Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 11, Number 6 (2015), pp. 5233-5240

http://www.ripublication.com

A Survey on Algorithms in Association rule mining

[*1] S.Irin Sherly
Faculty of Information Technology,
Panimalar Institute of Technology, Chennai, Tamil Nadu, India
irinkutty@gmail.com

[2] S.Hemamalini**
Faculty of Computer Science,
Panimalar Institute of Technology, Chennai, Tamil Nadu, India
shema2807@gmail.com

[*3] V.Mahalakshmi**
UG Scholar of Information Technology,
Panimalar Institute of Technology, Chennai, Tamil Nadu, India
maha.wyanee@gmail.com

[**4] T.Keerthika**
UG Scholar of Computer Science,
Panimalar Engineering College, Chennai, Tamil Nadu, India
keerthiprakash@yahoo.com

Abstract

Data is the important facts that is available everywhere. There are various repositories to store this data such as databases, data warehouses etc. This huge amount of data needs to be processed in order to get useful information. Data mining is the method to process huge data in order to get useful data. Data mining is one of various tool for examining data. It permits users to separate information from a wide range of measurements or points, classify it, and review the identified relationships. In Data mining, association rules are useful for examining and guessing the behavior of customer. They have critical impact in shopping basket data analysis, product clustering, catalog design and store layout. This paper presents a survey of the association rule mining algorithms. The advantages and limitations are discussed and concluded with an suggestion.

Keywords*:* Data mining, Association rule mining, AIS, SETM, Apriori, Aprioritid, Apriorihybrid, FP-Growth algorithm, Improved Apriori

1. INTRODUCTION

Data mining is the process of investigating data from dissimilar viewpoints and brief it into useful information It enables users to examine data from many different magnitudes, classify it, and review the identified relationships. Data mining is the process of finding associations among many of the fields in large relational databases. Data mining is principally employed by companies with a robust customer attention on retail, financial, communication, and advertising organizations. It allows these organizations to work out the connections among "inside" elements like value, item situating, or staff aptitudes, and "outer" variables like financial markers, rivalry and so on. And, it permits them to work out the impact on sales, customer satisfaction, and business profits. It additionally permits them to travel into define data to look at the transactional knowledge. With data processing, a merchant will utilize POS records of customer purchases to send targeted promotions based on the individual purchase history. By mining related knowledge from warrant cards, the distributor will develop merchandise and promotions to demand to specific client segments. For example, Blockbuster Entertainment mines its video rental history database to sponsor lease to man or woman consumers. American express can recommend the products to its cardholders based on gain knowledge of their month-to-month expenditures.

Data mining model takes information from various repositories like data warehouse, database, etc. It performs various operations like data improvement, integration, transformation and produces helpful information from that.

Association rules are if/then statements that helps to find the relationships between unrelated data in a relational database or other data store. A example of an association rule is "If a customer purchases twelve eggs, he is 80% prone to buy milk."

An Association rule has two sections, an antecedent (if) and a consequent (then). A antecedent is the item found in the data. A consequent is an item that is found in combination with the antecedent.

Association rules has two criteria namely support and confidence. Support indicates how frequently the items appear in the database. Confidence shows the number of times the if/then statements are found to be true.

Association rules are useful for examining and predicting the behavior of customer. They play an important role in applications like shopping basket data analysis, product clustering, catalog design and store design.

The following is an example database with 5 transactions and 5 items.

Transaction ID	Milk	Bread	Butter	Beer	Diapers
1	1	1	0	0	0
2	1	1	0	0	1
3	0	0	0	1	1
4	1	1	1	0	0
5	0	1	1	0	0

Association rule mining is defined as:
Let I={i1,12,…..,in} be a set of n binary attributes called *items*.
Let D={t1,t2,……..,tm} 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.
Every rule is composed by two different set of items, also known as *itemsets*, X an Y, where X is called *antecedent* and Y is called *consequent* .
To illustrate the concepts, we use a small example from the supermarket domain. The set of items is I={milk,bread,butter,beer,diapers} and in the table is shown a small database containing the items, where, in each entry, the value 1 means the presence of the item in the corresponding transaction, and the value 0 signify the absence of an item in a that transaction.
An example rule for the supermarket could be {bread,butter} → milk ,meaning that if butter and bread are bought then the customer will also buy milk.

1. **Support**

The support value of X with respect to T is defined as the proportion of transactions in the database which contains the item-set X.

2. **Confidence**

The *confidence* value with respect to a set of transactions T, is the proportion the transactions that contains X which also contains Y.

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

2. THEORETICAL SURVEY OF ALGORITHMS

This section presents a survey on Association rule mining algorithms.

