www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issue 10 Oct. 2016, Page No. 18363-18371

Enhanced Way of Association Rule Mining With Ontology

T.Bharathi, Dr. A.Nithya
Dept.of Computer Science
RVS Arts and Science College
Coimbatore, Tamilnadu, INDIA
pbharathi36@gmail.com,

Dept. of Computer Science
RVS Arts and Science College
Coimbatore, Tamilnadu, INDIA
nithy.arumugam@yahoo.com

Abstract – For a data mining process, association rule mining is one of the key components. In the realm of the mining of the data, association rules play a key role to condense interesting correlations, frequent patterns, associations or casual structures from the set of items present in the databases. The current paper focuses on the aspect of the competent mining of the association rules from larger databases. The problem of unearthing of large item sets can be sorted by the creation of an ontology tree. In the domain of instructional design and in the evolution of the course content, ontology plays a very vital part. The understanding about the content can be depicted with the help of ontology trees that in turn would aid the instructors in development of the content and the learners to get permission to use the content in an apt way. Even though ontologies are there for many domains, their fittingness for other subjects is still vague. Further, for many other domains, the ontologies even don't exist. Many have attempted to devise methods to enhance many dimensions of the ontology, namely, representation languages and inference mechanisms. But, unfortunately very less effort has been taken to improvise the practical results of development method application. In this paper, a discussion on the technique of Association rule mining with ontology (ARMO) is presented that is employed to find the most precise association rules in the area of ontology, ontology analysis, ontology tree and frequent item sets. The spotlight is more on the relationship type that permit one to model rich rules adequately.

Index Terms: Association rule mining, Ontology, fitness, velocity

I. INTRODUCTION

Mining of the association rules is regarded as one of the most pertinent responsibilities for Knowledge Discovery in Databases. The purpose behind is to find out the latent information that can be unearthed as an important piece of data when dealing with items in a large database. The allusion X→ Y is a way to depict the association rule, in which, X and Y are sets of items. The ability by which the association rule can extract useful information from the existing data determines the strength of association rule mining. But in huge databases, the strength becomes the weakness while the mining results are analyzed. The task for the decision maker to extract the interesting rules becomes difficult when the number of discovered rules is very huge. This implies that a way has to be found out so as to make the job of the decision maker simpler by decreasing the number of rules. As a solution to the above mentioned problem, it was proposed to include the post-processing task to aid in selection of the discovered rules. There are numerous techniques available for the post-processing. Excluding the unimportant or unnecessary rules is done in the pruning phase. The very first algorithm that was proposed in the domain of the association rule mining was the Apriori that was employed to find the frequent item set. Many more algorithms followed after that. The problem with many conventional algorithms is that that do not consider the interest of each item within the analyzed data. Hence, by application of these algorithms, the quality improves but the interestingness of the patterns are not taken into consideration. Besides, the rules were produced in a large number.

The concept of the Ontology Relation together with the algorithm particle swarm optimization that is also known as Association Rule Mining with Ontology (ARMO) was introduced to provide a solution for the above mentioned shortcoming. By doing so, it permits the user to focus on certain selected schemas of rules so that only those subset of rules would be chosen.

We have proposed a novel technique to shorten and filter the discovered rules in the current paper. The amalgamation of the user knowledge with the post processing task is further strengthened by the usage of the Domain Ontologies. Along with that with the aim of aiding the user in the analyzing task, we have designed an interactive and iterative framework. Furthermore, the Domain Ontology is employed to depict the user domain knowledge in the database. The notion of Rule Schema represents the user expectations and the rule operators guide the user actions. A very detailed and lucid depiction of the user knowledge, and rule schemas and rule operators is achieved by employing the ontologies. The user schema forms the basis for the creation of the ontology tree that is refined on the basis of the frequent item sets.

The paper is segmented for easy understanding. The issues related to the mining of the association rules is explained in Section 2. The reviews of the related work along with the description of the research domain is elucidated in section 3.

The proposed algorithm is made clear in section 4. Section 5 elaborately discusses the proposed method to mine the association rule in the most proficient manner by using the AMRO algorithm. The various elements are - the mining process, the user knowledge and the post processing step. Section 6 analyzes the results of the experiment and lastly, the conclusion and the path for further research if explained in section 7.

II PROBLEM DESRIPTION

To discover the relationships in an enormous database that are otherwise latent, one uses association analysis. The relationships that are not covered can be depicted in the form of association rules or sets of frequent items. Consider for an instance, the rule that can be condensed from the table 1 given below is as -

{Diapers} → {Beer}

T ID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

Table 1: transactional database

From the rule, it is very much evident that there is a definite relationship between the sale of diapers and beer as it is quite clear that many buyers who purchase diapers also purchase beer. Vendors can utilize such rules to recognize such instances to increase their sale.

In addition to the retail sector, the association rule mining also finds use in the area of other application domains such as bioinformatics, medical diagnosis, Web mining, and scientific data analysis. For instance, in the domain of Earth science, when we analyze the data, the association patterns may reveal interesting connections among the ocean, land, and atmospheric processes. These kind of data might assist the scientists to comprehend the relation between the various elements of the earth in a better manner. The main focus of this research paper is on the transactional data in the retail sector.

The important issues that are required to be attended to when applying association rule to market transactional data are - the cost involved in the discovery of patterns that form a large transaction database and the authenticity of the discovered patterns. The paper is divide into two segments. The basic concepts of the association rule mining and the existing algorithms that are employed for the efficient mining are detailed in the first segment. In the second segment, we concentrate on the evaluation of the discovered patterns so that bogus results may be prevented from being generated with the help of the proposed method.

