IJCSE International Journal of Computer Science and Engineering Open Access
Review Paper Volume-2, Issue-5 E-ISSN: 2347-2693

Association Rule Mining

Priyank Saxena[1*] and Rachna Jain[2]

[1]*Amity Institute of Information Technology, Amity University, Noida, India,* priyanks.pro@gmail.com
[2]*Amity Institute of Information Technology, Amity University, Noida, India,* rjain1@amity.edu

www.ijcseonline.org
Received: Revised: Accepted: Published:

Abstract— Today, Association Rules are considered to be one of the more studied fields under Data Mining. It recently has come under a lot of notice by the data base warehouses. Its main use is to extract interesting associations, co-relations and frequent patterns among the groups of items recorded of the transactional databases or some different form of data storages. In this paper, a categorization and comparison of the different association rule algorithms that are present today is provided.

Keywords/Index Term— Data Mining, Association Rules, AssociationRule Algorithms, Database, Data Analysis

I. INTRODUCTION

Data Mining is the process of finding hidden patterns and correlations present in a transactional database and is viewed as a big part of the knowledge discovery process [Chn96] [Fayyd96]. Data mining functions consist of classification, cluster identification, predictionsand associations. Mining association rules are considered to be the most vital data mining applications, today. They are utilized today for distinguishing proof of at one time obscure connections among a gathering of things in a transactional database, first came into the highlight in the year 1993 [Agrwl93]. The relationships aren't formed upon the basic characteristics of these data items themselves (like it happens in functional dependencies) but instead on the co-occurrences of these set of data items together. The example below will give an idea about how these association rules are formed:

EXAMPLE 1:The management for a grocery store notice that, in 35% of the total transactions the bread is purchased with milk and also 50% of the time butter is also bought with it. Now, as per these freshly discovered affiliations, extraordinary presentation of margarine and bread together is set close to the milk bundles. These things are not put discounted but instead this is ruined expanding the general deal volume by taking profit of the recurrence by which these set of things are obtained together.

The above example states two association rules. In first it specifies that bread is purchased 35% of the time when milk is purchased. The second association rule specifies 50% of the time when milk is purchased so is butter. Retail stores often use association rules for analysing market basket transactions. The administration utilizes these companionship standards for expanding the viability and likewise the clear diminished expense connected with showcasing, publicizing, stock area in the floor. These guidelines are likewise of crucial use in the requisitions identified with disappointment forecasts this is carried out by distinguishing which occasions typically happens before a disappointment is distinguished in the telecom systems. This result in this paper is based upon the basket market database analysis. The paper provides a thorough survey of previous researches made on the frequently used association rule algorithms. The next section will let you understand the definitions of association rules. In Section III, a new classification and comparison of the basic algorithms is provided. Section IV is the summary part. The References are given in the Section V, information on different source code and data sources available in the market is provided, and Section VI comprises of the notation that are used in the paper.

Corresponding Author: *Priyank Saxena*

II. ASSOCIATION RULE

A Formal definition [Agrwl93] [Chng96]:

Definition 1 : Let I={I1,I2,..,In} be a set of n number of distinct qualities. A database 'D', in which every record (tuple) T has one unique identifier, consist of a group of items suchthat $T \subseteq I$. An association rule is an implication of the form $X \Rightarrow Y$, where $X, Y \subset I$, are sets ofitems called 'itemsets', and $X \cap Y = \varphi$. Where, X and Y are known as antecedent and consequent respectively.

There are two important entities for judging the effectiveness of an association rule;support (s) and confidence (α).

Definition 2: The support of an association standard is equal to the ratio (in percent) of the records that hold XUY to the aggregate number of records in that database.

Therefore, if the backing of a tenet is 10% then it implies that 10% of the aggregate number of records contain XUY. Support can be described as the 'statistical significance' for an association rule. The grocery store managements wouldn't be bothered about how milk and bread are related if, only, less than 10% of their total transactions have this combination of the items sold. Although a higher value of

'support' is always desired for an association rule, this doesn't always happen.
Let's consider one another example, this time in the field of Telecommunication networks, if we were using these association rules based on the group of events that occur before the failure, these can be very helpful in predicting the failure of the employed switching nodes.
Now even if these events don't occur very often, then too it's important that those association rules are showing these relationships.