A. AIS ALGORITHM

Agrawal *et al*.[1] introduced the AIS (Agrawal, Imielinski, Swami) algorithm for mining association rules. It focuses on improving the quality of databases along with the required functionality to process queries and ensuing association rules are generated.
In the AIS process, the actual database was scanned many times to get the frequent itemsets.
To make this algorithm more efficient, an estimation method was introduced to prune those item. Since all the candidate item sets and frequent item sets are assumed to be stored in the main memory, memory management is also proposed for AIS when memory is not enough.
In AIS algorithm, the frequent item sets were generated by scanning the databases

several times. The support count of each individual item was accumulated during the first pass over the database. Based on the minimal support count those items whose support count less than its minimum value gets eliminated from the list of item. Candidate 2-itemsets are generated by extending frequent 1-itemsets with other items in the transaction. During the second pass over the database, the support count of those candidate 2-itemsets are accumulated and checked against the support threshold. Similarly those candidate (k+1)-item sets are generated by extending frequent k-item sets with items in the same transaction. The candidate item sets generation and frequent item sets generation process iterate until any one of them becomes empty.

The drawback of this algorithm is too many candidate itemsets are generated. It requires more space and it wastes much effort. Also this algorithm requires too many passes over the whole database.

B. SETM ALGORITHM

In the SETM algorithm, candidate itemsets are generated on-the-fly as the database is scanned, but counted at the end of the pass. Then new candidate itemsets are generated in the same way as in AIS algorithm, but the transaction identifier TID of the generating transaction is saved with the candidate itemset in a sequential structure. It separates candidate generation process from counting. At the end of the pass, the support count of candidate itemsets is determined by aggregating the sequential structure.

The SETM algorithm has the same disadvantage of the AIS algorithm. Another disadvantage is that for each candidate itemset, there are as many entries as its support value.

C. APRIORI ALGORITHM

Apriori algorithm is used for frequent item set mining and association rule learning.The algorithm use a level-wise search, where k-itemsets are used to explore (k+1)-itemsets, to mine frequent itemsets from transactional database for Boolean association rules. In this algorithm, frequent subsets are extended one item at a time and this step is known as candidate generation process. Then groups of candidates are tested against the data. To count candidate item sets efficiently, Apriori uses breadth-first search method and a hash tree structure.

It identifies the frequent individual items in the database and extends them to larger and larger item sets as long as those item sets appear sufficiently often in the database. Apriori algorithm determines frequent item sets that can be used to determine association rules which highlight general trends in the database.

There are two drawbacks of the Apriori algorithm. First is the complex candidate generation process which uses most of the time, space and memory. Another drawback is it requires multiple scans of the database.

D. APRIORITID ALGORITHM

In this algorithm [4], database is not used for counting the support of candidate itemsets after the first pass. The process of candidate itemset generation is same like the Apriori algorithm. Another set C' is generated of which each member has

the TID of each transaction and the large itemsets present in this transaction. The set generated i.e. C' is used to count the support of each candidate itemset.
The advantage of this algorithm is that, in the later passes the performance of Aprioritid is better than Apriori.

E. APRIORIHYBRID ALGORITHM

As Apriori does better than Aprioritid in the earlier passes and Aprioritid does better than Apriori in the later
passes. A new algorithm [4] is designed that is Apriorihybrid which uses features of both the above algorithms. It uses Apriori algorithm in earlier passes and Aprioritid algorithm in later passes.

F. FP-GROWTH ALGORITHM

To break the two drawbacks [5] of Apriori algorithm,FP-growth algorithm is used. FP-growth requires constructing FP-tree. For that, it requires two passes. FP- growth uses divide and conquer strategy. It requires two scans on the database. It first computes a list of frequent items sorted by frequency in descending order (F-List) and during its first database scan. In the second scan, the database is compressed into a FP-tree [6]. This algorithm performs mining on FP-tree recursively. There is a problem of finding frequent itemsets which is converted to searching and constructing trees recursively. The frequent itemsets are generated with only two passes over the database and without any candidate generation process. There are two sub processes of frequent patterns generation process which includes: construction of the FP-tree, and generation of the frequent patterns from the FP-tree.

FP-tree is constructed over the data-set using 2 passes are as follows:
Pass 1:

1) Scan the data and find support for each item.
2) Discard infrequent items.
3) Sort frequent items in descending order which is based on their support.

Algorithm: FP growth. Mine frequent itemsets using an FP-tree by pattern fragment growth.

Input: *D*, a transaction database; *min sup*, the minimum support count threshold. Output: The complete set of frequent patterns. Method:
1. The FP-tree is constructed in the following steps: (a) Scan the transaction database *D* once. Collect *F*, the set of frequent items, and their support counts. Sort *F* in support count descending order as *L*, the *list* of frequent items.
(b)Create the root of an FP-tree, and label it as "null." For each transaction *Trans* in *D* do the following.
Select and sort the frequent items in *Trans* according to the order of *L*. Let the sorted frequent item list in *Trans* be [*pjP*], where *p* is the first element and *P* is the remaining list. Call insert tree ([*pjP*], *T*), which is performed as follows. If *T* has a child *N* such that *N.item-name=p.item-name*, then increment *N*'s count by 1; else

create a new node *N*, and let its count be 1, its parent link be linked to *T*, and its node-link to the nodes with the same *item-name* via the node-link structure. If *P* is nonempty, call insert tree(*P*, *N*) recursively.

