ISSN No. 0976-5697

Volume 4, No. 11, Nov-Dec 2013

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A Study of Association Rule Mining Algorithms

Ila Chandrakar
Department of CSE, BMSIT
Bangalore, India
ilaprithvi@gmail.com

Mari Kirthima
Department of CSE, BMSIT
Bangalore, India
krithi.a@bmsit.in

***Abstract*:** Association rule mining is a technique in data mining for finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transactional databases, relational databases, and other information repositories.Association rules are created by analysing data for frequent if/then patterns and using the criteria *support* and *confidence* to identify the most important relationships. *Support* is an indication of how frequently the items appear in the database. *Confidence* indicates the number of times the if/then statements have been found to be true. In this paper, we study different approaches for mining these association rules from relational databases for different areas of application.

Keywords: Data mining, association rules mining, database

I. INTRODUCTION

Data mining: Data mining is the computational process of finding patterns in large data sets using methods like artificial intelligence, machine learning and statistics. The main aim of the data mining process is to extract information from a database and transform it into a simpler form which can be further used.It involves the raw data analysis step, database and data management aspects, data pre-processing, model and inference considerations, metrics, complexity considerations, post-processing of discovered structures, visualization, and online updating. It involves many techniques like clustering, classification, association rule mining etc.

Association rule miming: Association rule mining is a popular and well researched method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using different measures of interestingness. Association rules for discovering regularities between products in large-scale transaction data recorded by point-of-sale (POS) systems in supermarkets. For example, the rule*{soap, shampoo} => {conditioner}*found in the sales data of a supermarket would indicate that if a customer buys soap and shampoo together, he or she is likely to also buy conditioner. These types of information can be used as the basis for taking decisions about marketing strategies such asapplying offers in related items, promotional pricing or placing these products in nearby shelves. Like for market basket analysis in supermarket, association rules can also be used in many application areas including Continuous production, Web usage mining, intrusion detection and bioinformatics.

II. PROBLEM DEFINITION

The problem of association rule mining is defined as: Let I= {i1,i2, ….in)be a set of n binary attributes called *items*. Let DD={t1, t2,…tn} be a set of transactions called the *database*. Each transaction in D has a unique transaction ID and contains a subset of the items in I. A *rule* is defined as an implication of the form X=>Y where X, Y⊆I and X ⋂ Y=∅ .The sets of items (for short *item sets*) X and Y are called *antecedent* (left-hand-side or LHS) and *consequent* (right-hand-side or RHS) of the rule respectively[6].

To illustrate the concepts, we use a small example from the supermarket domain. The set of items is *I={bread, jam, butter, soap }*and a small database(Table 1) containing the items (1 representsthat item is present and 0 represents that item is not present in a transaction). An example rule for the supermarket could be *{bread, jam} => {butter}* meaning that if bread and jam are bought, customers also buy butter.

Table 1: Sample Database

T	Bread	Jam	Butter	Soap
T1	1	1	0	0
T2	1	1	1	0
T3	1	0	1	1
T4	0	1	1	0
T5	1	1	0	0

To select interesting rules from the set of all possible rules, constraints on various measures of significance and interest can be used. The best-known constraints are minimum thresholds on support and confidence.

- The *support* supp(x) of an item set X is defined as the proportion of transactions in the data set which contain the item set. In the example database, the item set{bread, jam,butter} has a support of 1/5=0.2. Since it occurs in 20% of all transactions (1 out of 5 transactions).

- The *confidence* of a rule is defined

Conf(X=>Y)=sup(X∪Y)/supp. For example, the rule *{bread,jam}=>{ butter}*has a confidence of 0.2/0.4 = 0.5in the database, which means that for 50% of the transactions

containing bread and jam the rule is correct (50% of the times a customer buys bread and jam, butter is bought as well). Here supp(X∪Y) means "*support for occurrences of transactions where **X and Y both appear***", not "*support for occurrences of transactions where **either X or Y appears***", the latter interpretation arising because set union is equivalent to logical disjunction. The argument of supp() is a set of preconditions, and thus becomes more restrictive as it grows Confidence can be interpreted as an estimate of the probability P(Y|X), the probability of finding the RHS of the rule in transactions under the condition that these transactions also contain the LHS[7].

III. RECENT ALGORITHMS

A. AprioriAlgorithm[1]:

Apriori is a classic algorithm for mining association rules. Apriori is designed to operate on databases which contain transactions like supermarket database which contains sales details of items for different customers.Apriori uses breadth-first search and a Hash tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k-1. Then it extracts the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.The pseudo code for the algorithm is given below for a transaction database T and a support threshold of €. $C_k$is the candidate set for level k. Generate () algorithm is assumed to generate the candidate sets from the large item sets of the preceding level. count[c] accesses a field of the data structure that represents candidate set C, which is initially assumed to be zero.

Algorithm[5]:

$\mathrm{Apriori}(T, \epsilon)$
$L_1 \leftarrow \{\text{large } 1-\text{itemsets}\}$
$k \leftarrow 2$
while $L_{k-1} \neq emptyset$
$C_k \leftarrow \{a \cup \{b\} | a \in L_{k-1} \wedge b \in \bigcup L_{k-1} \wedge b \notin a\}$
for transactions $t \in T$
$C_t \leftarrow \{c | c \in C_k \wedge c \subseteq t\}$
for candidates $c \in C_t$
$count[c] \leftarrow count[c] + 1$
$L_k \leftarrow \{c | c \in C_k \wedge count[c] \geq \epsilon\}$
$k \leftarrow k + 1$
return $\bigcup_k L_k$

The problem with this algorithm is the number of database passes which is equal to the maximum length of frequent item set.