Definition 3:The 'confidence' denoted by (α) is the ratio (in percentage) of the no. of records that consist of XUY to the no. of records which have 'X'. Therefore, if the confidence of a rule is 75%, then it describes that 75% of the total no. of records that contain 'X' also have 'Y'. The trust of a companionship tenet implies the sum (or the degree) of relationship in the set of information between "X" and 'Y'. Certainty is a measure of the quality of that specific acquaintanceship standard.

Usually, a large value of 'confidence' is expected fora rule. If in a network, a series of events occur only for a small percentage of the total times before a switch fails or if an item is bought very rarely alongwith milk, then these newly identified relationships won't be of much relevance for the management care takers.

The search and identification of these rules from a database source is the process of association rule mining that complete the thresholds of 'support' and 'confidence', as specified by the user.

III. CLASSIFICATION AND COMPARISION OF ALGORITHMS

This Section, for differentiating a large number of algorithms, provides a scheme for the classification of the association rule algorithms and also gives a qualitative correlation of these methodologies. The schematic order gives us the fundamental system that may be used for highlighting the major differences between all the algorithms that are present today (and also those which might be used in future).And in the second part of this section a qualitative comparison is provided to give currently proposed algorithms a performance analysis of high-level.

(a) Classification

This section identifies the various features by which we can classify these algorithms. Our approach categorizes the algorithms on the basis of various basic features and dimensions that are best for differentiating them.
Here, in our arrangement, the fundamental courses in which these methodologies vary from each other, are identified.

Table: Classification

DIMENSION	VALUES
Target	Complete,Constrained,Qualitative
Type	Regular,Generalized, Quantitative etc.
DataType	Database Data ,Text
DataSource	MarketBasket ,BeyondBasket
Technique	LargeItemset, StronglyCollectiveItemset
Itemset Strategy	Complete,Apriori,Dynamic, Hybrid
ItemsetDataStructure	Hash Tree, Trie, Virtual Trie, Lattice
Transaction Strategy	Complete,Sample,Partitioned
Transaction DataStructure	Flat File, TID
Optimization	Memory, Skewed,Pruning
Architecture	Sequential,Parallel
Parallel Strategy	None, Data,Task

Our classification uses the following dimensions:

Target:
Desired 'support' and 'confidence' thresholds are used by the basic association rule algorithms for finding all the rules. However, if only a subset of the algorithms were to be found, more efficient algorithms could be devised. One such algorithm which has been able to do this, did itbytaking the already generated rules and then adding constraint on them. The classification of algorithms is done as, namely: 'Complete' these are the ones where all association rules are found those comply with the 'support' and 'confidence' thresholds, 'Constrained' are the ones where some subsets of all the rules are found, based on some methods for limiting them, and 'Qualitative' are those where subset of standards are created focused around some extra procedures, (apart from the support and confidence) are needed to be satisfied.

Type:
Here, the type of that particular association rule is generated (eg. Regular (Boolean), Spatial, Qualitative, Generalized, etc.)

Data Type:
Separated from the information that is put away in the database, tenets of a "plain" content may have the capacity to uncover extremely imperative data.example: In some article of knowledge discovery,"data", "mining" and "decision" might be high in dependence.

Data Source:
The data which is absent in the database of a company , even association rules of those data plays a very important role for the motive of decision making in the company along with the market basket data as well.

Technique:
All of the approaches which are discovered from starting to till now are actually based upon the first finding the large

itemsets. There are also other different techniques which do not need these large itemsets to be found first. Although until now such techniques are not yet found out which don't generate large itemsets, but obviously the possibility of finding out such techniques, in future, does present along with the capacity which provides improved performances. However, [Agrwl98] suggested "strongly collective itemsets" for the evaluation and finding of the itemsets. The expressions "backing" and "trust" are fundamentally not the same as the substantial itemset approach. An itemset "I" is falling under "unequivocally aggregate" at level 'N', if the aggregate quality C(n) of 'I'and and in addition any subset of "I" is at any rate 'N'.