2.The FP-tree is mined by calling FP growth(*FP tree*, *null*), which is implemented as follows. procedure FP growth(*Tree,* a)

(1) If *Tree* contains a single path *P* then

(2) For each combination (denoted as b) of the nodes in the path *P*

(3) Generate pattern b[a with *support count = minimum support count of nodes in* b;

(4) Else for each *ai* in the header of *Tree {*

(5) Generate pattern b = *ai* [a with *support count = ai*:*support count*;

(6) Construct b's conditional pattern base and then b's conditional FP tree *Tree* b;

(7) if *Tree*b 6= /0 then

(8) call FP growth(*Tree* s); }

Disadvantages of FP- Growth Algorithm

1. FP-Tree may not fit in memory!!
2. FP-Tree is expensive to build

G. IMPROVEDAPRIORITID ALGORITHM

This algorithm decrease the number of candidate items in the candidate item set C_k.. In the Apriori algorithm, C_{k-1} is compared with support level once it was found. Item sets less than the support level will be pruned and L_{k-1} will come out which will connect with itself and lead to C_k. The optimized algorithm is that, before the candidate item sets C_k come out, further prune L_{k-1}, count the times of all items occurred in L_{k-1}, delete item sets with this number less than k-1 in L_{k-1}. In this way, the number of connecting items sets will decrease, so that the number of candidate items will decline.

The following is the description of the optimized algorithm:

Input: affairs database *D*: minimum support level threshold is minsup

Output: frequent item sets *L* in *D*

1) L_1=frequent_1-itemsets(D);

2) For (k=2;$L_{k-1}\neq\varphi$;k++);

3) Prune*1*(L_{k-1});

4) C_k=apriori_gen(L_{k-1};minsup);

5) for all transactions $t\in D$

{

6) C= sumset (C_k,t); find out the subset including C_k

7) for all candidates $c\in C_t$

8) { c.count ++; }

9) Lk ={$c\in C_k|c$.count≥minsup} //result of Prune2(C_k) } }

10) Return Answer$\cup_k L_k$

Algorithm: Prune Function:
Input: set *k*-1 frequent items of L_{k-1} as input parameter
Output: go back and delete item sets with this number less than k-1 in L_{k-1}
Procedure Prune *1*(L_{k-1})
1) for all itemsets $L_1 \in L_{k-1}$
2) if count(L_1)$\leq k$-1
3) then delete all L_j from L_{k-1} ($L_1 \in L_{k-1}$)
4) return L'_{k-1} // go back and delete item sets with this number less than *k*-1 in L_{k-1}

3. CONCLUSION AND FUTURE WORK

There are various association rule mining algorithms.In this paper we have discussed six association rule mining algorithms with their example: AIS, SETM, Apriori, Aprioritid, Apriorihybrid, FP-growth. Each algorithm has some advantages and disadvantages. Our future work is to describe an optimized algorithm for association rule mining that is more accurate and efficient in terms of memory.

REFERENCES

[1] Agrawal, R., Imielinski, T. and Swami, A. Mining association rules between sets of items in large databases. In Proceedings of the ACM SIGMOD International Conference on Management of Data,207-216, 1993.

[2] Komal Khurana, Mrs. Simple Sharma, *A Comparative Analysis of Association Rules Mining Algorithms*, International Journal of Scientific and Research Publications, Volume 3, Issue 5 , May 2013 ISSN 2250-3153

[3] Ish Nath Jha Samarjeet Borah, *An Analysis on Association Rule Mining Techniques*, International Conference on Computing, Communication and Sensor Network (CCSN) 2012

[4] Manisha Girotra, Kanika Nagpal Saloni inocha Neha Sharma *Comparative Survey on Association Rule Mining Algorithms,* International Journal of Computer Applications (0975 – 8887) Volume 84 – No 10, December 2013

[5] Sotiris Kotsiantis, Dimitris Kanellopoulos, *AssociationRules Mining: A Recent Overview*, GESTS International Transactions on Computer Science and Engineering, Vol.32 (1), 2006, pp. 71-82

[6] Gagandeep Kaur, Shruti Aggarwal , *Performance Analysis of Association Rule Mining Algorithms*, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 8, August 2013, ISSN: 2277 128X

[7] Pratiksha Shendge, Tina Gupta, *Comparitive Study of Apriori & FP Growth Algorithms*, Indian journal of research, Volume 2, Issue 3, March 2013.

[8] Ming-Syan Chen, Jiawei Han, P.S.Yu, *Data mining: an overview from a database perspective*, IEEE Transactions on Knowledge and Data Engineering, Volume:8, Issue: 6 ISSN: 1041-4347, 866 – 883

[9] Parita Parikh, Dinesh Waghela, *Comparative Study of Association Rule Mining Algorithms*, *Parita Parikh* et al, UNIASCIT, Vol 2 (1), 2012, 170-172, ISSN 2250-0987.