Model of Finding Associative Rules in Inhomogeneous Data of Semantic Networks

Nataliya Boyko, Vladyslav Mykhailyshyn

Lviv Polytechnic National University, Profesorska Street 1, Lviv, 79013, Ukraine

Abstract

The paper examines the issues of hidden connections and potentially useful information from large data sets. Theoretical knowledge about associative rules is substantiated, their influence on connections in multimodal data sets is investigated. The methods of application of associative rules in practice are analyzed. The following are considered in detail: the basic concepts of associative rules and their connection with the idea of logical regularity; ways to determine the "strength" of these connections; basic algorithms for finding patterns; practical implementation of the search for associative rules. The regularities in the "templates" are analyzed: support and confidence value. The correct choice of these values, which directly affect the results of the search for rules, is experimentally determined. Research in this paper aimed to consider the basic concepts and find the Associative Rules both in traditional ways and in heterogeneous data of semantic networks, which creates specific problems when using existing algorithms. The data of semantic networks are analyzed, which in most cases serve a particular field and are highly specialized. The research presents the process of finding associative rules through the work of the classical Apriori algorithm and an alternative algorithm for finding associative rules. Previously, this problem was considered only to a small extent. The results of experiments on accurate SW data showed promising results.

Keywords [1]

Artificial Intelligence Systems, Big Data Mining, Associative Rule, Database, Semantic Web, Data Mining, Health-e-Child.

1. Introduction

The primary purpose of data mining is to "reveal" the hidden connections and potentially useful information from large datasets [3, 7, 8]. Associative rules are one of the ways that help to identify these connections. Currently, there are many areas where the search for associative rules is used, as in IT (search for associations between data in the list of databases transactions, analysis of weblogs) and in the consumer sphere (the problem of the product basket, product placement, demand forecasting), areas of marketing (search for market segments, trends, identification of firms clients groups). This work will be discussed in detail:

- The basic concepts of associative rules and their connection with logical regularity.
- Ways to determine the "strength" of these connections.
- Basic algorithms for finding frequency.
- Practical implementation of searching for associative rules.

For the first time, searching for associative rules arose in the consumer area: it was necessary to identify specific "patterns" of consumer purchases to increase sales of goods due to these data. The Associative rule acquired of the form: "Event X is followed by event Y", as a result of which it is possible to obtain a certain regularity - if the purchase (transaction) has a set of goods (elements) X,

[1]The Fifth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2022), May 12, 2022, Zaporizhzhia, Ukraine
EMAIL: nataliya.i.boyko@lpnu.ua (N. Boyko); vladyslavmykhailyshyn@gmail.com (V. Mykhailyshyn)
ORCID: 0000-0002-6962-9363 (N. Boyko); 0000-0003-1889-9053 (V. Mykhailyshyn)

then with some probability to assume that the set Y will also appear in it [4]. These rules are characterized by support and confidence values. The correct choice of these values directly affects the search results of the rules. Yes, if the support value is too large, the algorithm's results will be already known and quite noticeable. On the other hand, too small the support value will help identify many different patterns, but there can be some doubts about their reliability. The same with confidence value - the design will be less "valuable" at too low values [1, 6].

The department grouped all products in grocery stores to find what they needed more quickly. It reduces spending time shopping in a store and is also interested in buying something else. Keep in mind that associative rules will not help, in this case, the consumer's personal preferences. Still, with their help, it is possible to find connections between items in each purchase transaction (as opposed to filtering preferences, which considers all purchases of one consumer to recommend to him goods or services in the future). Therefore, the data to search for associative rules are regarded as separate purchases of different consumers by one group.

2. General theoretical information and the concept of associative rule

The purpose of AR [5, 10] is to find all the relationships (also called associations) between datasets, elements of certain regularity between date [13]. The basic concept of AR can be represented as follows: let $S = \{s_1, s_2, \dots, s_m\}$ the list of things, then $T = \{t_1, t_2, \dots, t_n\}$the list of transactions (purchases), where each transaction is a set of things from the list S(that is $t_i \subseteq S$). Exactly AR is presented in the form $X \to Y$, where $X \subset S$,$Y \subset S$ and $X \cap Y = \emptyset$(X and Y - set of things) [13]. Let our shopping database DB look like this way (Table 1):

Table 1

Example of rule: $\{B, C\} \to \{A\}$

ID	Items
0001	A, F, B, C
0002	A, C
0003	A, D
0004	D, E, C
0005	F, A, D

Let's say we want to analyze how sales are related to certain goods in the store. In this case, the S list will include all goods in the store, and the transaction (purchase) will be a set of things in the buyer's basket. Let's find AR: $\{A, F\} \to \{C\}$(where$\{A, F\} = X$, $\{C\} = Y$): transaction $t_i \in T$ must contain a list of things X (that is, it is a subset t_i, "covers" the transaction).

2.1. Concept of Support value

In the case of a product basket problem, consider the following example of a rule: Bread → Apples (support value = 20%, confidence value = 45%). The results show that 20% of buyers buy bread along with apples at the time like 45% of buyers who buy bread, they also buy apples. Support value and confidence value determine "power" of this rule [2, 5, 9].

The calculation of the support value rule is to represent transactions $t_i \in T$, which are subordinate $X \cup Y$, and in some way is the probability $P(X \cup Y)$, in other words, it is the amount or percentage of transactions that have a set of specific elements. The support value of the $X \to Y$ rule will be calculated by the Formula 1:

$$support\ value = \frac{support\ count(X \cup Y)}{n}, \quad (1)$$

where n - the number of transactions in the T list (in our database); $support\ count(X \cup Y)$- the number of transactions in T that include X and Y [2, 9].

Support value is useful in cases where it's too small value indicates that the rule can happen "accidentally". Support value list of things $\{A, F, C\} = 1/5 = 20\%$ in our DB (Table 1).

2.2. Concept of Confidence value

Confidence value rule consist in representing transactions $t_i \in T$with values from the list Y, which include X, and in some way is a conditional probability $P(Y|X)$. In other words, confidence value determines how often "things" in list Y appear in transactions with things in list X. Confidence value rule X → Y will be calculated by the Formula 2 [12, 15, 7]:

$$confidence\ value = \frac{support\ count(X \cup Y)}{support\ count(X)}, \quad (2)$$

where $support\ count(x)$- the number of transactions in T that include X; $support\ count(X \cup Y)$- the number of transactions in T that include X and Y [15, 7].

The confidence value determines the "predictability" of a rule. When its value is too small, there is a problem of reliability of definition or prediction of Y and X. Confidence value rule $\{A, F\} \rightarrow \{C\} = 1/2 = 50\%$ in our DB (Table 1).