Itemset Strategy:
Each algorithm have to deal with the process of generation of itemsets which is different everytime. This feature shows exactly how a certain algorithm looks at the 'transactions' and also when the 'itemsets' are produced. The 'Complete' technique can produceand as well as can number all potentialitemsets. Here, the most widely recognized methodology is Apriori. In this approach, a set of 'itemsets to count' can producedbefore the transaction scan. This 'set' stays static during the complete process. A dynamic strategy is used for generating the itemsets during the database scan itself. A hybrid technique is used for generating some itemsets even before the scanning of the database, but during the scan addition of all these new itemsets is done to this counting set.

Transaction Strategy:
All the different algorithms deal with the group of transactions differently. This feature shows exactly how an algorithm scans the group of a transaction. The complete methodology is utilized for looking at of every last one of transactions in the database,. Under the example approach, a few subsets (examples) of the database are to be analyzed before the transforming of the complete database. For separating the database into separate partitions the partition techniques are used. The undertaking of filtering of the database obliges that all the segments must be inspected in place however independently.

Itemset- Data Structure:
Different forms of data structures can be used, as the itemsets get generated for keeping track of these itemsets. The most common and famousapproache is a Hash Tree. Also, a lattice or a trie is used as well. No less than, one methodology proposes a virtual trie structure where just some segment of a complete triegets emerged.

Transaction Data Structure:
Each methodology/calculation accept that transactions in a database are kept in some essential structure which typically is some even record (a database without structured relationships) or a Transaction ID list.

Optimization:
There are some late calculations that have been recommended that are a change on the prior calculations by putting an enhancement method. Different systems have taken a gander at enhancement by contemplating and the accessible principle memory, whether the information there is skewed or not, and additionally the on pruning of the itemsets to be considered.

Architecture:
For a centralized single processor architecture, algorithms are designed as a sequential function. And for a multiprocessor or a distributed architecture, the algorithms are designed so as to function in a parallel manner.

Parallel Strategy:
The Parallel algorithms falls under the task parallelism or data parallelism through the figure given below:

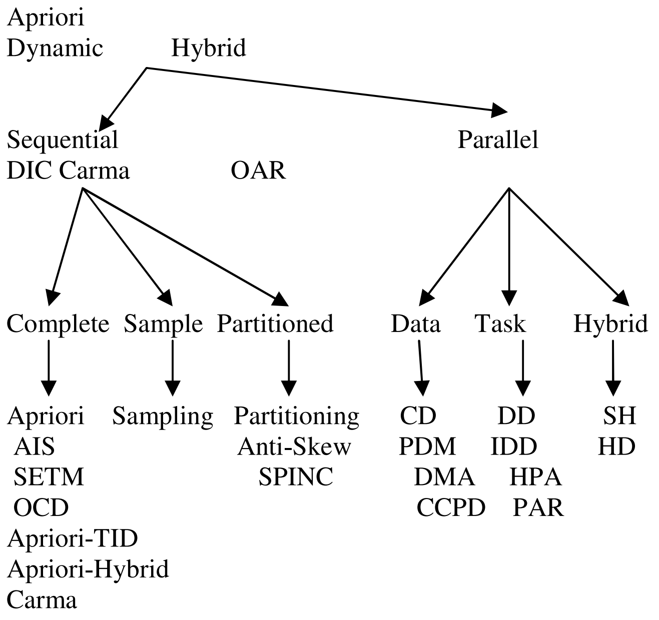

Figure here shows the order of a few calculations. Just the calculations that create extensive itemsets, and are finished and customary, are demonstrated. Calculations are demonstrated as the leaf hubs.

