A Survey on Association Rule Minning Algorithms

S Sreeraj
Dept. of Computer science and engineering
Sri Krishna College of Engineering and Technology,Coimbatore
sreeraj0095@gmail.com

S Sandeepa
Dept. of Cyber Forensic and Information Security
ER & DCI Institute of Technology,Trivandrum
sandeepa.s567@gmail.com

ABSTRACT:

In this paper, the detail description of survey focus on different kinds of Association Rule algorithms and also show case on the strength and weakness of each algorithm. we already know that, Data mining is the process of discovering novel ,useful and human understandable pattern in premises of information from large quantity of data. It has many methodologies for catching the information like Association rule mining, classification, clustering, regression etc.

Among them Association rule mining is one of the most significant standing out investigation area in data mining. In past investigation , many algorithms were constructed like Apriori, Fp-Growth, Eclat, STAG etc. In this paper we discuss this algorithms in detail based upon the qualitative behaviour of each algorithms.

Keywords: **Data Mining,Association rule mining,Aprori,Knowledge ,Data.**

I. INTRODUCTION

Data mining or data or knowledge discovery is the process of analyzing data from different perspectives and summarizing it into useful information (ie.) information that can be used to increase revenue, cuts costs, or both.

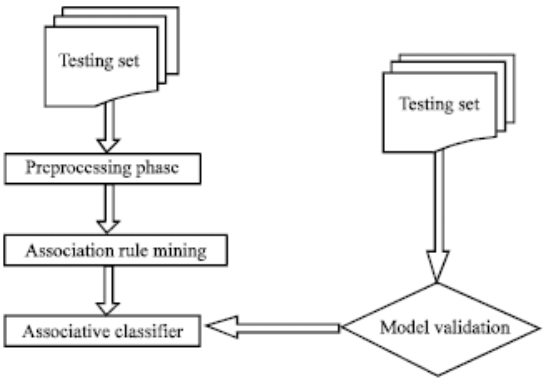
Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

II. ASSOCIATION RULE LEARNING

Association rule learning is a rule-based machine learning method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness.

Based on the concept of strong rules, association rules for discovering regularities between products in large-

scale transaction data recorded by point-of-sale (POS) systems in supermarkets.



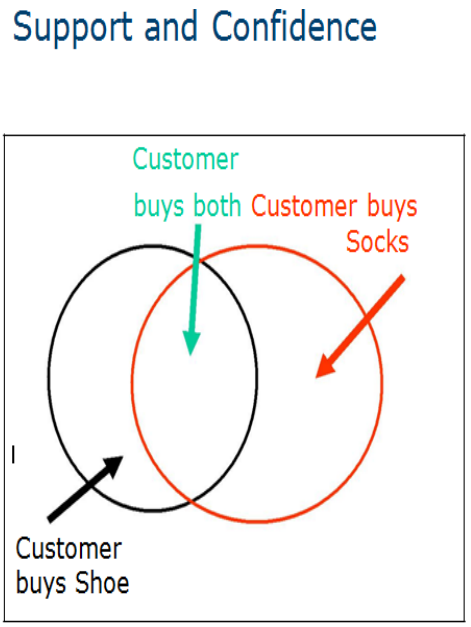
Association rules are if/then statements that help uncover relationships between seemingly unrelated data in a relational database or other information repository. An example of an association rule would be "If a customer buys a dozen eggs, he is 80% likely to also purchase milk."

For example: In 80% of the cases when people buy bread, they also buy milk. This tells us of the association between Shoes and Socks. We represent it as -

Shoes => Socks | 80%

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

Association rules are created by analyzing 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.



A. CONCEPT OF ASSOCIATION RULE MINNING:

Implication of the form X Y, where X and Y are itemsets

Example: {Shoes} {socks}

1) Rule Evaluation Metrics, Support & Confidence
2) Support (s): Frequency of occurrence of an itemset ,Fraction of transactions that contain an itemset .represented by **XUY**
3) Confidence (c): Measures how often items in Y appear in transactions that contain X represented by **X/Y**
4) Support count (*) Frequency of occurrence of an itemset.

B. ASSOCIATION RULE MINING ALGORITHMS:

1) APRIORI ALGORITHM :

This algorithm has been very often used for mining of frequent item sets and to key out associations. The major deviation in Apriori is the less candidate itemsets it supplies for testing in every database pass. The hunt for association rules is monitored by two parameteric quantities: support and confidence. Apriori Algorithm yields an association rule if its support and confidence rates are in a higher place customized threshold values. The output is ranked by confidence. If so many rules have as like confidence, they are ranked by support. Hence Apriori prefers more confident rules and split these rules as to a greater extent novel .This algorithm utilizes BFS methodology. It keeps track of support of item sets. It has a candidate generation function which makes utilizes of the downward closure attribute of the support. Apriori execution utilizes a data structure that right away maps a prefix tree. The tree grows top-down stage by stage, cuts off those branches that cannot hold a frequent item set. Apriori algorithm needs two significant things: minimum support and minimum confidence.