2.3. Concept of Lift value

There is a problem: what to do when confidence value, for example, of rule $\{A, F\} \rightarrow \{C\}$, is less than $P(\{C\})$? Lift value acts as an indicator of the "predictive power" of the rule compared to a random event, calculated (Formula 3) [8, 17, 21]:

$$Lift\ value(X \rightarrow Y) = \frac{P(Y|X)}{P(Y)}, \quad (3)$$

where (if lift value > 1, then Y will occur more likely at a given X; lift value<1 - Y will occur less likely at a given X).

Can be represented as $\frac{confidence\ value(X \rightarrow Y)}{support\ count(Y)}$ (if X and Y are independent, the value of lift value will be equal to 1; if X and Y occur more often than if they were independent, lift value> 1) [11, 15, 19].

3. Materials and methods

Studies of semantic annotations show that the increase in semantic networks (SW) data is constantly growing. Problems with RDF / (S) and OWL data extraction occur with graphics-oriented software. The solution to this problem may be to use new algorithms that use the axioms of ontologies (Tbox) to obtain the corresponding transactions, which will later use traditional association rule algorithms to find the result. With the help of the analyst, you can control the process represented by query templates. [4, 12, 20].

Today dictates new languages to create alternative associative rules. It combines semantic networks and data mining that allows you to carry out the process faster and without errors. Recently, a new direction has been widely developed, which researchers call Ontology Learning [2, 18, 21].

If you consider a database of unstructured and different data types, you need to use data extraction in semantic networks as the most appropriate for this data set. Its application allows the processing of data sets with different semantics and structure. In this study, we will consider the possibility of combining ontological instances expressed by OWL into search mechanisms by traditional algorithm search algorithms. The task is to reduce the space for searching for the data you are looking for.

Large data sets need to be used to find patterns in traditional networks. Semantic data requires a different approach than the one offered by machine learning [9, 12]. Therefore, in the process of finding patterns, there are several problems, including:

- In the process of processing the algorithm, a priori for homogeneous data should apply transactions. Accordingly, each transaction is a subset of the elements. If we use semantic networks, we will see that ontological axioms describe the subject area. Semantic annotations, in this case, are represented by statements that describe each instance through its property. In semantic networks, as an example, you can find a triplet - this is when the information is represented through the subject, predicate and object. When viewing such a scenario, the very definition of transactions and elements is not trivial. In semantic networks, elements can

correspond to both instances and literals. accordingly, a transaction can be defined by a subset of elements that are semantically related in the data warehouse [10, 11].

- The ontology of the semantic network represents data through formal properties and particular semantics. In this case, the network does not have a clear structure; respectively, the instances belonging to it may belong to a specific class and have a different form [19, 20].

Previous work on SW data extraction has focused mainly on clustering and instance classification. However, the presentation methods required for associative data mining are different from the usual clustering and classification tasks. In the general concept of transactions, the rules are specific observations of the frequency of occurrence of a particular set of elements. For example, in vector-numerical form, the presented data sets are the same as in a traditional format. They can be used to cluster and classify data. For our study, it is essential to process semi-structured and heterogeneous data. Therefore, an ontology should be used to identify data quickly. It is also necessary to consider the ways of creating elements and transactions in semantic networks, as they depend on the level of detail and the very structure of semantic data.

3.1. Definition of SW data

Analyzing the above, it can be argued that to incorporate semantics into current web content, and you need to use appropriate technologies. They can be used to provide presentation elements for the language. For example, you can consider specific presentation formats based on XML. Therefore, to describe semantic metadata, we use the resource description language (RDF), which contains three types of elements. The first is the resource, all web objects that a URI can identify. The second is literals, i.e. numbers, atomic values, dates, strings, etc. Third, properties are binary relationships between resources and literals that a URI can identify. The main components of the RDF are triplets: a binary relationship between two resources or between a resource and a literal. The resulting metadata can be considered as a graph, where nodes are resources and literals, and edges are the properties that connect them. RDFS extends RDF to allow you to define triplets by classes and properties. Thus, we can describe a schema that manages our metadata in the same description frame. An ontology web language (OWL) was later proposed to facilitate the work on the semantic description [3, 14, 17].

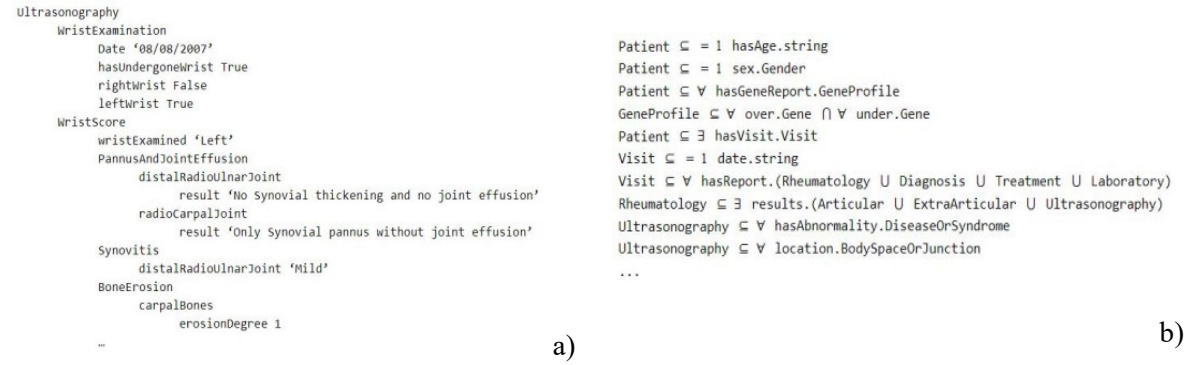

Figure 1: a) Excerpt from a patient's clinical report in the field of rheumatology; b) Axioms of ontologies (Tbox)

One of the areas of application may be medicine - here all the time a huge amount of semantic data is generated. In particular, most semi-structured and very heterogeneous data sources (e.g., laboratory test reports, ultrasound scans, images) are subjected to semantic annotation using UMLS, NCI and Galen ontologies. Suppose we have an excerpt from a clinical report presented in Figure 1a, the semantic annotation of which leads to certain axioms (Figure 1 b) and statements (Figure 2). Axioms in Figure 3 provide the semantics of all information concerning the patient (i.e., medical history, reports, laboratory results) by conceptualizing the domain. Figure 4 provides data on triplets that describe the patient (i.e., semantic annotations (Abox); subjects, predicates, and URI-formatted objects that point to relevant data resources). The generated data also represent complex relationships that are rapidly

evolving with the use of new biomedical research methods. Obviously, traditional analytical tools are not suitable for this type of data [1, 8, 12].