(b) Comparing Algorithms

Several metrics are considered here for the comparison of the various algorithms. The estimation of time requirements is done by evaluating the maximum number of required database scans (also called as the I/O estimate) and the most noteworthy number of correlation operations required (also called as the CPU estimate). The estimation for space prerequisites is carried out by taking a gander at the most astounding number of checked applicants throughout any database filter. Additionally, since the stockpiling for basically all the transaction databases is to be carried out on the auxiliary circles, and as we realize that the I/O overhead is of substantially more criticalness than the CPU overhead, the center is made on the aggregate number of sweeps in the complete database.now clearly, the most exceedingly awful situation, here, will be the place every transaction in the database has all the things in it. Here, let "n" be the amount of things in each of the transaction, and let Lk be the expansive itemsets with "k" things in a database 'D'. Presently, the amount of vast itemsets will be '2n'. Presently, in level-wise strategies, in the same way as AIS, SETM, Apriori, all the expansive itemsets in L1 found throughout the first database examine itself. Also, all the expansive itemsets in L2 found throughout the second database sweep, et cetera. Thusly the main itemset in Ln might be gotten throughout the nth output. All the calculations get ended when no extra sections are processed in the substantial itemsets, so an additional sweep is required. Consequently, the whole database will be filtered at most (n+1) times. An alternate approach additionally called as Apriori-TID checks the whole database in its first pass, and then instead of using the entire database, it just uses the Ck in its (k+1)thpass. However, in the worst scenario, it doesn't help even a bit, the reason being that during the entire process, the Ck will contain all the transactions along with their items. And on the other side, the OCD technique does the scanning of the entire database only once at the starting of the algorithm and as a result obtains the large itemsets in L1. Later on, the OCD approach and the Sampling approach use only a part of the entire database and the information obtained during the first pass for finding the candidate itemsets of Lk, where 1<k<=n. And in their second scan the 'support' of each of the candidate itemset is calculated. Thus, there will only be 2 scans in the worst case given enough main memory. The PARTITION technique declines the number of database scans to 2, and in additons also reduces the I/O overhead. Similarly, CARMA needs at most 2 database filters.

The viability (likewise called as the integrity) of a calculation relies on upon the precision of the amount of "genuine" applicants that it produces. For producing competitor sets, expansive itemsets of past pass(es) are utilized by all the calculations. Also for era of hopeful itemsets, huge itemsets of past itemsets are brought into the principle memory. And then again, for finding the support counts, the candidate itemsets are required in the main memory. Since, due to the possibilities of unavailability of enough memory, different algorithms suggest different types of storage structures and buffer management. AIS proposed that, if needed, Lk-1 can be disk-resident. SETM suggested that if Ck is too large to fit into the main memory then write it to the disk in FIFO manner. The Apriorifamily of calculations recommended that for discovering Ck, keep Lk-1 on circle and bring it into the fundamental memory one and only piece at once. Nonetheless, for discovering help tally in both Apriori-TID and Apriori-Hybrid, Ck needs to be in the main memory . However, interestingly on the other side, all the other approaches/techniques assumed that for handling these sorts of problems, there is enough memory. The various consecutive methods, for example, PARTITION, Sampling, DIC and CARMA, considerto be a suitable a piece of the whole database that could fit in the primary memory. The Apriori family proposed some suitable information structures like hash tree or show for substantial itemsets and, and also, applicant sets which are exhibited in the Table given below.

Algorithm	**Scan**	**Data Structure**	**Description**
AIS	n+1	Not Specified	Suitable for low cardinality scanty transaction database; Single ensuing
SETM	n+1	Not Specified	SQL compatible
Apriori	n+1	Lk-1 : HashTable Ck: HashTree	Transaction database with moderate cardinality; Outperforms both AIS and SETM; Base calculation for parallel calculations
Apriori-TID	n+1 Ck: array indexed by TID *Ck*: Sequential structure ID: bitmap	Lk-1 : HashTable	Moderate with bigger number of Ck; Outperforms Apriori with littler number of Ck;
Apriori-Hybrid	n+1	Lk-1 : HashTable 1st Phase: Ck: HashTree 2nd phase**:** Ck: array indexed by IDs *Ck*: Sequential structure ID: bitmap	Superior to Apriori. On the other hand, changing from Apriori to Apriori-TID is unreasonable; Very urgent to evaluate the move point.
OCD	2	Not specified	Relevant in huge DB with more level help edge.
Partition	2	HashTable	Suitable for huge DB with high cardinality of information; Favors homogenous information appropriation