III RESEARCH DOMAIN AND RELATED WORKS

Association Rule Mining:

The if/then statements that aid in unearthing the correlations between unrelated data in a database, relational database or any other information database is called as the association rule. To find the correlations amongst the items that are often purchased together association rules are employed. The numerous areas where the association rules are employed are - basket data analysis, classification, cross-marketing, clustering, catalogue design, and loss-leader analysis etc. For instance, we can consider that a buyer when purchases cornflakes will purchase milk as well. If a buyer purchases a mobile phone, then will purchase a case as well. Support and confidence are the two important criteria that is used by the association rules that are in turn employed to analyze the data patterns generated by the rules. These rules should meet the user-specified minimum support and a user –specified minimum confidence simultaneously.

$$Rule: X \Rightarrow Y \quad \begin{cases} Support = \dfrac{frq(X,Y)}{N} \\ Confidence = \dfrac{frq(X,Y)}{frq(X)} \end{cases}$$

Why Use Support and Confidence?

To eliminate any rule with low support that may appear just like that, the value of support is very crucial. Also, a rule that has low support value may not be useful from business point of view as it may not be profitable for the vendor to stock those items that seldom are purchased together by a customer. In such cases, rules with low support values will be eliminated. Besides, by exploiting the features of the support, one can discover efficient association rules.

On the other hand, confidence gives an idea about the dependability of the inferences drawn from any rule. For instance, if the rule is A→ B, and the confidence is also higher, then it is highly possible that B to be present in those transactions that have A. One should be cautious enough while interpreting the results of the association rules. The inference made by an association rule does not necessarily mean causality. But, it only suggests a correlation between the antecedent and consequent of the rule. Any association rule mining algorithm has a method that involves disintegrating the problem into two major steps:

- Frequent Itemset Generation, whose aim is to discover all the item-sets that satisfy the minsup threshold. These itemsets are called frequent item sets.
- Rule Generation, whose aim is to extract all the high-confidence rules from the frequent itemsets generated in the previous step. These rule are called strong rules.

Frequent Itemset Generation

There are many ways to lessen the computational complexity of frequent itemset generation.

- Reduce the number of candidate itemsets (M). The Apriori principl is an effective way to eliminate some of the candidate itemsets without counting their support values.
- Reduce the number of comparisons. Instead of matching each candidate itemset against every transaction, we can reduce the number of comparisons by using more advanced data structures, either to store the candidate itemsets or to compress the data set.

Candidate Generation and Pruning

The existing Algorithm generates candidate itemsets by performing the following two operations:

- Candidate Generation. This operation generates new candidate k- itemsets based on the frequent (k − 1)-itemsets found in the previous iteration.

- Candidate Pruning. This operation eliminates some of the candidate k-itemsets using the support-based pruning strategy.

The numerous mining methods that are being used are as given below:
AIS
SETM
Apriori
AprioriTID
Apriori hybrid
Eclat
Recursive Elimination
FP-Growth
Dyn- FP growth
Matrix Apriori
PSO

ONTOLOGY: What is ontology?

The process of comprehending domains those are of interest that may be utilized as a unifying framework is referred to as ontology. It encompasses certain view with respect to a given domain. The world view contains set of concepts with the definitions, the correlations that are otherwise called as conceptualization. It might be often implied. If we take an example of accounting package, it is assumed that it encompasses concepts like proof of purchase. This implied conceptualization is sometimes referred to as ontology.

What does ontology look like?

Motley of vocabulary along with their meaning forms an implied ontology. The vocabulary and the meaning are created as per the degree of formality that usually varies. The arbitrary points as per which the range are as mentioned below:
highly informal – considered loosely in natural language
semi informal – they are expressed in a very confined and organized manner in the natural language that in turn results in more lucidness and thereby reducing the uncertainty.
semi formal - expressed in an artificial formally defined language rigorously formal - painstakingly defined terms with formal semantics theorems and proofs of such properties as soundness and completeness.

Types:

The various kinds of ontology that exist as per the scope, language expressivity and domain are as mentioned below:
Information ontology
Software ontology
Gene ontology
Linguistic / terminological ontology
Formal ontology
Application ontology
Domain ontology
Core reference ontology
General ontology
Upper level ontology

Different Ontology Design Approaches

There are many techniques that are employed to design ontology. Meth ontology and On-to-knowledge are amongst the most complete ones. But again, this domain is still under improvement. There are many activities involved in all these techniques. There is no linearity in the development process where every activity can be recapitulated again. The most key activities are:
Ontology specification
Knowledge acquisition
Conceptualization
Formalization
Implementation
Evaluation
Maintenance
Documentation

Design Criteria :

We propose to employ domain ontology in our proposed technique. The *design decision* involves the question of when and how to represent something when we view ontologies as artefacts. This permits the evaluation of the artefact with respect to the objective criteria rather than with the *truth*. The five design criteria that are taken into consideration are as explained below:

1. **Clarity –** the definitions must be proper, absolute, objective and independent of social or computational context. Therefore, a large number of potential interpretations of a concept is restricted that in turn contributes towards the effective communication amongst the agents.
2. **Coherence –** permit only those inferences that are consistent with existing definitions.
3. **Extendibility –** the design of the ontology is such that it should act as a conceptual foundation for a range of anticipated tasks. The extension of the ontology must have no effect on the existing definitions. This result in a situation wherein there is no necessity to incorporate a vocabulary adequate enough to express the knowledge related to all anticipated extensions as we already have the methods to define the necessary specialisations.
4. **Minimal encoding bias** – one should stipulate the conceptualisations at a knowledge-level itself. Convenience of notation or implementation issues at a symbol-level must not be the reason for the selection of the representation choices.
5. **Minimal ontological commitment –** permission for the least ontological commitment that is required to sustain the intended knowledge sharing activities must be given. Again the weakest theory essential for a consistent communication should be mentioned. This ensures that the agents have the freedom to extend the ontology when a need arises.