Axioms of ontologies (Tbox) allow to define an area from the point of view of atomic concepts (classes in OWL) and roles (properties in OWL). OWL provides for the union ⋃, intersection ⋂ and negation ¬, as well as list classes (one Of), existence ∃, universality ∀ and constraints (≤, ≥, =) of the atomic concept of R or the inverse $\neg R$.

Subject	Predicate	Object
PTNXZ1	hasAge	'10'
PTNXZ1	sex	'Male'
VISIT1	date	'06/18/2008'
VISIT1	hasReport	RHEX1
RHEX1	damageIndex	'10'
RHEX1	results	ULTRA1
ULTRA1	hasAbnormality	'Malformation'
ULTRA1	hasAbnormality	'Knee'
VISIT1	hasReport	DIAG1
...	...	...

Figure 2: Three patient descriptions

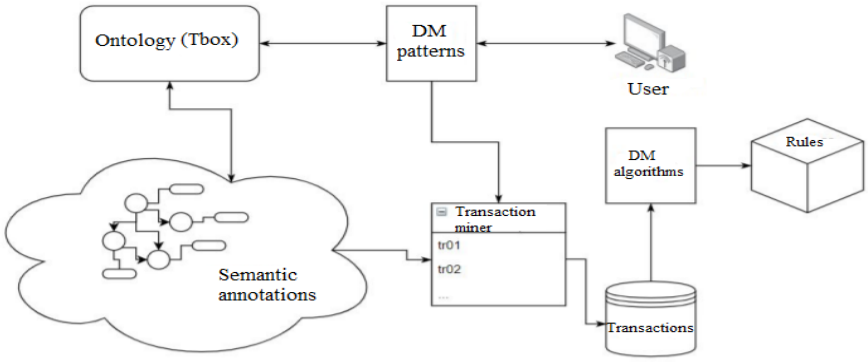

Figure 3: Scheme of the approach of extracting associative rules of semantic annotations

This section presents an overview of the method according to the scheme shown in Figure 3. The user specifies the extraction pattern using the query language syntax. The transaction miner can identify and construct transactions according to a previously defined mining scheme. Finally, the set of received transactions is processed by the traditional pattern mining algorithm, which finds the associative rules according to the minimum values of support value and confidence value defined in the template for the network shown in Figure 4.

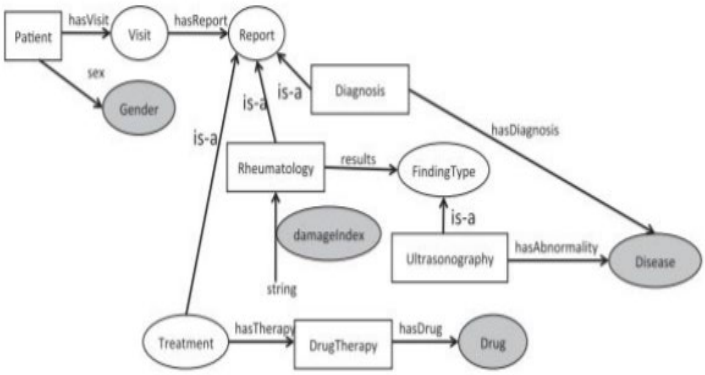

Figure 4: A fragment of the semantic network in the field of rheumatology

Both the ontology and the instances can be represented in the form of subject-predicate-object triplets (Figures 5-6), forming a graph where nodes are resources and literals, and edges are properties that connect. This dynamic graph-based structure contrasts with the well-structured and homogeneous datasets used in conventional associative rule search algorithms. Therefore, to obtain entities and transactions, users must specify the target concept of the analysis and related functions. The features must be relevant to the target concept, i.e., they will be extracted from the subgraph of each instance belonging to the target concept of the analysis.

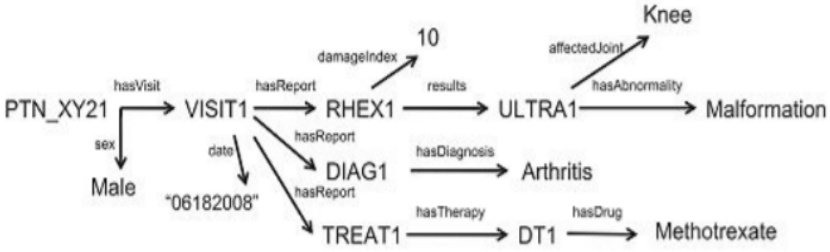

Figure 5: Fragment of the semantic annotations graph

Subject	Predicate	Object	Feature
PTN_XY21	(VISIT1, RHEX1, ULTRA1)	Malformation	Disease
PTN_XY21	(VISIT1, RHEX1)	RHEX1	Report ∩ ∃ damageIndex
PTN_XY21	(VISIT1, TREAT1, DT1)	Methotrexate	Drug
...	...	...	...

Figure 6: Subjects-subject-object triples

In this method, only important instances (i.e., features) are "extracted" and combined from the entire repository and embedded in regular transactions, capturing implicit data at the schema level in the ontology. You can then apply existing associative rules search algorithms. This type of search rule will become increasingly valuable in future research on both machine learning and SW data. In the future, you can apply generalized query schemes using ontology axioms, as well as automatically detect important instances and search for associative rules. In addition, the method can be used in a variety of

scenarios where mining tasks are transaction-oriented. The problem of the research is the use of data in the ontology to filter and narrow the identified rules, as well as to express the goals of the user. Another important area worth researching is the combination of clustering mining algorithms and associative rules. Previously, this technique has been implemented through hierarchical clustering based on a set of subjects (FIHC). Basically, the FIHC algorithm generates clusters from frequent sets of elements, which in turn constitute cluster descriptors. A new approach could be an algorithm based on finding frequent pairs of objects, which provides more homogeneous clusters and better descriptions than those obtained from FIHC. Also, many studies involve the use of more complex algorithms for data extraction and the formation of transactions from them, the study of their efficiency. No less interesting is the development of new algorithms for data exchange, which are based on semantically enriched elements of the generated transactions.

4. Algorithms for searching Associative Rules

This section will take a closer look at both the well-known AR search algorithms and the new AR search algorithm in semantic networks.

4.1. Traditional search algorithms of AR

AIS algorithm. The first algorithm for finding associative rules, called AIS, was developed by IBM Almaden Agrawal, Imielinski, and Swami in 1993. From this work began an interest in associative rules; in the mid-90s of the last century came the peak of research in this area, and since then every year there are several new algorithms. In the AIS algorithm, candidates for multiple sets are generated and counted "on the fly" while scanning the database [1, 17, 21].