a) Merits:

1. Fast
2. Less candidate sets.
3. Generates candidate sets from only those items that were found large.

b) Demerits:

1. Takes a lot of memory

2) APRIORI TRANSACTION IDENTIFIER ALGORITHM (Aprori TID)

Just as like as the Apriori algorithm, AprioriTID algorithm purposes the generation function in order to see the candidate item sets. The only deviation among the two algorithms is that, in AprioriTID algorithm the database is not denoed for counting support after the first pass itself. Here a group of candidate item sets is used for this purpose for k>1. When a transaction does not have a candidate k-item set in that kind of a case the group of candidate item sets will never have any entry for that transaction. This will cut down the number of transaction in the set holding the candidate item sets when compared to the database. As value of k goes up every entry will become lesser than the corresponding transactions as the number of candidates in the transactions will continue to go smaller. Apriori only will do better than AprioriTID in the initial passes but to a greater extent passes are given AprioriTID certainly has better operation than Apriori.

Merits:

1. Doesn't use whole database to count candidate sets.
2. Better than SETM.
3. Better than Apriori for small databases.
4. Time saving.

3) APRIORI HYBRID ALGORITHM:

Apriori and also AprioriTID use the like as candidate generation routine and therefore count the like as item sets. Apriori studies all the transaction in the database. On the other side, rather than skiming the database, AprioriTID skims candidate item sets used in the former pass for obtaining support counts. Apriori Hybrid utilizes priori in the initial passes and changes over to AprioriTid when it

anticipates that the candidate item sets at the death of the pass will be in memory.

a) Merits:

1. Better than both Apriori and AprioriTID.

4) ARTIFICIAL IMMUNE SYSTEM ALGORITHM (AIS):

This algorithm is utilized to trace the frequent item sets. It utilizes candidate generation in order to trace them. The candidates are generated on the fly and now they are then likened with the already generated frequent item sets. One of the de-merit of this algorithm adds the generation and counting of so many candidate item sets that turn out to be little. This was the first algorithm to bring in the trouble of generation of association rules. The de-merit of the AIS algorithm is that it makes more than one passes over the database. Furthermore, it generates and counts too many candidate itemsets that project out to be small, which needs more space and waste much attempts that projected out to be woth less .

Merits:

1. Better than SETM.
2. Easy to use

5) *SET ORIENTED MINING ALGORITHM (SETM)*

Just as same as, the AIS algorithm, this algorithm also do a very quick counting. It is grounded on the transaction learn from the database. But SETM was produced for SQL and utilizes relational operations. While, SETM utilizes standard SQL join operation for the generation of candidates and then sorts out candidate generation from counting. Intially the candidates are generated using equi-joins and then classified and the ones that don't fits the minimum support are taken off.

a. Merits:

1. Separates generation from counting.

b. DeMerits:

1. Very large execution time.
2. Size of candidate set large

6) ECLAT ALGORITHM

Eclat effectuation maps the set of transactions as a bit matrix and it intersects rows give the collaboration of item sets. It sticks to depth first traversal of a prefix tree. Eclat algorithm does holds extra calculation of operating cost of constructing or exploring complex data structures, or does it have to get all the su bsets of a piece of transaction. Eclat algorithm utilizes depth first search concept. It cannot utilize the horizontal dataset. If there exists any horizontal dataset.

eclat algorithm has various kinds of phases they are pointed out as follows:

1. Initialization phase
2. Transformation phase
3. Asynchronous phase
4. Reduction phase

i. Bit Matrices

A flexible way to map the transactions for the Eclat algorithm is bit matrix, in which each row fits to an item, each column fits to a transaction . A bit is set in this matrix if the item fitting to the row is arrested in the transaction fitting to the column, otherwise it is cleared. There are basically two ways in which such a bit matrix can be mapped: In one way as a true bit matrix,

with one memory bit for each item and transaction, or using for each row a list of those columns in which bits are set.

ii. Search Tree Traversal

As already cited, Eclat explores a prefix tree in depth first order. The alter of a node to its first child consists in constructing a new bit matrix by intersecting the first row with all the accompanying rows. For the second child the second row is intersected with all accompanying rows and so on. The item equating to the row that is intersected with the accompanying rows thus is included to make the common prefix of the item sets refined in the equating child node. Of course, rows equating to infrequent item sets should be removed from the matrix that is made which can be served most flexibly if we store with each row the equaing item identifier. Intersecting two rows can be served by a simple logical and on a static length integer vector if operatedwith a true bit matrix.

a. Merits:

1. Less memory usage.
2. Lower minimum support.

b. Demerits:

1. Apriori wins in cases where candidate sets are more

. 7) FREQUENT PATTERN GROWTH ALGORITHM(FP)