It is noteworthy that the most important aspect of a shared ontology is that it is only necessary to describe a vocabulary for *communicating about* a domain. But on the other hand, a *knowledge-base* has the knowledge that is required to solve problems or answer queries about such a domain by committing to ontology. The objective of the ontology is to capture the conceptual structures of a domain whereas a knowledge-base aims to specify a concrete state of the domain. In other words, ontology encompasses the *intentional* logical definitions whereas a knowledge-base consists of the *extensional* parts.

Domain ontology:

Domain ontology can be applied to a domain that has a particular view point that describes how a group of users conceptualize and visualize some specific phenomenon. It can be linked to certain application like electric network management system. Consider the following instance - the establishment of the domain , the source and the objective and the scope of the ontology forms the first step. Queries that must be covered at this stage involve - what domain will the ontology cover? , what is the purpose of the ontology? And for

what sorts of questions should the information in the ontology be able to provide answers? Prior to answering the design related issues, the domain of the investigation has to be decided. Database systems were chosen as the domain. Behind the decision were present numerous factors from the research point of view. The subject area was thus chosen that would be broad in the sense that it would result in a large number of concepts and associated relationships. With the help of which, the initial hypothesis was tested and it was proved that the 'Is-a' relationship is sufficient to express the semantics. The figure1 below depicts the domain operator and the division of the state based on ontology.

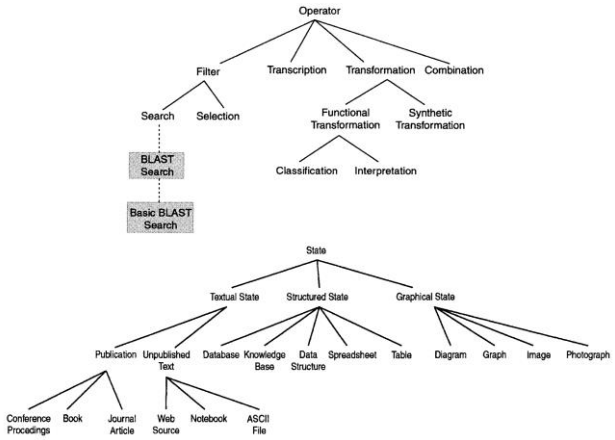


Fig.1 domain ontology

Uses of Ontology:

This section reviews and gives a detailed purpose for the usage of ontologies. This in turn leads in characterising the uses of the ontologies. The literature contains the elaborate details and the uses of the ontologies. Some of them even appear as if they can be reused. Some of these purposes are implicit in the various interpretations of the word ontology that are commonly found in the literature. The software that will be used along with a particular ontology forms the other dimension. The decision is between if the ontology is proposed to be shared within a small group and again used or to be used within a larger community. The ontology is simply thought as a means to construct a knowledge base, whereas others think that ontology to be a part of the knowledge base. The other essential impetus for ontologies is that of incorporation of models from various domains into a coherent framework. Hence, the space of uses for ontology is further divided into -

- Communication
- Inter Operability
- Systems engineering: specification reliability and reusability

IV PROPOSED SYSTEM

There are four stages in the proposed method. In the first phase, the data is condensed from the database and is transformed into binary format that is in turn used to find the search for a particular product in an easier fashion and then use the domain ontology to divide the products into different domains for potential use. The calculation for the IR value is done in the second stage and on the basis of the IR value, candidate sets are generated. Also, the threshold value for the minimum support is set in this stage. On the basis of the domain ontology and its analysis, the ontology tree is created. The support value calculation and frequent item sets generation is done in the third phase. Then the frequent item set values are given to the Ontology tree. From the ontology tree, certain general rules are created in this stage. The reduction of the ontology tree on the basis of the frequent itemset happens in the fourth stage. Association rule with a high support value is mined from the reduced Ontology tree.

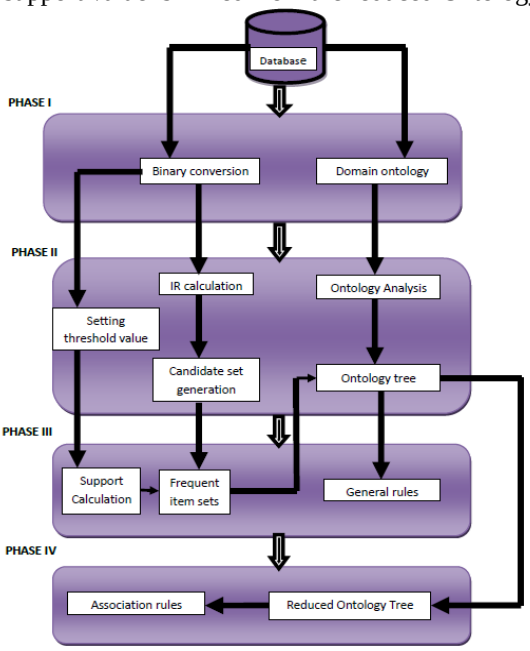


Fig.2 Proposed system

PHASEI:

Data Processing:

The database contains all kinds of data that might not be used for the processing of the rules. The required data will be condensed from the database in the first stage. The data pertaining to the customer id, product id that are bought by that customer and the time id is condensed from large amount of data sets.

Binary transformation:

The condensed data is then transformed into a binary format and stored either in 0 or 1 in a two dimensional array. This process accelerates the scanning process of the database and the calculation of the support value is done in a quick and easy manner. The transformation method is explained in the figure3.