SETM algorithm. The creation of this algorithm was motivated by the desire to use the SQL language to calculate frequent sets of goods. Like the AIS algorithm, SETM also generates candidates "on the fly" based on database transformations. To use the standard SQL join operation to form a candidate, SETM separates the candidate formation from their count.

The inconvenience of AIS and SETM algorithms is the excessive generation and calculation of the Support value of too many candidates, which as a result are not provided often. To improve their performance, the Apriori algorithm was proposed. [7, 13]

Apriori algorithm. The work of this algorithm consists of several stages - the formation of candidates and the counting of candidates. Candidate generation is the stage at which the algorithm, by scanning the database, creates many i-th candidates. At this stage, their Support value is not calculated. Candidate counting is the stage at which the Support value of each i-th candidate is calculated. Candidates whose Support value is less than the minimum value set by the user (min Support value) are also rejected here. The other i-th sets will be the ones that are often found in the database - that is, if the set {A, B} is common, then the sets {A}, {B} will also be common. This property is the Support value property (Formula 4): [2, 5, 8]

$$\forall X, Y: (X \subseteq Y) \Rightarrow Support\ value(X) \geq Support\ value(Y), \quad (4)$$

where *X, Y* - sets of elements.

Looking at the algorithm of simple search of values, in it there are 2^n variants of sets at the given n elements (Figure 7). Suppose we have a set (AB) with a low value Support value - the Apriori algorithm "cuts off" AB and its derivative sets, thereby accelerating (Figure 8).

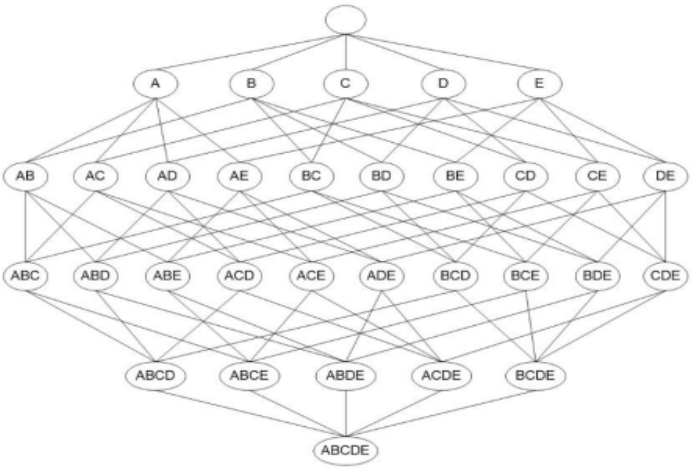

Figure 7: Number of sets (2^n) for these elements

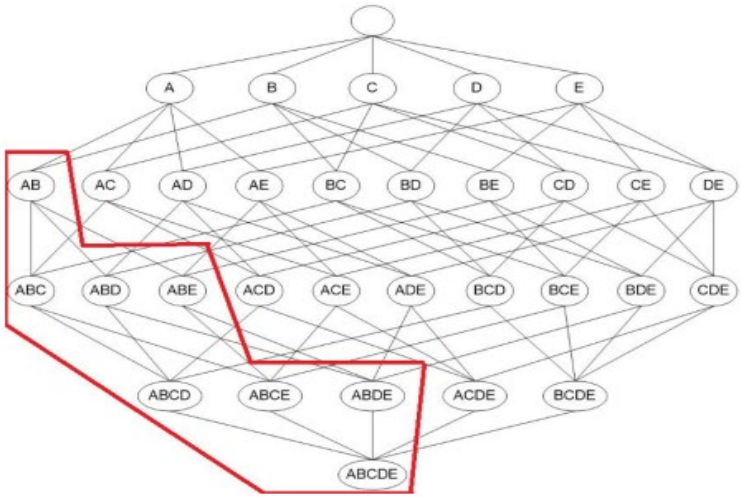

Figure 8: Clipping sets with low Support value

Let's consider the Apriori algorithm on an example, for this purpose we will change a little and we will expand our Table 1 with data (Table 2):

Table 2

Additions to table 1 by associative rules

ID	Items
001	A, B, C
002	B, C
003	B, A, D, C
004	E, D
005	A, B, C, D
006	F

Set min Support value = 3 (Figure 9).

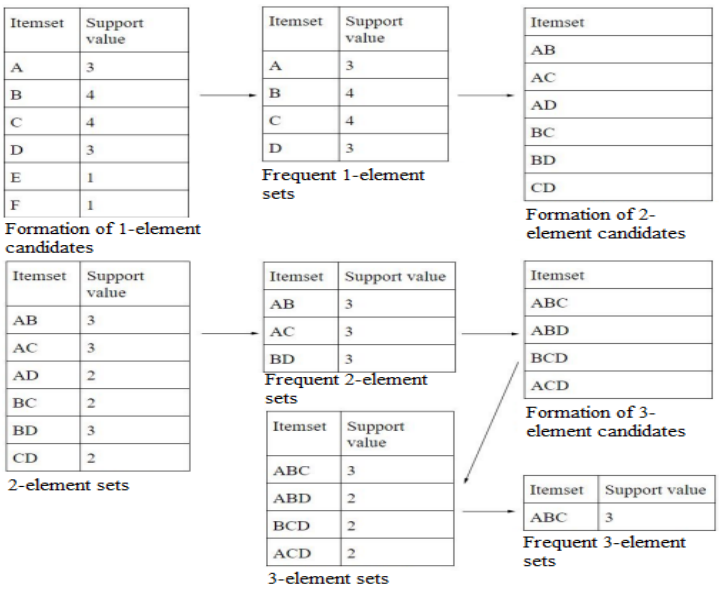

Figure 9: Apriori algorithm

At the first stage (Figure 9), there is a formation of 1-element candidates. Next, the algorithm calculates the Support value of 1-element sets. Sets with a Support value less than the specified (in our case 3) are cut off. In the example, these are sets E and F, which have *Support value* = 1. The remaining sets of elements are considered to be common: A, B, C, D.

Next is the formation of 2-element candidates, counting their Support value and cutting off sets from *Support value* < 3. The remaining 2-element sets AB, AC, BD, participate in the further work of the algorithm.

Continuing the work, the algorithm at the last stage forms 3-element sets of goods: ABC, ABD, BCD, ACD, calculates their Support value and again cuts off sets from *Support value* < 3. The result - a set of ABC products is the most common (Figure 9).

