**Implementation of SWapriori Algorithm**

Bryceleen D’souza1, Leon D’souza 2, Mugesh Nadar 3

1Department of Computer Engineering, FRCRCE

2Department of Computer Engineering, FRCRCE

3Department of Computer Engineering, FRCRCE

***Abstract-* Linked Data is used in the Web to create typed links between data from different sources. Connecting diffused data by using these links provides new data which could be employed in different applications. Association Rules Mining (ARM) is a data mining technique which aims to find interesting patterns and rules from a large set of data. With the introduction of the semantic web and its standardization as the third generation of the Web, has brought together more human attention than ever. Also the amount of semantic web data is constantly growing day by day. These semantic web data are a rich source of useful knowledge for feeding data mining techniques. Semantic web data have some complexities, such as the heterogeneous structure of the data, the lack of exactly defined transactions, the existence of typed relationships between entities etc. Association rule mining which is the data mining technique helps to find interesting rules based on frequent item-sets and also tries to fulfill complex applications by discovering and composing available services automatically and precisely, it is indispensable to develop an underlying model and the corresponding measure for semantic associations among given Web services. We thus propose a method that takes into consideration the complex nature of semantic web data and, without end-user involvement and any data conversion to traditional forms, mines association rules directly from semantic web datasets at the instance level. Here we assume that data is been stored in triple format (Subject, Predicate, and Object) in a single dataset. Thus for evaluation purposes the method will be applied to a drugs dataset and will show the ability of the proposed algorithm for mining ARs from semantic web data without end-user involvement.**

1. INTRODUCTION

Linked Data and Association rule mining(ARM) is a practical technique used to achieve Semantic Web goals. Linked Data is defined as "a set of best practices for publishing and connecting structured data on the Web". Hence many natural features of semantic web data, exists also in Linked Data. Extending the scope, we can say that data mining research from traditional data to semantic web data along with its Linked Structure could help in discovering and mine richer and more useful knowledge.

ARM, one of the main data mining techniques, tries to find frequent itemsets and based on these frequent itemsets, generates interesting association rules (ARs).

In trying to apply ARM to semantic web data, problems to face and differences compared with traditional data are as follows:

1) Heterogeneous data structure: Traditional data mining algorithms work with homogeneous datasets in which instances are stored in a well-ordered sequence and each instance has predefined attributes. But in semantic web data, data are heterogeneous. This means that specific

category/domain instances (such as people, cars, drugs and etc.) based on one ontology or multiple ontologies may have different attributes.

2) No exact definition of transactions: In conventional information systems, data are stored in databases using predetermined structures, and by using these structures it’s possible to recognize transactions and thus extract them from the dataset. Then traditional association rule mining algorithms work on these transactions.

3) Multiple relations between entities: Traditional ARM algorithms, in order to generate large itemsets4, consider only entities' values and suppose there is only one type relation between entities (for example bought together). But in semantic web data, there are multiple relations between entities. In fact predicates are relations between two entities or between one entity and one value. These different relations must be considered in the ARM process [4]

An algorithm, named SWApriori, has considered the above challenges and without the end user involvement, mines ARs directly from a single semantic web dataset. So we could state that the mentioned problem has been solved. The problems and challenges of collecting desired data, from different Linked Data sources, and the ARM of it has been investigated. Thus the suggested approach collects data from various data sources and connects them so that all data appear as a single and central dataset and then uses existing methods to mine ARs from the generated dataset.

2. PROPOSED ALGORITHM

* Inputs:
  + Dataset
  + Minimum Support
  + Minimum Confidence
* Outputs:
  + Large Data-Item
  + Association Rule

Algo1.png

Algo1.png

* Inputs:
  + List of object info instances
  + Minimum Support
* Outputs:
  + List of large Itemset

Untitled Diagram (2).png

* Inputs:
  + All large itemset
  + Minimum Confidence
* Outputs:
  + Association Rules

Untitled Diagram (2).png

Algorithm3.png

3. LITERATURE SURVEY

The ARM was first introduced with the aim of finding frequent itemsets and generating rules based on these frequent itemsets. Many ARM algorithms have been proposed which deal. with traditional datasets. These algorithms are classified into two main categories: Apriori based and FP-Tree based. These algorithms usually work with discretized values, but later an evolutionary algorithm was introduced for mining quantitative ARs from huge databases without any need to data discretization.