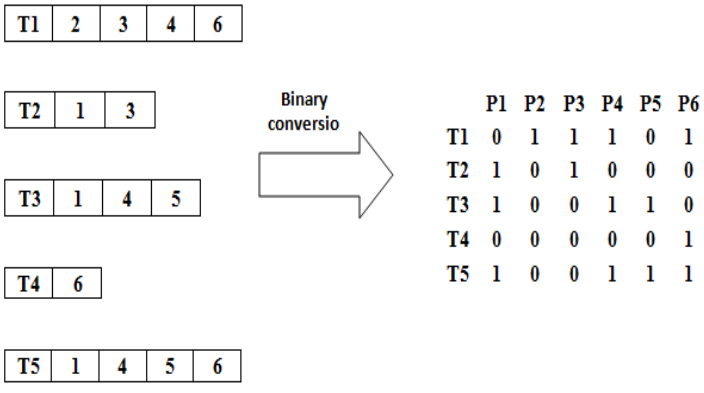


Fig.3 Binary conversion

Let us assume that there are five transactions, say T1 to T5 in the transactional database. Every transaction is converted into a binary array and stored. Let there be a total of six different products in the database. Hence, in the binary array, there will be six columns. Let us consider T5 as an example, wherein the items 1,4,5,and 6 have been purchased. So the columns 1,4,5,and 6 will have "1" and the columns 2 and 3 will have "0".

Domain Ontology:

Domain ontology facilitates the classification of the various products on the basis of different categories. Here some categories of food items are given for the supermarket.

Meat
Beverage
Sweet
Milk products
Fruit
Grain
Spice
Vegetables

PHASE II:

The objective of the IR calculation in process of association rule mining is the production of interesting rules that will in turn result in the improvement of the search efficiency. Also, the IR analysis helps in avoidance of unwanted searches that would result in rules that are not relevant. The IR value is calculated after the binary transformation of the data in the previous stage in accordance to the following function.

IR= [log(A*Trans(A))+log(B*Trans(B))][Trans(A,B)/Total_trans)]

Where 'A' is the length of the item set and Trans(A) denotes number of transactions that has A products. 'B' is the length of the item set and Trans(B) denotes number of transactions containing B products. Trans(A,B) denotes number of transactions purchasing A&B products. Total_trans depicts the number of total transaction in the transactional database.

Candidate item set generation & setting threshold value:

The candidate set is then generated based on the IR value. For instance, if the IR = 2.978, then the length of the item set is 3, implying that each item set has 3 products. Hence, we can generate a candidate set like { (1,8,3),(1,3,5),(2,4,6),(2,3,4),(3,5,6),(3,5,9)…..}. the user has the freedom to set the threshold value for the minimum support, which again depends on the size of the data base, number of products and number of transactions. In the current experiment, the threshold value has been set as 3.568. The item sets that have support value less than the threshold value will be eliminated from the frequent item set.

Ontology analysis:

The ontology analysis starts after the division of the domain. The key intention of using the ontology analysis is to discover the items that are regularly bought by the buyer in every transaction. Once the analysis is complete, we come to a conclusion about which items are repeatedly bought by the buyers. The ontology will help us to generate the rules for individual buyers that is nothing but the taking the experiment to the next level.

Ontology tree:

After the completion of the ontology analysis, on the basis of the domain and purchase history, ontology tree will be generated. The tree will help in generation of the rules that would have many replica and unnecessary transactions. Ontology tree for the individual customer can be created to get an idea about their pattern of purchase. The figure4 depicts a sample ontology tree for the food items. Similarly, one can create the tree for every product on the basis of the domain.

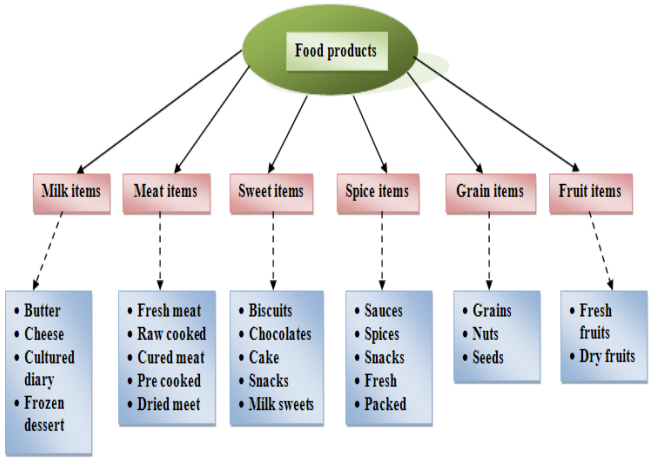


Fig. 4 Ontology Tree based on Domain

PHASE III:

Frequent item set generation:

Now the support value is calculated for initial association rule k (U=>V) using the following formula for the already created candidate set.

Support(U=>V)=Num_trans(u,v)/Total_trans.

Only those itemsets that have support value greater than or equal to the minimum support value will get included to the frequent item set. First, the support value for each product will be calculated. In this case, the rule is U=>U, i.e., the individual support value. The product which has the least support value will get eliminated and the rest of the products would get included to the frequent item set. In the one dimensional item set, the least value is 1. So the product with support value 1 will be removed from the candidate set.

One dimension:

Item set	support
{1}	4
{2}	5
{3}	4
{4}	3
~~{5}~~	~~1~~
{6}	10
{7}	8
~~{8}~~	~~1~~
{9}	6

product	support
{1}	4
{2}	5
{3}	4
{4}	3
{6}	10
{7}	8
{9}	6

Fig. 5 support value for 1-Dimension

As the next step, the item set will move towards the two, three dimensions on the basis of the support value. Hence, the candidate item set will become smaller and simultaneously the frequency item set will get new additions. This process will be repeated until threshold value is reached.

