[Volume 1 issue 10 Nov 2013]
Page No.270-272 ISSN :2320-7167

INTERNATIONAL JOURNAL OF MATHEMATICS AND COMPUTER RESEARCH

A Survey on Association Rule Mining Algorithms

Ila Chandrakar[1], *A. Mari Kirthima*[2]
Assistant Professor Dept. of CSE BMSIT Bangalore, India
ilaprithvi@gmail.com
krithi.a@bmsit.in

***Abstract*: Association rule mining is a technique in data mining which is used to mine different association rules from the given database. Association rules are if/then statements that help uncover relationships between seemingly unrelated data in a relational database or other information repository. In this paper, we mention the different approaches for mining association rules from relational databases and their usage in different areas.**

Keywords: Data mining, association rules mining, database

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.

I. 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.

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 setX 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.

II. CURRENT ALGORITHMS

A. AprioriAlgorithm:

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:

```
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. Aproiri TID algorithm:

The AprioriTid algorithm also determines the candidate item sets before the pass begins. The interesting feature of this algorithm is that the database D is not used for counting support after the first pass.

Algorithm

```
L1 = {large 1-itemsets};
D1 = database D;
for ( k = 2; Lk-1 != Φ; k++ ) do begin
Ck = apriori-gen (Lk-1 );
Dk = Φ;
for all entries t€2 Dk-1 do begin
Ct = { c€Ck | (c -c[k]) €t:set-of-itemsets ^ (c -c[k-1]) €t.set-of-itemsetsg;
for all candidates c €Ct do
c.count++;
if (Ct != Φ) then Dk += <t.TID,Ct>;
end
Lk = {c €Ck | c.count>= minsup}
End
Answer = UkLk;
```

In earlier passes, Apriori takes less time but in later passes,AprioriTID takes less time than Apriori.

C. FP algorithm:

This algorithm allows frequent itemset discovery withoutcandidateitemset generation. It requires database scan twice only.

Algorithm:

1. Build FP-Tree
   - Scan database to discover frequent 1-itemsets
   - Set the order of items in the transaction as the order of decreasing support, e.g., A, B, C, D (s(A) > s(B) > ...)
   - Scan database again to build a compact representation of transactions in form of FP-Tree
2. Discover frequent itemsets using FP-Tree
   - Recursively find frequent itemsets with common suffix, ending with items having lower support firstE.g., finding itemsets ending with D, C, B, A.

D. Genetic Algorithm:

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. Standard Genetic Algorithm apply genetic operators such selection, crossover and mutation on an initiallyrandom population in order to compute a whole generation of new strings.*Selection* deals with the probabilistic survival of the fittest, in that fit chromosome arechosen to survive, where fitness is a comparable measure of how well a chromosome solves the problem at hand.*Crossover* takes individual chromosomes from Parents,combines them to form new ones.*Mutation* alters the new solutions so as to add stochastic in the search for better solutions. 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.

Algorithm:

1. An initial population is created. A Population is a group of individuals (Chromosomes) and represents a candidate solution. A Chromosome is a string of genes.

2. Select chromosomes with higher fitness.
3. Crossover between the selected chromosomes to produce new offspring with better higher fitness.
4. Mutate the new chromosomes if needed.
5. Terminate when an optimum solution is found.
This generational process is repeated until a termination condition has been reached. Common terminating conditions are:

- •A solution is found that satisfies minimum criteria
- •Fixed number of generations reached
- •Allocated budget (computation time/money) reached.
- •The highest ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results
- •Manual inspection
- •Combinations of the above

E. Partition algorithm:

Partition algorithm accomplishes association rule mining in two scans of the database. In one scan, it generates a set which is a superset of all large item sets by scanning the database only once. This superset may contain false positives but it will not have false negatives. In second scan, counters is generated for each of item set and the counters actual support is measured in only one database scan. The algorithm executes in two phases in which, first phase of Partition algorithm logically divides the database into a number of non-overlapping partitions andin second phase, it generates the actual support for these item sets and the large item sets are identified. The partition sizes should be such that, each partition can be accommodated in the main memory so that the partitions are read only once in each phase[2].

III. CONCLUSION

In this paper, we have done survey of various current algorithms for association rule mining used for different applications. The Apriori algorithm is used for transaction database like supermarket database, AprioriTID is an improvement over Apriori algorithm by reducing time of execution, FP algorithm is improvement over Apriori and AprioriTID by reducing the number of database scans, Genetic algorithm is used for producing genes from parents and partition algorithm is used to improve performance by portioning the database into multiple parts.

IV. FUTURE WORK

As a future work, some new algorithms can be developed which requires only one time database scan, which can be used for multiple databases and which can be scalable with increasing size database.

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] KanhaiyaLal,N.C.Mahanti, “Mining Association Rules in Large Database by Implementing Pipelining Technique in Partition Algorithm”, International Journal of Computer Applications,2(4):33–39, June 2010.
[3] 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
[4]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.
[5]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.
[6]Jaiswal Arvind, Dubey Gaurav, “Identifying Best Association Rules and Their Optimization Using Genetic Algorithm”, International Journal of Emerging Science and Engineering (IJESE)ISSN: 2319–6378, Volume-1, Issue-7, May 2013.