Among the varieties of the Apriori algorithm are the following:

- AprioriTID. The peculiarity of this algorithm is that the database of elements is not used to calculate the Support value of the recruitment candidates after the first step. For this purpose, the candidate coding performed in the previous steps is used. In the following steps, the size of the encoded sets can be much smaller than the database itself, thus saving significant resources.
- AprioriHybrid. Analysis of the running time of the Apriori and AprioriTID algorithms shows that in earlier steps Apriori achieves better speed than AprioriTID; however, AprioriTID works better than Apriori at later steps. In addition, they form the same procedure for candidate sets. Based on this observation, the AprioriHybrid algorithm is proposed to combine the best properties of the Apriori and AprioriTID algorithms. AprioriHybrid uses the Apriori algorithm in the initial steps and moves to the AprioriTID algorithm when large sets of memory can be used. However, switching from Apriori to AprioriTID requires resources.

Some authors have proposed other algorithms for finding associative rules, which were also improvements to the Apriori algorithm. One of them is the DHP algorithm, also called the hashing algorithm (proposed by J. Park, M. Chen and P. Yu, 1995). Based on its probabilistic calculation of sets of candidates, valid for reducing the count of candidates for the duration of the Apriori algorithm. The reduction is provided by the fact that of the k-element sets of candidates, in addition to the step of reducing the passage of the hashing step. In the algorithm at the k-1 stage during the selection of the candidate, the so-called hash table is created. Each hash table entry is a counter of all reference values of k-element sets that correspond to a row in the hash table. The algorithm uses this information on the k-th to reduce the elements of the candidate sets. After reducing the subset, as in Apriori, the algorithm can delete the candidate set if its value in the hash table is less than the specified Support value.

Also to other advanced algorithms: PARTITION, DIC, the algorithm of "sample analysis". PARTITION algorithm (proposed by A. Savasere, E. Omiecinski and S. Navathe, 1995). This algorithm

of partitioning (division) is contained in the database of scanning operations through the section of its section, each of which can fit in RAM. In the first place in each of the sections using the Apriori algorithm displays sets that are common. The second confirms the importance of supporting each such set. Thus, the stages are available on all other common data sets. To compare the operation of the algorithms, it would also be essential to analyze the DIC algorithm (Dynamic Itemset Counting), which was proposed by scientists S. Brin R. Motwani, J. Ullman and S. Tsur in 1997, divides the database into several blocks, each of which is marked by so-called "start points", and then cyclically scans the database.

4.2. Search algorithm of AR in semantic networks and data

Until now, AR search methods have been applied to traditional data in tabular format or on the basis of graphs. This section explores the problem of finding rules in semantic web data and proposes a new approach to finding APs directly from semantic web data. This approach takes into account the complex nature of semantic web data in contrast to traditional data and, in contrast to existing methods, eliminates the need to convert data and involve end-users in the search process. In trying to apply this search to atypical data, we encounter certain problems and differences compared to traditional ones:

- Heterogeneous: traditional mining algorithms work with homogeneous data sets in which instances are stored in a well-ordered system, and each instance has predefined attributes. But the semantic data are heterogeneous. This means that specific instances of categories / domains (e.g. people, cars, medicines, etc.) based on the same ontology or individual ontologies may have different characteristics.
- There is no clear definition of transactions: in conventional information systems, data is stored in databases using predefined structures, and these structures can be used to recognize transactions and thus extract them from the data set. Then the traditional AR search algorithms process these transactions. For example, in the case of a “buyer basket”, transactions are formed from products that are purchased together, and these products will have the same ID as the transaction ID. Conversely, in a semantic network, different attributes for an instance may be formed at different times, and therefore an instance may have an attribute that does not exist in another instance of the same type.
- Multiple relationships between entities: Traditional AR search algorithms to generate large sets of elements take into account only the values of the objects and assume that there is only one type of relationship between the entities (for example, purchased together). But in semantic data, there are many relationships between entities. In fact, predicates are relations between two entities or between one entity and one value.

Because semantic annotations are encoded in RDF / (S) and OWL, you should extend SPARQL with new elements that can specify a search pattern. The syntax is somewhat similar to Microsoft Data Mining Extension (DMX), which is an SQL extension for working with DM models in Microsoft SQL Server. The extended SPARQL grammar is shown in Figure 10 and Figure 11 shows an example of the SPARQL view for AR search schemes (Formula 5):

$$Q = (Patient, Report, \{Disease, Drug, Report \cap \exists damageUndex\}). \quad (5)$$

```
Query ::= Prologue(SelectQuery|ConstructQuery|DescribeQuery|AskQuery|MiningQuery)
MiningQuery ::= CREATE MINING MODEL Source `{`
                    Var `RESOURCE` `TARGET`
                        (Var(`RESOURCE`|`LITERAL`)
                            `MAXCARD1`? `PREDICT`? `CONTEXT`?)+`}`
                        DatasetClause* WhereClause UsingClause MeasuresClause
UsingClause ::= `USING`SourceSelector BracketedExpression
MeasuresClause ::= `ADD MEASURE` measure +
```

Figure 10: Advanced SPARQL grammar for the CREATE MINING MODEL query

The SPARQL query has been expanded by adding a new character called MiningQuery. The body of the query consists of variables that the user targets when searching for data. Next to each variable, we define its type: RESOURCE for variables that contain RDF, and LITERAL for those that have regular data types. In case we want to find patterns with only one variable, we add the keyword MAX-CARD1 to the variable. By default, found templates can contain more than one occurrence of each variable. In addition, we define the "sequence" of this rule by adding the PREDICT keyword (optional). Finally, the TARGET keyword refers to the analyzed resource, which should be an ontology concept. The purpose of the analysis determines the set of rules obtained. In WhereClause, we specify restrictions on previous variables. The advantage is that the user's knowledge of the ontology structure is not required. Therefore, users only need to specify the type (concept of ontology) to which the variables refer.

```
CREATE MINING MODEL {
        ?patient  RESOURCE TARGET
        ?drug  RESOURCE
        ?jadi  LITERAL
        ?disease  RESOURCE PREDICT
        ?report  RESOURCE CONTEXT
}
WHERE {
        ?patient  rdf:type  Patient .
        ?drug  rdf:type  Drug .
        ?disease  rdf:type  Disease .
        ?report  rdf:type  Report .
        ?report  damageIndex ?jadi .
}
USING Apriori (SUPPORT = 0.06, CONFIDENCE = 0.7)
ADD MEASURE lift, leverage
```

Figure 11: Extended view of the SPARQL CREATE MINING MODEL query

Let the user choose the patient as the desired "concept" of the analysis. The set of characteristics that will make up the transaction includes diagnosed diseases, prescribed drugs and damage rate. Finally, the transaction will be based on the details of the report, i.e. the transactions will not include the characteristics in all reports in general, but only the characteristics of each doctor's report.