As will be seen, semantic web dataset contents are convertible to graph. Other related approaches in ARM are the use of frequent sub-graph and frequent sub-tree techniques for pattern discovery from graph structured data. The logic behind these algorithms is to generate a tree/graph based on existing transactions and then mine the generated tree/graph. Although these methods are interesting, they are not appropriate for our work, because in semantic web data there is no exact definition of transactions, and also after converting dataset contents to graph, each vertex of the graph, independent of its incoming link, is not replicated in the whole graph more than once. On the other hand, graph vertices are unique and thus discovering sub-graph/sub-tree redundancy is not possible.

Not all graph-based approaches are based on sub-graph techniques. In an algorithm has been introduced that inputs data into a graph structure and then by a novel approach without the use of sub-graph redundancy, mines ARs from these data. This work is not useful for our problem because the algorithm finds only maximal frequent itemsets instead of all frequent itemsets and also, like other traditional ARM algorithms, this algorithm works only with well-defined transactions.

**ARM ALGORITHMS**

In [1] an algorithm has been introduced that by using a mining pattern which the end-user provides, mines ARs from semantic web data. This algorithm uses dynamic and graph-based structure data that must be converted to well-structured and homogeneous datasets so that traditional ARM methods can use them. To convert data, users must state the target concept of analysis and their involved features by a mining pattern following an extended SPARQL syntax. This work, similarly to other related approaches in mining ARs from semantic web.

1. The software requires semantic web datasets obtained by using user-defined SPARQL queries. Then clubs the results and converts them in tabular format.

2. Next pre-processing has to be done on these data and then the traditional data mining algorithms is applied.

The actual problem with the LiDDM is the end user involvement during the mining process. Thus one needs to be aware of the ontologies and dataset structure. Thus he needs to gude the mininfg process step by step.

Another approach similar to the LiDDM is the RapidMiner semweb plugin. This technique allpied the data mining technique on the semantic web and linked data along with reformatting set-valued data, such as converting multiple values of a feature into a simple nominal feature to decrease the number of generated features and thus the approach scales well. Even in this process the end user involvement is required.

In RDF structure, a triple defines every data statement names and is identified with three values: subject, predicate and object. In order to generate transactions, it is possible to use one of these three values to group transactions (transaction identifier) and use one of the remaining values as transaction items. Six different combinations of these values along with their usage are shown in Table 1 [3]. For example, grouping triples by predicate and using objects for generating transactions has usage in clustering. This approach eliminates one part of triples parts and doesn't consider it in mining process that isn't interested.

SPARQL-ML [4] is another approach to mining semantic web data that provides special statement as an extension to SPARQL query language to create and learn a model for specific concept of retrieved data. It applies classification and regression techniques to data, but other data mining techniques such as clustering and ARM are not covered by this approach. Another limitation is that this technique is applicable only on those datasets for which the SPARQL endpoints support SPARQL-ML, which is currently not very widespread. Our proposed algorithm can deal with all kinds of datasets and ontologies.

4. PROPOSED SYSTEM

**4.1** **MINING** **ASSOCIATION** **RULES** **FROM** **LINKED** **DATA**

In order to mine ARs from Linked Data, desired data has been collected from two datasets (DBPedia and Factbook) and then converts them to a singled and central dataset and finally mines ARs by using the proposed algorithm in [5]. Next sections show the datasets used and acquired results.

This part describes the used methodology of collecting desired data.

1) Selecting Domain: The generated ARs quality depends on the input dataset quality and how much the provided data is being specialized in a domain, the generated rules are more specific and more useful respectively. Also because generalized data are diffused, the MinSup value needs to be low so that these generalized rules appear in the result. Currently we selected Country as data domain.

2) Selecting Datasets: In next step we select suitable datasets that have appropriate data about countries. As mentioned earlier, there are two approaches to automatically select suitable datasets. Our strategy for selecting datasets is to consider a dataset as the start point and extract desired data from it, afterward select another dataset based on the extracted data and finally traverse these new selected datasets. This operation will be continued until a special criterion is being satisfied (for e.g. traverse at most 5 datasets). The selected dataset for start point is very important. In addition to have many links to other datasets, the start point dataset must have suitable data about selected domain. DBPedia1 has been selected as start point dataset since it has many links to other datasets.