Two dimension:

DOI: 10.18535/ijecs/v5i10.24

Item set	Support
{1,2}	5
{1,3}	8
~~{2,3}~~	~~1~~
{2,6}	2
~~{3,4}~~	~~1~~
{4,7}	1
{4,9}	4
{6,7}	6
{7,9}	3

Item set	Support
{1,2}	5
{1,3}	8
{2,6}	2
{4,9}	4
{6,7}	6
{7,9}	3

Fig. 6 support value for 2-Dimension

General rules:

General rules will be created using the ontology tree. The basis for these rules will be the purchase history of the customer. Ontology analysis does the job for creation of such rules. By these rules, one is able to understand the pattern of purchase of a customer that would in turn help to serve the customer better. The figure 7 shows a sample ontology tree. In this example, the customer bought milk products regularly followed by the spice items. So we give preference to those items when the customer shops.

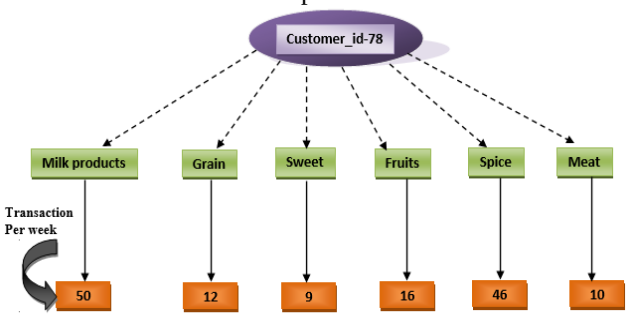


Fig. 7 Sample ontology tree

PHASE IV

Reducing the Ontology Tree:

The frequency item set acts as the input for the generation of the ontology tree. Hence, items that are not present in the frequency item set will not be included in the ontology tree. Only those items that have support value greater than or equal to the minimal support are included in the tree. The reduced tree helps in the generation of meaningful rules. The figure 8 depicts a sample reduced ontology tree.

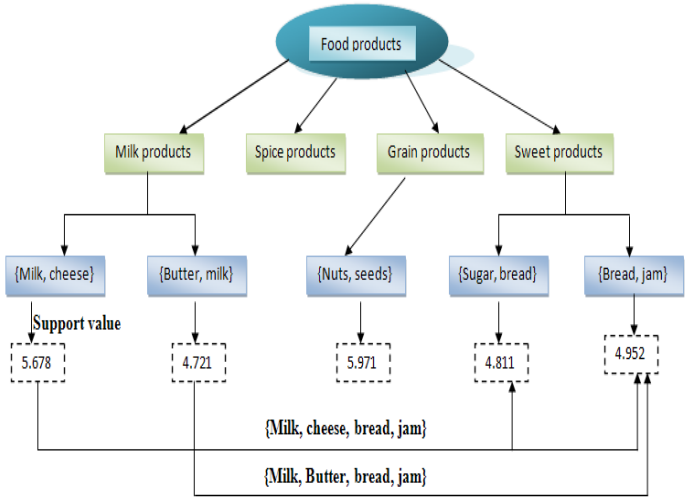


Fig. 8 Reduced Ontology Tree

Mining Association rules:

Once the ontology tree is reduced, association rules can be generated based on the least support value. If the support of the rule{ Milk=>Bread} and {Milk=>Butter} is high, then we can generate a rule with the measures which is {Milk}=>{Bread, Butter}. Similarly, we can generate rules for the reduced ontology tree. From the above final hash tree we can form the following rules.

{Bread}=>{Jam} and {Bread}=>{Milk} , so the rule is {Bread}=>{Jam,Milk};

{Milk}=>{Biscuit, Cheese} and {Milk}=>{Bread}, so the rule is

{Milk}=>{Biscuit, Cheese, Bread};

The prospect of buying the product set {Bread, Jam,Milk} and {Milk, Biscuit, Cheese, Bread} is high. The ontology tree discards the useless comparison of transaction from the beginning. It simplifies the generation of association rule using the frequent item set.

V EXPERIMENTAL RESULTS

Sample input/output:

Input:

Products=> p1= Bread
P2= Milk
P3= Jam
P4= Cheese
P5= Nuts
P6= Butter

Transactional history of 6 customers:

C1= {p1, p2, p4}
C2= {p3, p1, p6}
C3= {p1, p2, p3, p4, p5, p6}
C4= {p2, p4, p6}
C5= {p1, p2, p3}
C6= {p1, p2, p3, p6}

Binary conversion:

	P1	P2	P3	P4	P5	P6
C1	**1**	**1**	**0**	**1**	**0**	**0**
C2	**1**	**0**	**1**	**0**	**0**	**1**
C3	**1**	**1**	**1**	**1**	**1**	**1**
C4	**0**	**1**	**0**	**1**	**0**	**1**
C5	**1**	**1**	**1**	**0**	**0**	**0**
C6	**1**	**1**	**1**	**0**	**0**	**1**

Domain ontology:

For the given products, we can divide them into 3 fields. They are milk products, grains and sweet. Based on this we can create a domain ontology tree which is shown in figure 9.

DOI: 10.18535/ijecs/v5i10.24

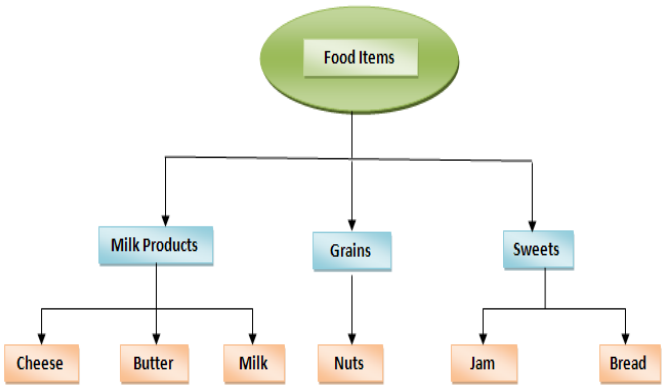


Fig.9 Domain Ontology

Support value for individual item:
Support (p1) = 5/6 = 0.83
Support (p2) = 5/6 = 0.83
Support (p3) = 4/6 = 0.66
Support (p4) = 3/6 = 0.50
Support (p5) = 1/6 = 0.16
Support (p6) = 4/6 = 0.66

Support value for individual item is included in the ontology tree shown in figure 10.