The variable jadi refers to the index of injury to the patient's joint, the user specifies the report and damageIndex as a property of the resource and the type of data from which they can be obtained. UsingClause defines the name and parameters of the algorithm.

Because we do not ask the user to specify the exact relationship, the query model introduces some ambiguity about the elements that perform the transaction. Thus, the same conceptual changes (selected features) can be used under different contexts of ontology. For example, Disease can diagnose the patient's own illness or the illness of a family member. This ambiguity becomes a problem in determining what the intentions of the users really are. In fact, the user can use this ambiguity by specifying in the extended SPARQL query to understand the ontology using the "triplets" WHERE. However, this task can be cumbersome. For the query to be really correct, the user can select the desired context using CONTEXT added to the corresponding concept. In addition, the system will build transactions, taking into account all possible contexts.

Recalling the form of subject-predicate-object triplets (Fig. 2.2.4), they will be useful for the above-mentioned AR search scheme $Q = (Patient, Report, \{Disease, Drug, Report \cap \exists damageUndex\})$. Instances of RHEX1, RHEX1, TREAT1 will belong to the concepts of context Q. The transactions of the elements obtained from these three compositions are shown in Figure 12:

Transactions
1. {*RheuExam.Disease.Malformation*, *RheuExam.RheuExam.damageIndex* → 10}
2. {*Treatment.Drug.Methotrexate*}
3. {*RheuExam.Disease.Malformation*, *RheuExam.Disease.BadRotation*, *RheuExam.RheuExam.damageIndex* → 15}
4. {*Treatment.Drug.Methotrexate*, *Treatment.Drug.Corticosteroids*}

Figure 12: Transactions of elements of triples of compositions

The algorithm itself will follow the following steps:

- First, compute sets of common elements that reduce their total number (especially when there are a large number of transactions).
- Then the sets of elements are truncated by the method described in subsection 4.1 (Figure 9) with *Support value* < 0.7 to filter out those that combine frequent and rare elements. These transactions are usually false.
- Finally, you can get rules from element sets by specifying min *Confidence value* = 0.8.

5. Results of research and experiments

To ensure that the algorithm is correct and relevant, it will be tested on real-world OWL instances of patient observation. According to the example in Fig. 1b, annotations were formed based on the Health-e-Child (HeC) project. These annotations correspond to the ontology of the project. The structure of semantic annotations is very heterogeneous and contains information about 588 patients classified into three different groups according to their disease: juvenile idiopathic arthritis (JIAPatient), heart disease (CardioPatient) and neurological disease (NeuroPatient). The total number of semantic annotations is 629,000, which is an average of more than 1,000 annotations per patient.

To avoid errors, query schemes were automatically generated for 12 different concept concepts (disease, treatment, medication, ...) and 3 concepts for contexts: patient, visit and report. It is worth noting that the Report concept has 20 sub-concepts that correspond to the various clinical reports of the HeC project. The current implementation of transaction extraction has been developed on the basis of the ontology indexing system, which also provides a simple mechanism for creating ontological indexes. To confirm the relevance and results of the found transactions, there is a range of different AP search algorithms, among which genetic algorithms (GA) for AP search have recently been proposed.

On the Table 3 shows three selected contexts for experiments, as well as the number of generated transactions and their average length.

Table 3

Results of experiments on the number of generated transactions and their average length

Context	Transactions	Medium length
Patient	588	29.57
Visit	1458	12.84
Report	3608	5.24

The number and nature of transactions received in each context are completely different and will therefore affect the rules created. More general contexts tend to generate longer transactions, which in turn increases the likelihood of obtaining more rules. Instead, more specific contexts generate smaller transactions, which narrows the scope for detecting rules. This discrepancy in the nature of transactions necessitates adequate adjustment of the minimum Support value threshold of each set-in order to be able to find the association rules.

In the Table 4 shows the number of created rules together with their average Confidence value, average Lift value and ϕ-coefficient for three sets of transactions.

Table 4

Results of experiments on the number of generated transactions and their average length

Context	Min Support value	Amount of rules	AVG Confidence value	AVG Lift value	ϕ-coefficient	Correlation
Patient (588)	0.187	109	0.993	2.944	0.796	0.678

Visit (1458)	0.047	93	0.976	9.975	0.865	0.480
Report (3608)	0.017	151	0.964	27.69	0.836	0.169

All created rules have a high Confidence value. In addition, the more limited the context, the better the rules are formed. Moreover, the ϕ-coefficient shows a strong correlation in all cases.

In the Table 5 shows the effect of applying certain restrictions (i.e. selecting only specific report types) in the search template.

Table 5

The result of the impact of the application of the imposed restrictions in the search template

Transactions	Amount of rules	AVG Confidence value	AVG Lift value	% of report rules
All reports (588)	655	0.943	4.125	83
Rejected 5 (585)	22	0.963	6.966	56
Rejected 12 (438)	26	0.934	6.652	35

Each line displays received transactions and rules for all reports, canceling the 5 most common reports and discarding the 12 most common reports in the patient context. This table also includes the percentage of rules that contain items from different reports.

Figure 13 analyzes the coverage of the formed rules by different thresholds of support as an indicator of their quality.

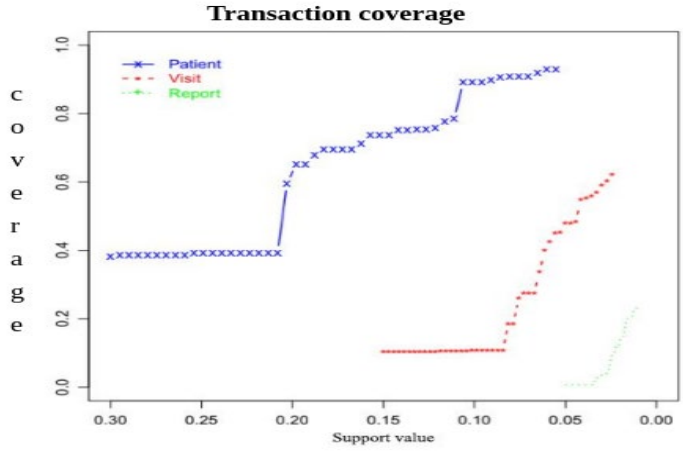

Figure 13: Coverage of transactions achieved by generated rules with different minimum threshold levels Support value

The rules obtained from the Patient set achieve good coverage with relatively high thresholds of Support value. However, other received sets of rules are not able to confirm the high percentage of transactions. The fact is that the length of other sets of transactions is shorter, which usually reduces the number of detected AR. In the case of the Report transaction set, the coverage is even less because the transactions are derived from different types of reports. Therefore, good rules may arise, but with very low Support value thresholds. In these cases, it would be advisable to use more sophisticated AR search algorithms that are not based on the concept of Support value.