FP-Tree frequent pattern mining is utilized in the growth of association rule mining. FP-Tree algorithm sweeps over the problem found in Apriori algorithm. By keeping off the candidate generation procedure and less passes over the database, FP-Tree was found to be quicker than the Apriori algorithm It takes a divide and conquer strategy. Intially it packs together the database mapping frequent items into a frequent –pattern tree or FP-tree. It holds the item set association information and packed databases are spilited into a group of conditional databases, a piece of one associated with a frequent item. It takes the support of prefix tree delegacy of the given database of transactions (called FP tree), which spares considerable quantity of memory for storing the transactions. An FP-Tree is a prefix tree for transactions. Every node in the tree maps one item and each path maps the set of transactions that require with the specific item. All nodes pointing to the like as item are connected together in a list, so that all the transactions that holding the as like item can be well grounded and counted. Large databases are packed together into compact FP tree model. FP tree modelholds significant data about frequent item sets in a database.

a. Merits

1. Only 2 passes of dataset.
2. Compresses data set.
3. No candidate set generation required
4. Better than éclat, Apriori

b. Demerits:

1.Using tree structure creates complexity

8) RECURSIVE ELIMINATION ALGORITHM

Recursive elimination is grouned on a pace by pace elimination of items from the transaction database combined with a recursive processing of transaction subsets. RElim is a program to discover frequent item sets (additionally shut and maximal) with the relim algorithm

(recursive elimination), which is roused by the FP-growth algorithm, however does its work without prefix trees or some other confused data structures. The principle character of this algorithm is not its speed (Inspite of the fact that it is not moderate, but rather even beats Apriori and Eclat on a few data sets), however the straightforwardness of its structure. Fundamentally all the work is done in one recursive capacity of genuinely few lines of code.

a. Merits:

1. Better than Apriori in all cases.

b. DeMerits:

1. Less than eclat in all cases.

9) ALGORITHM Stacked Graph (STAG)

A fresh association rule mining algorithm, STAG (Stacked Graph) has been introduced, grounded on graph theoretic approach. Two troubles have been solved: the first one targeting at shortening the I/O drastically and the second one to bring a sliming in computational time, run time storage and I/O at the same instance. This is achieved by one-scan STAG and two-scan STAG algorithm. STAG overtakes the impossibility of answering a very less backing online query by the user, if used for OLAP functions. In comparison to disk based algorithms like Apriori,Aprori hybrid; it shortens input-output processingby skimming a database only once or at most twice and the inclusion of new trans- actions do not need re-skimming of existing transactions. Some association rule mining algorithms need the items in a transaction to be lexicographically ordered or incorporate an extra step of ordering the items as per the support value but there is no such infliction on items in STAG. The order of skimming of transactions is extraneous and the items need not be ordered . The algorithm uses a depth first scheme to spread the search space of potential frequent item sets. The algorithm consists of two steps:

Building a graph structure by skimming the transactions in the database and usiing this model in the second step to trace out the frequent item sets, without skimming the database.

a. Merits:

1.shortening I/O drastically.
2.cut down execution time.

b. Demerits:

1.increases overhead highly

III. CONCLUSION

In this paper, We have focused on comparatively examining the strength and weakness of various types of Association Rule mining algorithms like Aprori,aprori TID,ELCAT,STAG etc by considering various behavioural aspects of ARM algorithms like Meomry consumption ,run time,performance aspects,usage criteria etc. This survey greatly helps in analysing the characters of various ARM algorithms

REFERENCES

[1] Mr. Prasanth Kumar ,Mr. B. Yakhoob ,Mrs. N. Deepika—Association Rule Mining Algorithms: A Comparative Survey, International Journal of Advanced Engineering and Global Technology Vol-04, Issue-06, November 2016 .

[2]Mr. Subhash Rohit ― Association Rule Mining Algorithms: Survey : International Research Journal of Engineering and Technology (IRJET) Volume: 03 Issue: 10Oct2016.

[3] B. Chandra and Gaurav―Indian Institute of Technology, Delhi,(Eds.): ICONIP 2007, Part II, LNCS 4985, pp. 366–375, 2008.

[4] Aggarwal, R., and Srikant, R. ―Fast Algorithms for Mining Association Rules. Proceedings of the 20th VLDB Conference Santiago, Chile, 1994.

[5] Borgelt, C. "Efficient Implementations of Apriori and Eclat‖. Workshop of frequent item set mining implementations (FIMI 2003, Melbourne, FL, USA).

[6] Khurana, K., and Sharma, S. ―A comparative analysis of association rule mining algorithms‖.International Journal of Scientific and Research Publications, Volume 3, Issue 5, May 2013.

[7] Hunyadi, D. ―Performance comparison of Apriori and FP- Growth Algorithms in Generating Association Rules‖. Proceedings of the European Computing Conference ISBN: 978-960-474-297-4.

[8] Webb, I.G. ―Efficient Search for Association Rules‖. 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2000), Boston, MA, New York, NY.

[9] Thieme, S.L. ―Algorithmic Features of Eclat‖. FIMI, Volume 126 of CEUR Workshop Proceedings, CEUR- WS.org, 2004.