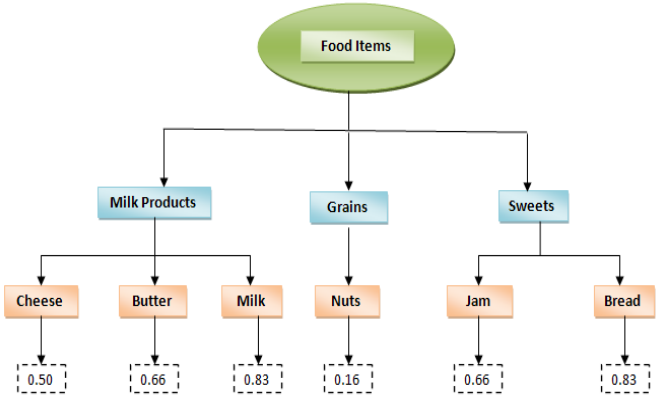


Fig.10 Ontology tree with support value

IR value Calculation:

IR=[log(x*Num_trans(x))+log(y*Num_trans(y))]
[Num_trans(x,y)/Total_trans)]

{p1, p2} = [log (1*5) + log (1*5)] * 4/6= 0.932
{p1, p3} = [log 5 + log 4] * 4/6 = 0.867
{p1, p4} = [log 5 + log 3] * 2/6 = 0.392
{p1, p5} = [log 5 + log 1] * 1/6 = 0.116
{p1, p6} = [log 5 + log 4] * 3/6 = 0.651
{p2, p3}= [log 5 + log 4] * 3/6 = 0.651
{p2, p4}= [log 5 + log 3] * 3/6 = 0.588
{p2, p5}= [log 5 + log 1] * 1/6 = 0.116
{p2, p6}= [log 5 + log 4] * 3/6 = 0.651
{p3, p4}= [log 4 + log 3] * 1/6 = 0.179
{p3, p5}= [log 4 + log 1] * 1/6 = 0.100
{p3, p6}= [log 4 + log 4] * 3/6 = 0.602
{p4, p5}= [log 3 + log 1] * 1/6 = 0.079
{p4, p6}= [log 3 + log 4] * 2/6 = 0.3597
{p5, p6}= [log 1 + log 4] * 1/6 = 0.100

IR value will be round off and the maximum value is 1.

So the length is 1 for next IR value. Item sets which are having IR value 1 are given below.

{p1, p2}, {p1, p3},{p1, p6},{p2, p3},{p2, p4},{p2, p6},{p3, p6}
The IR value for length 1 is calculated as follows:

{p1, p2}=> p3= [log (2 * 4) + log (1 * 4)] * 3/6= 0.753
{p1, p2}=> p4= [log 8 + log 3] * 2/6= 0.460
{p1, p2}=> p5= [log 8 + log 1] * 1/6= 0.151
{p1, p2}=> p6= [log 8 + log 4] * 2/6= 0.502
{p1, p3}=> p4= [log 8 + log 3] * 1/6= 0.230
{p1, p3}=> p5= [log 8 + log 1] * 1/6= 0.151
{p1, p3}=> p6= [log 8 + log 4] * 3/6= 0.753
{p1, p6}=> p4= [log 6 + log 3] * 1/6= 0.209
{p1, p6}=> p5= [log 6 + log 1] * 1/6= 0.129
{p2, p3}=> p4= [log 6 + log 3] * 1/6= 0.290
{p2, p3}=> p5= [log 6 + log 1] * 1/6= 0.129
{p2, p3}=> p6= [log 6 + log 4] * 2/6= 0.460
{p2, p4}=> p5= [log 6 + log 1] * 1/6= 0.129
{p2, p4}=> p6= [log 6 + log 4] * 2/6= 0.460
{p2, p6}=> p5= [log 6 + log 1] * 1/6= 0.129
{p3, p6}=> p4= [log 6 + log 3] * 1/6= 0.290
{p3, p6}=> p5= [log 6 + log 1] * 1/6= 0.129

Generation of candidate item set:
From the above IR values top most value will be taken for forming the candidate item set. Here we are taking the values 0.753, 0.502, 0.460 and 0.290. The formed candidate set is given below:

{(p1, p2, p3), (p1, p2, p4), (p1, p2, p6), (p1, p3, p6), (p2, p3, p4), (p2, p3, p6), (p2, p4, p6), (p3, p6, p4)}

Support value Calculation:

Support (u=>v) =Num_trans(u,v)/Total_trans

Support (p1, p2, p3) = 3/6 = 0.5
Support (p1, p2, p4) = 2/6 =0.33
Support (p1, p2, p6) = 2/6 =0.33
Support (p1, p3, p6) = 3/6 = 0.5
Support (p2, p3, p4) = 1/6 =0.16
Support (p2, p3, p6) = 2/6 =0.33
Support (p2, p4, p6) = 2/6 =0.33
Support (p3, p6, p4) = 1/6 =0.16

Frequent item set creation:
Here we are taking the threshold value as 0.3. The item sets which are having support value less than 0.3 are removed from the candidate set and formed the frequent item set. Here {p2, p3, p4} and {p3, p6, p4} are removed and the frequent item set is given below:
.
{(p1, p2, p3), (p1, p2, p4), (p1, p2, p6), (p1, p3, p6), (p2, p3, p6), (p2, p4, p6)}

Reduced ontology tree:
From the frequent item set, we can reduce the ontology tree. The reduced ontology tree is shown in figure 11.