In Figure 14 shows the average Confidence value of the generated rules with different threshold Support value.

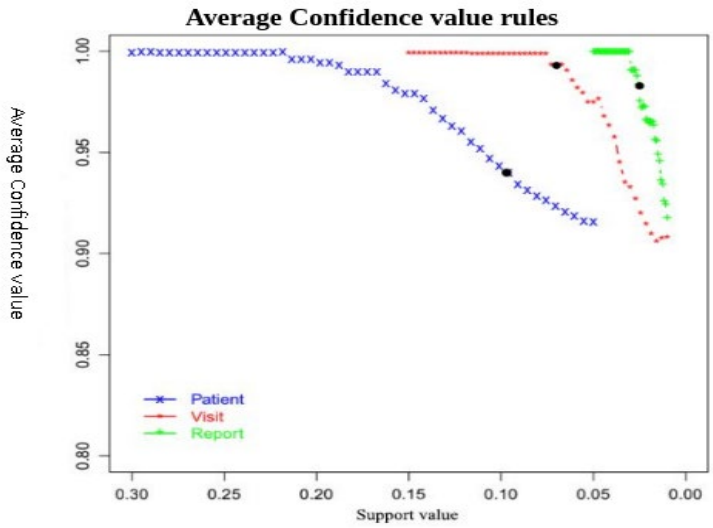

Figure 14: Average Confidence value of generated rules with different thresholds Support value

The support value for the three sets of transactions remains high even for low thresholds, which confirms the quality of the rules.

Based on the two previous measures (coverage and average Confidence value) we can select a minimum Support value threshold for each set of transactions and further analyze the quality of the rules obtained. In Figure 15 shows the coating, multiplied by the average Confidence value. For each set of transactions, the Support value threshold is selected, at which both measures are maximally involved.

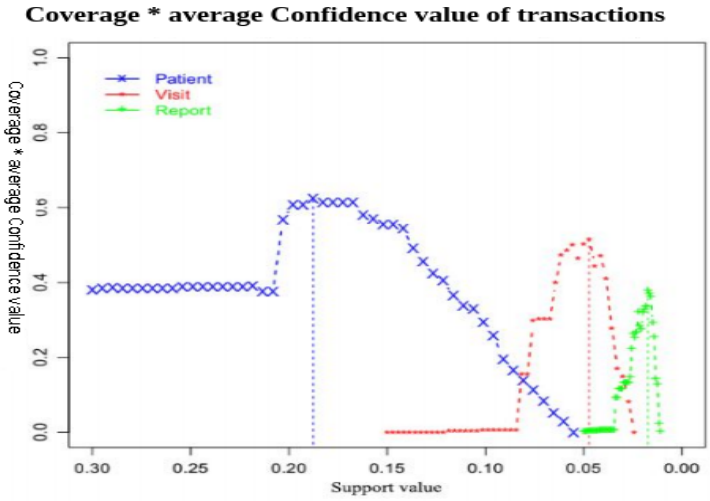

Figure 15: Coverage multiplied by the average Confidence value of the generated rules with different Support value thresholds

Finally, the three Figures 16, 17 and 18 show an example of AR obtained in the context of Patient, Visit and Report.

Rule	Support value	Confidence value	Lift value
{JIAPatient. (Disease.disease)→ oligoarthritis, JIAPatient. (Finding.sacroiliac_ tenderness & Anatomy.sacroiliac_ joint & MedicalProcedure.p resence)} ⇒ {JIAPatient. (Symptom.inflammat ory_pain)}}	0.26	1.0	18.833
{CardioPatient. (MedicalProcedure. genetic_research & ExternalActivities.ge netic_molecular), CardioPatient. (Molecular.tbx5)} ⇒ {CardioPatient. (Molecular.5_nkx2)}	0.107	1.0	9.187
{NeuroPatient. (ExternalActivities.e ndocrinology)} ⇒ {NeuroPatient. (MedicalProcedure.r esonance_magnetic_ material_imaging_c ontrast &Chemical.metal)}	0.107	0.969	8.380
{JIAPatient. (Symptom.tendernes s & Quantity.score)→1} ⇒ {JIAPatient. (Finding.pain)→1}	*0.144*	*1.0*	*6.917*

Figure 16: The result of an associative rule obtained in the context of the Patient

When considering the results, the Support value can also be interpreted as a percentage, for example for the first rule it was 0.260 - this means that sets of diseases in the rule occur in 26% of transactions. In many ARs found in the table above, the Confidence value is close to 1. Considering the first rule in the example, this means that in 100% of transactions a patient with oligoarthritic and lumbar pain has active tissue inflammation in this department. Lift value characterizes how good the prediction is and how interdependent the factors are. The greater the value, the greater the dependence of factors, i.e. how much the presence of one factor affects another. With low Lift values, on the contrary, the lower it is, the greater the negative effect one factor has on another.

Rule	Support value	Confidence value	Lift value
{CardioVisit. (MedicalProcedure. auscultation_lung), CardioVisit. (Quantity.compensa tion)} ⇒ {CardioVisit. (Finding.breath_so unds & Quality.equality)}	0.074	1.0	11.950
{JIAVisit. (Chemical.methotre xate)→Weekly} ⇒ {JIAVisit. (Chemical.methotre xate)}	0.08	0.990	7.197

Figure 17: The result of an associative rule obtained in the context of Visits

In this case, for example, the first rule shows good results in Confidence and Lift value. When listening to the patient's lungs with a stethoscope, the doctor can better judge the presence of problems with them and better assesses their general condition.

Rule	Support value	Confidence value	Lift value
{JIADiagnosis. (Finding.fever)} ⇒ {JIADiagnosis. (Finding.erythematous_rash)}	0.010	0.812	57.439
{Surgery. (Finding.surgery_performed), Surgery. (MedicalProcedure.resonance_magnetic_material_imaging_contrast & Chemical.metal)} ⇒ {Surgery. (MedicalProcedure.tumour_removal)}	0.012	0.815	46.137
{JIALaboratoryOPBG. (Immunology.antibody_antinuclear)} ⇒ {JIALaboratoryOPBG. (Immunology.rheumatoid_factor)}	0.0133	0.85	37.951

Figure 18: The result of an associative rule obtained in the context of Reports

This Figure 18 has some really interesting context-based rules Report. At observation at the patient of a fever in most cases it specified on the appearance of erythema, indicating a complex inflammation of the joint or tissues.