Sampli ng	2	Not Specified	Applicable in very large DB with lower support.
DIC	Depends on interval size	Trie	Database saw as interims of transactions; Hopefuls of expanded size are produced at the end of an interim
CARM A	2	HashTable	Relevant where transaction successions are perused from a Network; Online, clients get ceaseless criticism and change help and/or trust whenever throughout procedure.
CD	n+1	HashTable and Tree	DataParallelism.
PDM	n+1	HashTable and Tree	DataParallelism; with early candidate pruning
DMA	n+1	HashTable and Tree	DataParallelism; with candidate pruning
CCPD	n+1	HashTable and Tree	DataParallelism; on shared-memory machine
DD	n+1	HashTable and Tree	TaskParallelism; round-robin partition
IDD	n+1	HashTable and Tree	TaskParallelism; partition by the first items
HPA	n+1	HashTable and Tree	TaskParallelism; partition by hash function
SH	n+1	HashTable and Tree	DataParallelism; candidates generated independently by each processor.
HD	n+1	HashTable and Tree	HybridData and TaskParallelism; grid parallel architecture.

Table: Comparison of Algorithms

However, some approaches like AIS and SETM didn't advice any storage structures.In Most cases, the commercially available implementations for generating association rules rely upon the use of the Apriori technique. There are some algorithms which are more suitable for use in specific conditions. For instance, when the number of items in the database is large, AIS does not perform well. Thus, AIS is more suitable for transaction databases which have low cardinality. Apriori-TID takes more execution time than Apriori in earlier passes. However, in later passes Apriori-TID outperforms Apriori. Through proper switching Apriori-Hybrid shows excellent performance. Although, switching from Apriori to Apriori-TID is very crucial and costly affair. OCD is an approximate technique, it is still very much effective to find frequent itemsets with lower threshold support. CARMA is an online user interactive feedback oriented technique. It is best suited where transaction successions are perused from a system. The table given above condenses and gives intends to analyze the different calculations. This table incorporates the most extreme number of outputs, information structures proposed, and particular remarks.

IV. SUMMARY

Mining Association Rules is considered to be one of the more investigated andused capacities in information mining. Cooperation principles are essential to both database analysts the information mining clients. This paper gives a study of at one time made investigates in the field of cooperation standard calculations, it additionally gives a grouping techniques and likewise correlation of the distinctive methodologies that are being used today.

ACKNOWLEDGMENT

I WOULD LIKE TO THANKS MY GUIDE RACHNA JAIN, FAMILY MEMBERS AND FELLOW STUDENTS AND MY FRIENDS.

REFERENCES

[Agrwl93] RakeshAgrawal, Tomasz_Imielinski and Arun N. Swami, Mining_Association_RulesBetweenSets of Items in Large_Databases.

[Agrwl98] Charu C. Aggarwal and Philip_S. Yu, A New Framework for Itemset_Generation.

[Chn96] Ming-Syan Chen, Jiawei Han and Philip_S. Yu, Data Mining: An Overview from a Database_Perspective.

[Fayyd96] Usama M. Fayyad, Gregory_Piatetsky-Shapiro, and Padhraic Smyth, From Data Mining to knowledge Discovery: An Overview, Advances in Knowledge Discovery and Data Mining, pp 1-34.

[Chng96c] David Wai-Lok Cheung, Ada Wai-Chee Fu, Vincent T. Ng, and Yongjian Fu, Efficient Mining of Association_Rules in Distributed_Databases, Vol. 8, No. 6, pp. 911-922.

NOTATIONS

[1] I: Set of data items
[2] n: No. of data items
[3] D: Transactional database
[4] s: Support
[5] α: Confidence
[6] T: Tuples in database

[7] X,Y: Itemsets
[8] $X \Rightarrow Y$: Association rule
[9] Lk: Set of large itemsets of size 'k'
[10] Li: Set of large itemsets for partition Di
[11] L: Set of large itemsets
[12] l :Large itemset