B. Systematic algorithm[2]:

Systematic algorithm is a variation of aprori algorithm, minimum support threshold values will not be specified by the user to find the frequent patterns but it is generated by the system itself which is an improvement over Apriori and other algorithms. In this approach, the user is well aware of entire information which helps him to take correct decisions. The algorithm called timing algorithm is also introduced along with the systematic algorithm, which assigns a unique value to each record of the transactional database statically. This technique requires only one database scan instead of multiple scans required in Apriori algorithm which results in better performance by reducing time taken to mine the rules and provides facility of using highest support values in the systematic table. This can be done by constructing systematic table for each data set in the database.

The systematic tables for every item sets of the datasets can be calculated by the following conditions -

- Supp(A---> B) = sup(A) + supp(B) + supp (AUB)
- Supp (A--->¬B) =supp (A) - supp (AUB)
- Supp (¬A---> B) =supp (B) - supp (A UB)
- Supp (¬A ---> ¬B) =1 - supp (A) - supp (B) + supp(A UB)

C. Genetic Algorithm[3]:

This algorithm is a stochastic search algorithm modelled on the process of natural selection, which underlines biological evolution. Genetic Algorithm has been successfully applied in many search, optimization, and machine learning problems. Genetic Algorithm process works in an iteration manner by generating new populationsof strings from old ones. Every string is the encoded binary, real etc., version of a candidate solution. Thegenetic algorithm based method for finding frequent itemsets repeatedly transforms the population by executing the following steps:

(1) Fitness Evaluation: The fitness (i.e., an objective function) is calculated for each individual.

(2) Selection: Individuals are chosen from the current population as parents to be involved in recombination.

(3)Recombination: New individuals (called offspring) are produced from the parents by applying genetic operators such
as crossover and mutation.

(4) Replacement: Some of the offspring are replaced with some individuals (usually with their parents). One cycle of transforming a population is called a generation. In each generation, a fraction of the population is replaced with offspring and its proportion to the entire population is called the generation gap (between 0 and 1).

In general the main motivation for usingGenetic Algorithms in the discovery of high-level prediction rules is that they perform a global search and cope better with attribute interaction than the greedy rule induction algorithms often usedin data mining.

D. RUPF (Rapid Update in Frequent Pattern)[4]:

Temporal data mining is a technique which deals with changing data. Actually, temporal databases are continually appended or updated so that the discovered rules need to be updated.This algorithm is used to mine temporal association rules from temporal database so it keeps track of added and removed data items in the database.

Algorithm:

Initialize: K: = 1, C1 = all the 1- item sets;
Read the database to count the support of C1 to
Determine L1.
L1:= {frequent 1- item sets};
K: =2; //k represents the pass number//
While (Lk-1 ≠ Φ) do
Begin
Ck: = gen_candidate_itemsets with the given Lk-1

Prune (Ck)
For all candidates in Ck do
Count the number of transactions that are common in each item Є Ck
Lk: = All candidates in Ck with minimum Support;
K: = K + 1;
End

IV. CONCLUSION

In this paper, we have done survey of various recent algorithms for association rule mining used for different areas of applications. The Apriori algorithm is used for transaction database like supermarket database, systematic algorithm is an improvement over Apriori algorithm by reducing time of execution with only one database scan, Genetic algorithm is used for producing genes from parents and RUPF algorithm is used to mine association rules from a temporal kind of database so that correct association rules can be obtained even from updated database.

V. REFERENCES

[1] GhorechaVimal, "Comparative Evaluation of Association Rule Mining Algorithmswith Frequent Item Sets", IOSR Journal of Computer Engineering (IOSR-JCE), Volume 9, Issue 5 (Mar. - Apr. 2013), PP 08-14

[2] R. Agrawal. T. Imielinski. and A Swami. "Mining Association Rules between Sets of Itemsin Large Databases," Proc. 1993 ACM SIGMOD Int'I Conf. Management of Data (SIGMOD '93), pp. 207-216, 1993.

[3] ArvindJaiswal, GauravDubey, Identifying Best Association Rules and Their Optimization Using Genetic Algorithm, International Journal Emerging Science and Engineering (IJESE)ISSN:2319–6378, Volume-1, Issue-7, May 2013[

[4] AbhayMundra, PoonamTomar, Deepak Kulhare, " Rapid Update in Frequent Pattern form Large Dynamic Database to Increase Scalability", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-6, January 2013

[5] Ms Shweta, Dr.GargKanwal, "Mining Efficient Association Rules Through AprioriAlgorithm Using Attributes and Comparative Analysis ofVarious Association Rule Algorithms" ,International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 6, June 2013

[6] SanjeevRao, Prianka Gupta, "Implimenting Improved Algorithm Over APRIORI Data Mining Association Rule Algorithm", In: proceeding of IJCST, ISSN 0876-8491, VOL.3, Issue 1, Jan-March 2012.

[7] Jogi.Suresh, T.Ramanjaneyulu, "Mining Frequent Itemsets Using Apriori Algorithm", In: Proceeding of International Journal of Computer Trends and Technology, ISSN 2231-2803, Vol. 4, Issue 4, April 2013.

