International Journal of Advanced Research in IT and Engineering

ISSN: 2278-6244

USING ASSOCIATION RULES IN DATA MINING

Shahin Samimi*

Fariba Rastegar*

Abstract: *Mining associative rules in an investigative issue related to data analysis and also related to discovery of investing and important relationship between information items exists in large data base and transaction stores, that is allocated many investigative efforts, recently. The main applications of them are the famous subject of basket purchase, clustering and classification. The total structure of associative rules between X and Y. X and Y are called prior and rules, differently criterions and given that base on them, it would be possible to choose the good rules among the wide collection of possible rules. The most famous and the most applicable criterions are: Minimum support rate and Minimum assurance rank. Supporting a collection of items like X include the ratio of number of transactions all of the exits items in X on the number of the total transaction.*

Keywords: *Associative rules, data mining, extraction of concepts, classification, transaction*

*Department of Computer Engineering, Behbahan Branch, Islamic Azad University, Behbahan, Iran

1- INTRODUCTION

Association rules Show the relations and mutual dependence between large set of data items. To find such rules can be considered in different domains and have different applications.

For example, the discovery of association's relations among the huge volumes of business transaction can be used in Fraud diagnosis in the medical field and data mining about information of web usage by users and Personalization.

A common example in relation to the discovery of association rules is" analysis Shopping Cart ". In this process due to the different items that customers have been purchasing habits and purchase behavior of customers is analyzed. For example, it become clear how likely is it that customers who have come to buy bread in the store, buy milk also. The goal of this process is to find automatically rules such as: "60 % of people who buy bread, will buy milk also".

2- BASIC CONCEPTS

In Preservation private Privacy Issues, in Data Mining Some Basic Concepts Is borrowed from the mining association rules. So, before explaining about Preservation private Privacy in Data Mining we needed to define, searching the association rules. The issue of searching the association rules was introduced by Agrawal. Suppose that $I=\{i_1,i_2,...,i_m\}$ is a collection of elements. Database $D=\{T_1,T_2,...,T_n\}$ is a set of transactions. Each transaction is a subset of *I*. Association rules are like A→B in which A and B are element sets and occur frequently in Transactional database [1]. They are the subset of I and $A\cap B=0$. Supporting an association rule express the Percentage of transactions which are included in $A\cup B$. Equation (1) express the Formula of Calculating Support of An association rule:

(1) $$Support(A\rightarrow B)=\frac{|A\cup B|}{N}$$

Association rule Confidence express the percent of transactions which if they are consisted of A they should be included B also. Equation (2) express how to calculate the Law Confidence:

(2) $$confidence(A\rightarrow B)=\frac{|A\cup B|}{|A|}$$

An association rule is interesting, if support is larger or equal to the minimum Support and its reliability is larger or equal to minimum reliability.

The wrapping algorithm of Recurring Set of Ingredients prevents Search for Recurring Set of Ingredients. Problem Could Be expressed as follows:

"A Transaction Database Called D, Minimum Support Threshold and series of Recurring elements Sensitive, Set by the user was given [2]. How we can Immunize the database Somehow in which the Sensitive Set of Recurring elements does not explore the, But set of Insensitive Recurring elements be explored? "

For example, Assumed The database is in Table 1 and the set of recurring sensitive elements {*ab*, *cd*, *ef*} should be hiding and Minimum Support threshold is equal to 2.

Table 1: A sample database with 10 transactions and 9 Elements

Number of transaction	elements
T_1	a,b,c,d,f,g,h
T_2	a,d,e,f
T_3	b,c,d,f,g,h
T_4	a,b,c,f,h
T_5	c,d,e,g,i
T_6	a,c,f,i
T_7	b,c,d,e,f,g
T_8	c,d,f,h,i
T_9	a,d,e,f,i
T_{10}	a,c,e,f,h

Algorithm SIF-IDF, to hide the collections of sensitive repetitive elements, at first select sensitive transactions, which means choosing transactions involving at least a set of sensitive elements ,and they can count SIF-IDF, then chooses the transactions that have maximum amount to change. See Table 2.

Table 2: SIF-IDF values for transactions

Number of transaction	elements	SIF-IDF
T_1	a,b,c,d,f,g,h	0.53
T_2	a,d,e,f	0.666
T_3	b,c,d,f,g,h	0.431
T_4	a,b,c,f,h	0.453
T_5	c,d,e,g,i	0.508
T_7	b,c,d,e,f,g	0.701
T_8	c,d,f,h,i	0.412
T_9	a,d,e,f,i	0.533
T_{10}	a,c,e,f,h	0.508

In this example, transaction T_7 have selected for Change. The elements which have maximum frequency among the collection of sensitive elements was choose to remove. $\{a=1, b=1, c=1, d=1, e=1, f=1\}$, this collection shows frequency of sensitive elements in the set of sensitive elements ,Since the frequency of all critical elements is equal, so an element is selected to remove from the transaction, then Support of repetitive sensitive elements is updated. The procedure is repeated as long as all sets of repetitive elements sensitive hided. As was the case, removing the element from the supportive transaction did not reduce any of the sensitive elements. So, SIF-IDF algorithm modified. A sensitive element of the transaction is removed which both have the most abundant element in the collection of sensitive elements and be a member of at least one set of sensitive elements. A set which the selected transactions is included in it. Due to this solution element is removed from the transaction [3].