DOI: 10.18535/ijecs/v5i10.24

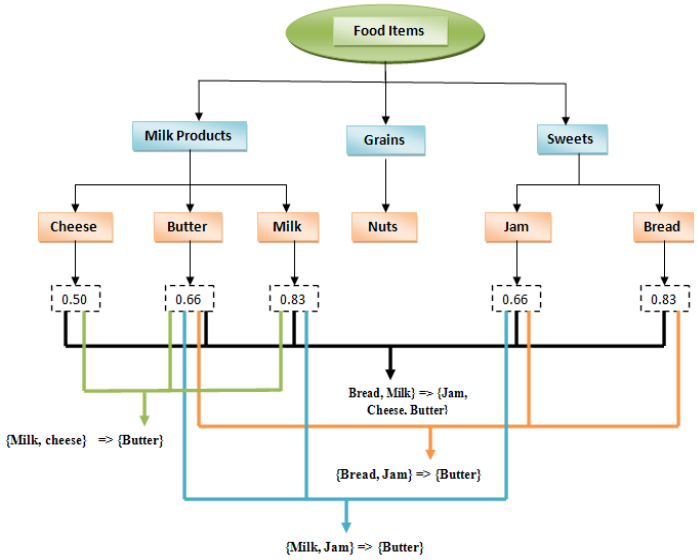


Fig.11 Reduced Ontology Tree

Mining Association rules:
From the reduced ontology tree, possibility of buying the following item sets is high.

{(p1, p2, p3), (p1, p2, p4), (p1, p2, p6), (p1, p3, p6), (p2, p3, p6), (p2, p4, p6)}

Rules for the item sets:

{Milk, cheese} => {Butter}
{Bread, Jam} => {Butter}
{Milk, Jam} => {Butter}
{Bread, Milk} => {Jam}
{Bread, Milk} => {Cheese}
{Bread, Milk} => {Butter}

Bread, Milk} => {Jam, Cheese, Butter}

Ontology Tree Creation:
The process of building an ontological tree is a time consuming process with the time required directly proportional to the number of the transaction i.e. to the size of the database. In the current experiment, we have chosen diverse customers transactions with a total of 30 items. If the number of the iteration is high, then the time requirement for creating an ontological tree will also be high. The size of our database is chosen as 80, thereby requiring 207.29 seconds to create the tree.

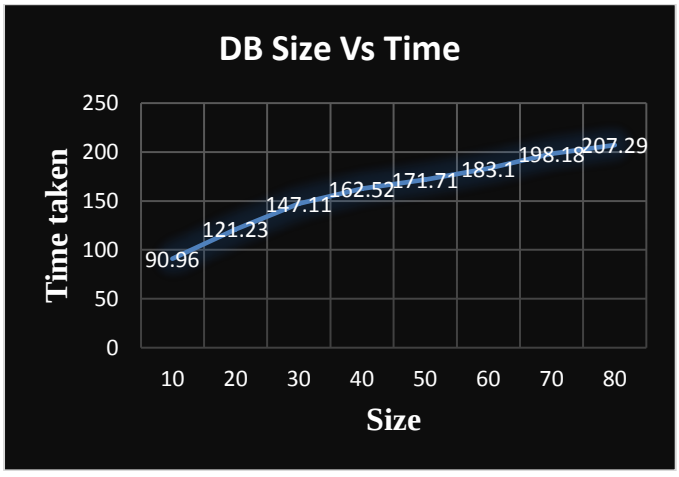
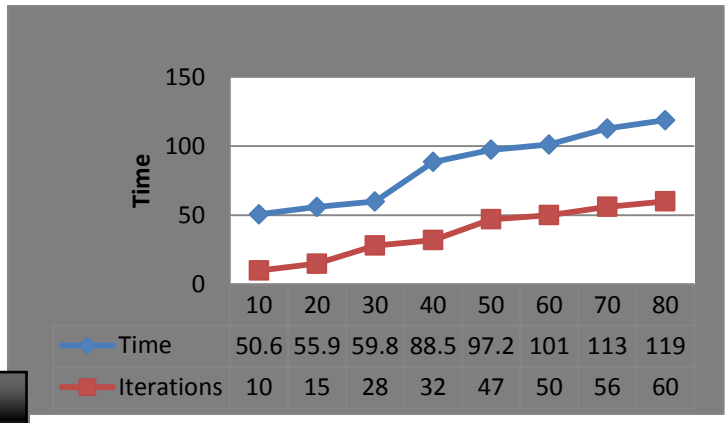


Fig.12 Graph of Ontology Tree Creation

Frequent item Set Generation:
The IR and the support value totally decide the frequency item set generation process. In figure 13, the blue line depicts the time consumed to discover the frequent item sets with different transactional data base. The number of the iterations taken for each transactional database is depicted by the red line. There are a minimum of at least 1-30 items in each of the transaction record of food mart 2015. Following the calculation of the IR value as 3.4712, the candidate item set is generated with a length as 3. Then, the support value is calculated (e.g.: support value of {1,6,4}=> {5,2,9}) . Then, the candidate set having the least value is eliminated. Hence, following each iteration, the candidate set gets reduced and one is left the frequent item set.



g.13 Graph of Frequent time set generation

Mining Association Rules:
The computational results differ for each iteration. Hence, totally, 15 iterations were conducted for a transactional database of size 20. The efficient rules were found in the 15th iteration. Also, the results proved that the purchasing of the items were very high in the final results. The final results are also a means to understand the pattern of the buyers to purchase a particular product. The graph below depicts the total time taken for mining the optimized association rule.