3) Connecting Datasets: There are two ways for connecting related data of different datasets. The first way is to acquire suitable data independently and then append them to each other or connect them by using common attribute [6]. The second way is to collect data from a dataset and then traverse another datasets by using objects from triples of collected data. The latter method requires an explicit relation among datasets- i.e. object of a triple refers to an entity (a subject) that exists in another dataset. These explicit relations are grouped into two relations' groups: more information relations and equivalent relations. More information relations are those relations that state one attribute of an entity which has been defined and exist in another dataset. Equivalent relations are those relations that state two entities in two different datasets are equivalent. The most common equivalent relation is owl:sameAs. It's clear that using more information relations don't help to collect suitable data about a domain since these relations only show external attributes causing irrelevant entities in the collection process. This is why only equivalent relations have been used to collect suitable and relevant data after collecting primary data from start point dataset.

4) Ontology Mapping: As mentioned before, due to existence of multiple data sources, ontology mapping must be applied on data so data become coherent. There are many approaches do ontology mapping [7-9]. For this project, ontology mapping will been done manually so that data become suitable for ARM by integrating subjects and predicates- i.e. detecting identical subjects or predicates with different names, and also to scale numerical objects.

5) Duplicated Data: This problem can be removed by selecting the best and the most valid data and eliminate other duplicates. Thus, the issue can be solved by selecting data from specialized dataset and eliminating other duplicated data.

6) Last step: After collecting desired data and mapping ontology and removing duplicated data, the data have to be placed in a single and central dataset with a unified ontology.

**4.2 DATA STRUCTURE**

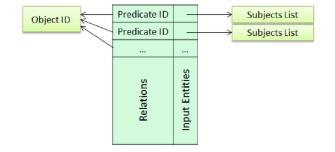
The algorithm inputs a set of triples (subject, predicate and object). For the purpose of storing data in main memory, the simplest and the most efficient way is to use a cuboid (3D array) as data structure, in such a way that the first dimension stores source (subject), the second stores destination (object) and the third stores relation (predicate) between source and destination. Each cuboid entry value is 0 or 1. If the (i,j,k)th entry value is equal to 1, this means there is a relation with k type from ith entity (as subject) to jth entity (as object). Although a cuboid structure is very fast and easy to use it requires a large amount of memory space. An alternative is to use a linked list data structure. To store each object scheme (predicates and subjects that are connected to the object), there is an ObjectInfo class with these attributes:

1- Object ID: Object identifier

2- A Linked List that its entries have two parts:

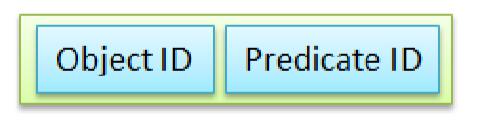
1. Predicate ID: Predicate identifier
2. Subjects List: pointer to a list that contains subjects which refer to this Object ID with this Predicate ID.

The ObjectInfo image has been depicted in Figure 4.1



**Figure 4.1 ObjectInfo Structure**

Using this data structure policy, triples are grouped based on objects and the algorithm defines an ObjectInfo instance and then specifies them based on each predicate, what other subjects refer to this object. The need for grouping is to increase the mining process speed based on the proposed algorithm. Finally there is a list that has entries equal to the objects count. Each entry of this list refers to one of the ObjectInfo instances. This algorithm, in addition to entity values, considers relations between entities in the ARM process. Thus here each Item not only is equal to an entity but also each Item consists of an Entity (Object) and a Relation (Predicate) that is connected to that object. To store each Item there is an Item class that has ObjectID and PredicateID attributes. Figure 4.2 shows the image of class Item.

****

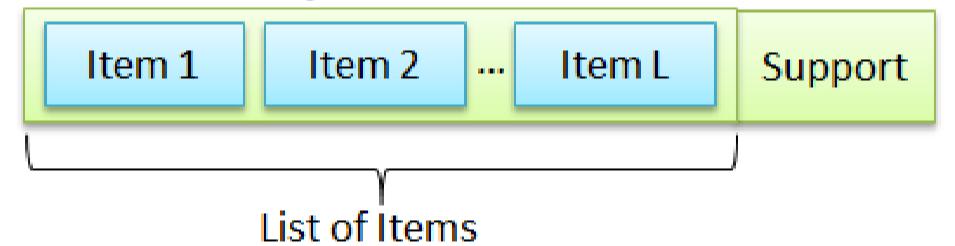
**Figure 4.2 Item Structure**

Generating ARs is based on large itemsets. Each itemset is non-empty set of Items. In order to storing generated (candidate/large) itemsets, there is an Itemset class that contains these attributes:

1- List of Items: that holds L items ( ≥ 2). |

2- Support: number of subjects that refer to all Items via correspond predicates.

