [Conf(r_2)]$$

To find the confidence of (r_1, r_2) extended action rule *R* in *S*, we divide the number of objects supporting (r_1, r_2) extended action rule in S by the number of objects supporting left hand side of (r_1, r_2) extended action rule times the confidence of the second classification rules r_2 in *S.*

5. Conclusion and future work

As we are witnessing the increasing trend in both successful application of action rule mining and the spreading of datasets using semantic linked data, we see that applying action rule mining to semantic web data can have tremendous potential for knowledge extraction from different data source. In this paper, rule-based action rules and object-based action rules with their interestingness measurements were defined for semantic web data. We plan to extend our previous work *SAG* (***S**emantic **A**ssociation **G**enerator*) (Tsay, et al., 2015) on building actionable patterns from large-scale datasets.

References

Auer, S., Lehmann, J., Ngom, A.-C. N. & Zaveri, A., 2011. Introduction to Linked Data and Its Lifecycle on the Web. *Reasoning Web,* Volume 485, pp. 1-95.

Berendt, B. et al., 2003. *A Roadmap for Web Mining: From Web to Semantic Web.* Berlin, Springer, pp. 1-22.

Berners-Lee, T., Hendler, J. & Lassila, O., 2001. The Semantic Web. *Scientific American: Feature Article*, May, pp. 1-4.

Jiang, T. & Tan, A.-H., 2006. *Mining RDF Metadata for Generalized Association Rules.* Berlin Heidelberg, Springer, pp. 223-233.

Jimenez, E. S. & Goodman, E. L., 2012. *Triangle Finding: How Graph Theory can Help the Semantic Web.* s.l.:SSWS+HPCSW@ISWC.

Kiefer, C., Bernstein, A. & Locher, A., 2008. *Adding Data Mining Support to SPARQL Via Statistical Relational Learning Methods.* s.l., Springer, Berlin, Heidelberg, pp. 478-492.

Kochut , K. J. & Janik, . M., 2007. *SPARQLeR: Extended Sparql for Semantic.* s.l., Springer, Berlin, Heidelberg, pp. 145-159.

Liu, B., Hsu , W. & Ma, Y., 1998. *Integrating Classification and Association Rule Mining.* New York City, New York, USA, AAAI Press, pp. 80-86.

Lorey, J., Abedjan, Z., Naumann, F. & Böhm, C., 2011. *RDF Ontology (Re-)Engineering through Large-scale Data Mining.* Bonnm Germany, Springer.

Maedche, A. & Staab, S., 2000. *Discovering Conceptual Relations from Text.* Berlin Germany, IOS Pres, pp. 321-325.

Mardini, M. . T. & Raś, Z. W., 2019. Extraction of actionable knowledge to reduce hospital readmissions through patients personalization. *Information Sciences,* June, 485(C), pp. 1-17.

Powell, L., Gelich, A. & Raś, Z. W., 2020. *The Construction of Action Rules to Raise Artwork Prices.* Cham, Springer, pp. 11-20.

Ramezani, R., Saraee, M. H. & Ali, M., 2014. *Ramezani, R., Saraee, M.H., & Nematbakhsh, M.A. (2014). SWApriori : a new approach to mining association rules from semantic web data..* s.l., s.n.

Raś, B. W. & Tsay , L.-S., 2003. Discovering Extended Action-Rules (System DEAR). *Intelligent Information Processing and Web Mining. Advances in Soft Computing,* Volume 22, pp. 293-300.

Raś, Z. W. & Wieczorkowska , A., 2000. *Action-Rules: How to Increase Profit of a Company.* s.l., Springer, Berlin, Heidelberg, p. 587–592.

Savasere, A., Omiecinski, E. & Navathe, S. B., 1995. *An Efficient Algorithm for Mining Association Rules in Large Databases.* Zurich, Swizerland, Morgan Kaufmann Publishers Inc., p. 432–444.

Srikant, R. & Agrawal, R., 1995. *Mining Generalized Association Rules.* Zurich, Switzerland, Morgan Kaufmann Publishers Inc., pp. 407-419.

Tarnowska, K. & Raś, Z. W., 2018. *"From Knowledge Discovery to Customer Attrition.* s.l., Springer, Cham., p. 417–425.

Tsay, L.-S., Sukumar, S. . R. & Rob, L. W., 2015. *Scalable association rule mining with predication on semantic representations of data.* Tainan, Taiwan, IEEE, pp. 180-186.

Völker , J. & Niepert , M., 2011. *Statistical Schema Induction.* Heraklion, Greece, Springer, Berlin, Heidelberg, p. 124–138.