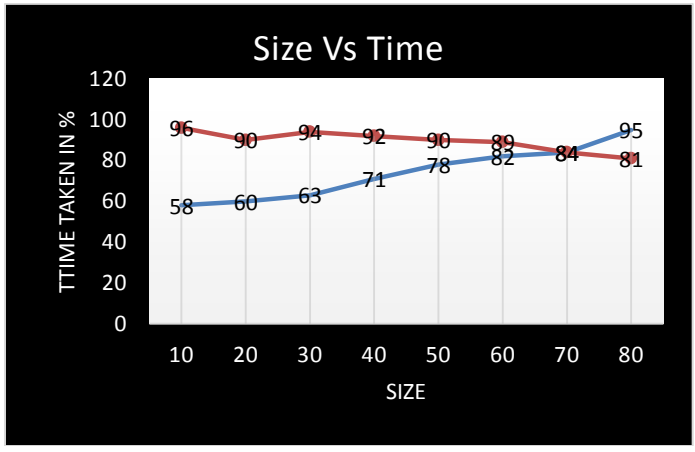


Fig.14 Graph of Mining Association Rules

DOI: 10.18535/ijecs/v5i10.24

ALGORITHM NAME	DATA SUPPORT	ACCURACY	APPLICATIONS
AIS	Less	Very Less	Used for small problems
SETM	Less	Less	Not frequently used
APRIORI	Limited	Less	Best for closed item sets
FP GROWTH	Very large	More accuracy than Apriori	Used for large applications
PSO	Very large	More accuracy than FP growth &Apriori	Used for large applications which contain free item sets, closed item sets, etc.
ARMO	Very large	More accuracy than FP-growth and PSO.	Used for large applications which contain different types of item sets

Comparison with Other Algorithms:

The comparison between the ARMO algorithm and the other association rule mining algorithms is as depicted in the table below. Data support, accuracy and applications of the individual algorithm form the basis of the division. The table enumerates a clear analysis on the basis of which a user can form conclusion on as to which algorithm to be employed for the best result in any given case. The accuracy of the ARMO algorithm is higher as compared to the other algorithms although it requires more time.

Table.2 Comparison of mining Algorithms

VI CONCLUSION

The current paper aids the user in executing the post-processing step of association rule mining in an efficient manner. The ontology that is created based on the user knowledge is connected to the data. A study for the mining of the association rules for the large database was done. Based on the ontology tree, a new technique to find the optimized association rules was proposed that is named as ARMO (Association rule mining with ontology). The experiment results proved that ARMO is more efficient that the existing methods. The results are more accurate in terms of number of queries. The number of rules was reduced by the application of the new technique to large databases in the post-processing step of association rule mining. The quality of the rules thus obtained was approved by expert throughout the interactive process.

VII REFERENCE

[1] R.Agarwal, T.Imielinski, and A.Swami, (1993), “Mining Association Rules Between Sets of Items in Large Databases”, In the proceedings of the ACM SIGMOD International Conference on Management of Data. pp: 207-216.

[2] R.Agarwal and Srikant.R, (1994), “Fast Algorithms for Mining Association Rules”, In the proceedings of 20th International Conference on Very Large Data Bases. pp: 487-499.

[3] V venkateswararao and E rambabu “Association Rule Mining UsingFPTree” IEEE-International Conference On Advances In Engineering,Science And Management (ICAESM -2012) March 30, 31, 2012.

[4] R Agarwal and A swaami “Mining an association rules between set ofitems in large transactional database”, Proceedings of the 1993 ACMSIGMOD Conference Washington DC, USA.

[5] R.J. Kuoa,∗, C.M. Chao b, Y.T. Chiuc "Application of particle swarm optimization to association rule mining", Elsevier Applied Soft Computing 11 (2011) 326–336.

[6] Irina Tudor, "Association Rule Mining as Data Mining technique" BULETTNU Luniversi Noll2008,page 49-56.

[7] SauravMallik, AnirbanMukhopadhyay, "RANWAR: Rank-Based Weighted Association Rule
Mining from Gene Expression and Methylation Data", IEEE Transactions on NanoBioscience.

[8]AzadehSoltani and M.-R.Akbarzadeh-T.,"Confabulation-Inspired Association Rule Miningfor Rare and Frequent Itemsets" IEEE Transactions On Neural Networks And Learning Systems.

[9]Tarinder Singh; ManojSethi, "Sandwich-Apriori: A combine approach of Apriori and Reverse-Apriori",2015 Annual IEEE India Conference (INDICON) Year: 2015 Pages: 1 - 4.

[10]RakeshAgrawal and amakrishnanSrikant, "Fast Algorithms for Mining Association Rules", IBM Almaden Research Center.

[11]F. Bodon. A Fast AprioriImplementation.Proc.1st IEEE ICDM Workshop on Frequent Item Set Mining Implementations (FIMI 2003, Melbourne, FL).CEUR Workshop Proceedings 90, Aachen, Germany 2003. http://www.ceur-ws.org/Vol-90/

[12]C. Borgelt. Efficient Implementations of Apriori and Eclat.Proc. 1st IEEE ICDM Workshop on Frequent Item Set Mining Implementations (FIMI 2003, Melbourne, FL).CEUR Workshop Proceedings 90, Aachen, Germany 2003.http://www.ceur-ws.org/Vol-90/

[13]Li Min; Wang Chunyan; Yan Yuguang,"The Research of FP-Growth Method Based on Apriori Algorithm in MDSS", Digital Manufacturing and Automation (ICDMA), 2010 International Conference on Year: 2010, Volume: 2 Pages: 770 - 773.

[14] D. J Allocco, I. S Kohane and A. J Butte: "Quantifying therelationship between co-expression, co-regulation and gene function",BMC Bioinformatics, 5:18 2004.

[15] M. B Eisen, P. T Spellman, P. 0 Brown, D. Botstein, "Cluster analysis and display of genome-wide expression patterns", Proc Nat AcadSci USA, 95: 14863-14868, 1998.