The Itemset is large if Support is equal to or greater than MinSup value. Figure 4.3 shows the image of class Itemset.



**Figure 4.3 - Itemset Structure**

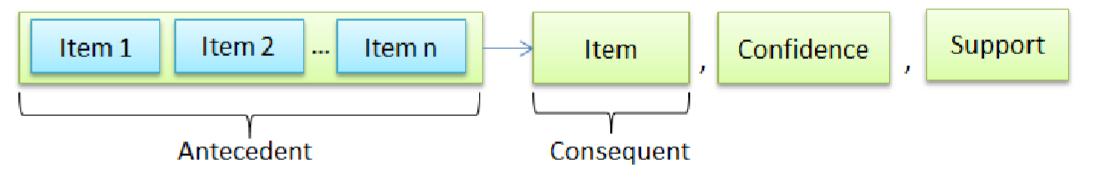
It was said that ARs are constructed from Items and each rule has only one Item in the consequent part. To store generated ARs, there is a Rule class that contains these attributes: 1- List of Items as Antecedent

2- An Item as Consequent

3- Rule Confidence

4- Rule Support

In Figure 4.4 you can observe the Rule class image.



**Figure 4.4 Rule Structure**

5. CONCLUSION

Linked Data is used in the Web to create typed links between data from different sources. these links and connecting datasets helps us to create new connected data that can be used in data mining. We thus try to solve the problems of link data query challenges and the problem of mining ARs from Linked Data. Then, by considering these challenges we launched to extract and connect desired data and put them in a single and central dataset. Finally, by using the proposed algorithm in [1] ARs were mined from this new created dataset.

Certain difficulties that could be found during the implementation of the proposed system is as follows:

1. The method is not intelligent enough to involve meaning of data (provided by ontology) in the mining process to guide the process intelligently and generate only interested and useful rules.
2. If the content of the input data is general and the end-user does not filter it, the number of generated ARs would be enormous and a large part of them may be uninteresting.

These difficulties can overcome during the future work where we can try to improve the system to avoid the problems.

6. REFERENCES

1. R. B. V.Nebot, "Finding association rules in semantic web data," Knowledge-Based Systems, pp. 51-62, 2012.
2. M. A. Khan, Gunnar Aastrand Grimnes, and Andreas Dengel, "Two pre-processing operators for improved learning from semantic web data," presented at the In First RapidMiner Community Meeting And Conference (RCOMM), 2010
3. F. N. Ziawasch Abedjan, "Context and Target Configurations for Mining RDF Data," presented at the SMER '11 Proceedings of the 1st international workshop on Search and mining entity-relationship data 2011.
4. A. B. Christoph Kiefer, André Locher, "Adding data mining support to SPARQL via statistical relational learning," in ESWC'08 Proceedings of the 5th European semantic web conference on The semantic web: research and applications methods, 2008, pp. 478-492
5. R. Ramezani, "Finding Association Rules in Linked Data," M.Sc. Thesis, Computer & Electerical Engineering Department, Isfahan University of Technology, Isfahan, Iran, 2012, The proposed algorithm for ARM from semantic web data (SWApriori) will apear in a paper titled "A New approach to mining Association Rules from Semantic Web data". This paper is under publication.
6. R. I. V.Narasimha, O.P.Vyas, "LiDDM: A Data Mining System for Linked Data," presented at the Proceedings of the LDOW2011, Hyderabad, India, 2009.
7. L. Huang, Guoxiong Hu, and Xinghe Yang, "Review of ontology mapping," presented at the Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on. IEEE, 2012.
8. I.-Y. S. Namyoun Choi, Hyoil Han, "A Survey on Ontology Mapping," presented at the ACM SIGMOD Record, 2006.

M. S. Yannis Kalfoglou, "Ontology mapping: the state of the art," The Knowledge Engineering Review, vol. 18, pp. 1 - 31, 2003.