Most of the operations before which magnetic resonance imaging is performed with using additional chemical compounds of iron, were just for removal tumors. However, with high Confidence and Lift value, these rules are low Support value, which indicates the small number of occurrences of these sets in transactions.

6. Conclusions

Summarizing all the above, research in this work was directed to consider the basic concepts and search for AR in both traditional ways and in inhomogeneous data of semantic networks, which creates certain problems when using existing algorithms. It is worth noting that one of the most popular areas of application for AR search still remains consumer and marketing. Semantic network data in most cases serve for a specific field and are highly specialized. Probably that's why direction you can do a lot of interesting research, one of which is the search associations among heterogeneous data.

A new method for finding ARs from inhomogeneous ones was also presented data in semantic networks expressed in RDF / (S) and OWL. Previously, this problem considered only to a small extent. Experiments on real SW data show good results. An interesting problem for future work is data mining in the ontology for filtering and cutting off the detected rules. Yet one important area that can be considered in the future concerns combination of clustering and AR search algorithms for generalization of arrays documents. This technology has previously been implemented to some extent hierarchical clustering of sets (FIHC). Basically, the FIHC algorithm generates clusters of sets of elements, which, in turn, make up the cluster descriptors. A new approach based on hierarchy has also recently been proposed element sets, which provides more homogeneous clusters and better descriptions than those obtained from FIHC. Undoubtedly, each algorithm can be improved and improve, apply better ways of embedding data to generated transactions and study their effectiveness. It is no less interesting development of new data exchange algorithms based on SW data and are accelerated by new ways of processing the generated transactions.

7. References

[1] C. Giannella, H. Jiawei, P. Jian, Mining frequent patterns in data streams at multiple, in : ESMA 2018 IOP Conf. Series: Earth and Environmental Science, 2019, pp. 61-84. doi:10.1088/1755-1315/252/3/032219

[2] B. Patel, Vishal H. Bhemwala, A. PatelAnalytical, Study of Association Rule Mining Methods in Data Mining. International Journal of Scientific Research in Computer Science Engineering and Information Technology, Vol. 3, 2018, pp. 818-831. DOI:10.32628/CSEIT1833244

[3] N. Boyko, K. Kmetyk-Podubinska, and I. Andrusiak, Application of Ensemble Methods of Strengthening in Search of Legal Information, in: Lecture Notes on Data Engineering and Communications Technologies, Vol. 77, 2021, pp. 188-200. https://doi.org/10.1007/978-3-030-82014-5_13

[4] P. Jian, H. Jiawei, B. Mortazavi-Asl, PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth, in : Proc of the 17th International Conference on Data Engineering, 2001, pp. 215-224.

[5] N. Boyko, N. Tkachuk, Processing of Medical Different Types of Data Using Hadoop and Java MapReduce, in: The 3rd International Conference on Informatics & Data-Driven Medicine (IDDM 2020), Växjö, Sweden, November 19-21, 2020 pp. 405-414.

[6] L. Duanyang, F. Jian, L. Xiaofan, A frequent sequence pattern mining algorithm based on logic. Jornal of Computer science, Vol. 42 (5), 2015, pp. 260-264.

[7] W. Xindong, X. Fei, H. Yongming, A sequence pattern with wildcards and one-off conditions Dig. Software journal, Vol. 24(8), 2013, pp. 1804-1815.

[8] Zh. Jialu, Y. Jun, H. Jing, Constrained association rule mining based on a set of transaction ids Mining algorithm. Computer engineering and design, Vol. 34(5), 2013, pp. 1663- 1667. doi:10.1088/1755-1315/252/3/032219.

[9] N. Boyko, A look trough methods of intellectual data analysis and their applying in informational systems, in: XIth International Scientific and Technical Conference Computer Sciences and Information Technologies (CSIT), 2016, pp. 183-185.

[10] H. Hong Juan, Zh. Jian, Ch. Shaohua, Constraint maximum frequent itemset mining based on frequent pattern trees Algorithm. Computer engineering, Vol. 37(9), 2011, pp. 78-80.

[11] W. Xiuzhi, Association classification algorithm based on intelligent optimization of support and confidence. Computer applications and software, Vol. 30(11), 2013, pp. 184-186.

[12] Ch. Shutong, X. From Rich, B. Hongwei, Research on efficient privacy protection frequent pattern mining algorithm. Computer science, Vol. 42(4), 2015, pp. 194-198.

[13] Ch. Aidong, L. Guohua, F. Fan, Uncertain data association rules that satisfy uniform distribution Mining algorithm.Computer research and development, Vol. 50, 2013, pp. 186-195.

[14] Sh. Yan, D. Min, L. Qiliang, Mining methods for association rules of Marine and continental climate events. Journal to Ball information science, Vol. 16(2), 2014, pp. 182-189.

[15] R. Idoudi, K. S. Ettabaa, B. Solaiman, K. Hamrouni, Ontology Knowledge Mining Based Association Rules Ranking, in: Procedia Computer Science Published by Elsevier, Vol. 96, 2016, pp. 345-354. DOI: 10.1016/j.procs.2016.08.147

[16] R. Paul1, T. Groza1, J. Hunter and A. Zankl, Semantic interestingness measures for discovering association rules in the skeletal dysplasia domain. Journal of Biomedical Semantics Vol. 5(8), 2014, pp. 2-13.

[17] O. Daramola, A. Ibukun, O. Okuboyejo, Semantic association rule mining in text using domain ontology. International Journal Metadata Semantic Ontology, Vol. 12(1), 2017, pp. 12-28.

[18] M. Barati, Q. Bai, Q. Liu, Mining semantic association rules from RDF data. Knowledge Based System, Vol. 133, 2017, pp. 183–196.

[19] L. Galárraga, C. Teflioudi, K.Hose, F.M. Suchanek, Fast rule mining in ontological knowledge bases with AMIE++. The International Journal on Very Large Data Bases, Vol. 24, 2015, pp. 707–730.

[20] L.A. Galárraga, C. Teflioudi, K. Hose, F. Suchanek, AMIE: Association Rule Mining Under Incomplete Evidence in Ontological Knowledge Bases, in: Proceedings of the 22nd International

Conference on World Wide Web, WWW'13, Rio de Janeiro, Brazil, 13–17 May 2013; ACM: New York, NY, USA, 2013; pp. 413–422.
[21] K. A. Kale, R.P. Sonar, Review on Mining Association Rule from Semantic Data. International Journal of Computer Science and Information Technologies, Vol. 7 (3), 2016, 1328-1331