3- CLASSIFICATION USING ASSOCIATION RULES

The finding forum rules, in the general case, we do not follow a targeted search and looking forward to find all the relationships and dependencies. Whereas in categorizing the goal is clear. Simply and with a little changes we can convert the problem of find the forum rule to the problem of find Classification rules. And by using the resulted rules build a classifier based on association rules.

Our problem is to find forum rules in $A \rightarrow C_i$. In which, A is a set of possible elements and classes. Terms “Support” and “confidence”, for these rules, are introduced like before. And with the previous definitions, we are looking for Strong regulations [4]. For classification, firstly, we should find the strong classification Rules. Then, for classification we should find all strong classification forum rules, and then by using (a subset of) these classification rules build the manufacturer. The first part of the work, rules production, is performable by making small changes in Apriority (figure 1).

Firstly, the algorithm passing through site, so that it can find the one part important collection items rules, the ones which have a pen on the left side of the rule (1-ruleItems). Regarding these items, simply we can made laws and then modify them. (With the same criteria C4.5). In next passing, similar to Apriority, k at each stage made by using ruleitem of previous stage of candidate set of k-ruleitems and then their support was calculated. And

important Ruleitem of K which is part of FK is obtained. Then due to this rules, laws made at each step (and possibly modified).

```
1   F1 = {large 1-ruleitems};
2   CAR1 = genRules(F1);
3   prCAR1 = pruneRules(CAR1);
4   for (k = 2; Fk-1 ≠ Ø; k++) do
5       Cd = candidateGen(Fk-1);
6       for each data case d ∈ D do
7           Cd = ruleSubset(Ck , d);
8           for each candidate c ∈ Cd do
9               c.condsupCount ++;
10              if d.class = c.class then c.rulesupcount++;
11          end
12      end
13      Fk = { c ∈ Ck | c.rulesupcount • nimsup};
14      CARk = genRules(Fk);
15      prCARk = pruneRules(CARk);
16  end
17  CARs = Uk CARk;
18  prCARs = Uk prCARk;
```

Figure1: Algorithm of finding of classification rules

The next step is built a classifier by using these rules. Surely, the best case is considering all possible subsets of rules and choosing the best ones according to their performance. As, this has a very high implementation costs a search are used to find Rules in which a fully ordered relationship considers on the law:

Having two rules r_j, r_i relationship $r_j \prec r_i$ (r_j is prior to r_i) in which:

- Confidence of law r_i is more than r_j.
- Having equal Confidence, but the support of law r_i is more than r_j.
- Having equal Confidence and Support, but r_i is produced earlier then r_j

The resulting classifier arranged to form complex $< r_1, r_2, .., r_n, deault >$ in which we have $r_j \prec r_i$ for each $i<j$. and the default is a class which, attributed to the samples in default. In this series the first rule which can classify an example, specifies its class. To produce this set a simple algorithm is presented below (Figure 2).

This algorithm do its work based on coated samples by the law. Starting from The first law, eliminates Examples which covered, and proceed to the next law#if At least classified one sample repeats it and otherwise the law eliminates [5].

Work continues until either all the samples are removed or reach the end of the law (in which case default adds the law).

```
R = sort(R)     /*according the precedence*/
for each rule r ∈ R in sequence do
        if there are still cases in D AND r classification at least one case correctly then
                delete all training examples cover by r from D;
                add r to the classifier
        end
end
add the majority class as the default class to the classifier
```

Figure 2: Algorithm of rules selection

Implementing this method has high costs and method is more efficient which almost requires two passes on site. Experiments shows, on average, this algorithm has better performance than systems such as C4.5.

In other approach different thresholds considers Support for different classes, so it can face to the uneven distribution of samples in classes, using a MinSup determines it generally. And the threshold between different classes divided in terms of frequency of occurrence of each class in sample. The aim of this work is to find appropriate laws for low occurrence classes. (In the previous case, we had MinSup, these classes did not have minimum support) and still prevented produce a large number of rules for class with high occurrence.

4. CONCLUSION

With according to the increasing use of large databases and large storage transaction, recently a lot of attention to produce an efficient method for mining association rules has been attracted. Most of the existing methods search all the items in the data, in its first phase, all frequent items (simple or complex), that it requires repeated reading of data from disk. Many methods have been proposed recently to try to at least some degree of support and degree of occurrence of items can be calculated directly without scrolling data. But less attention to the question of how to optimize the method for counting the frequency of occurrence of items when the components count, no way achieve.

REFERENCES

[1] Verma, K., Vyas, O.P., Vyas, R.,"*Temporal Approach to Association Rule Mining Using T-Tree and P-Tree*", Lecture Notes in Computer Science, Volume 3587, Jul 2005, Pages 651–659.

[2] Cendrowska, J. (1997). "*PRISM: An algorithm for inducing modular rules*". International Journal of Man-Machine Studies. Vol.27, No.4, pp.349-370.

[3] W.Li, J.Han, and J.Pei, (2001). "*CMAR: Accurate and efficient classification based on multiple-class association rule*". In ICDM'01, pp.369-376, San Jose, CA.

[4] Agrawal, R. , Imielinski, T. , and Swami, A. N. 1993. "*Mining asso ciation rules b etween sets of items in large databases*". In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, P.Buneman and S.Jajodia, Eds. Washington, D.C., 207-216.

[5] Fayyad, W., Piatetsky - Shapiro, G., Smyth, P. From data mining to knowledge discovery: "*An overview, In: Advances in Knowledge Discovery and Data Mining*", AAAI/MIT Press, Cambridge/USA, pp. 1 – 3, 1996