Comparative Survey on Association Rule Mining Algorithms

Manisha Girotra
Northern India Engineering College
Guru Gobind Singh Indraprastha University, Delhi, India

Kanika Nagpal
Northern India Engineering College
Guru Gobind Singh Indraprastha University, Delhi, India

Saloni Minocha
Northern India Engineering College
Guru Gobind Singh Indraprastha University, Delhi, India

Neha Sharma
Northern India Engineering College
Guru Gobind Singh Indraprastha University, Delhi, India

ABSTRACT
Association rule mining has become particularly popular among marketers. In fact, an example of association rule mining is known as market basket analysis. The task is to find which items are frequently purchased together. This knowledge can be used by professionals to plan layouts and to place items that are frequently bought together in close proximity to each other, thus helping to improve the sales. Association rule mining involves the relationships between items in a data set. Association rule mining classifies a given transaction as a subset of the set of all possible items. Association rule mining finds out item sets which have minimum support and are represented in a relatively high number of transactions. These transactions are simply known as frequent item sets. The algorithms that use association rules are divided into two stages, the first is to find the frequent sets and the second is to use these frequent sets to generate the association rules. In this paper the applications, merits and demerits of these algorithms have been studied. This paper discusses the respective characteristics and the shortcomings of the algorithms for mining association rules. It also provides a comparative study of different association rule mining techniques stating which algorithm is best suitable in which case.

Keywords
Association rule mining, market based analysis, frequent sets, transactions, association rules, relationships.

1. INTRODUCTION
In today's scenario, every company needs information related to strategies for the purpose of business analysis. This information is to be extracted from a collection of raw data sets and then is to be represented in useful and understandable structure. Data mining is the efficient discovery of valuable, non-obvious information from a large collection of data [1]. It is a process of "Knowledge Discovery in Databases" process also known as KDD. It is an interdisciplinary subfield of computer science. It is the process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, statistics and database systems. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. Association rule learning is a popular method which is used in data mining for discovering relations between variables in extensive databases. These are applied in order to identify strong rules discovered in databases using different measures [2]. One limitation of the standard approach of discovering associations is by searching massive number of possible associations to look for collections of items that appear to be associated, there is a large risk of finding many spurious associations. These are collections of items that co-occur with unexpected frequency in the data, but only do so by chance. For example, suppose a collection of 10,000 items is considered and looking for rules containing two items in the left-hand-side and 1 item in the right-hand-side. There are approximately 1,000,000,000,000 such rules. If statistical test is applied for independence with a significance level of 0.05 it means there is only a 5% chance of accepting a rule if there is no association. If it is assumed there are no associations, it should nonetheless be expected to find 50,000,000,000 rules. Statistically sound association discovery controls this risk, in most cases reducing the risk of finding any spurious associations to a user-specified significance level [3]. Many algorithms for generating association rules were presented over time. Different types of mining techniques explained in this paper are listed below:

- AIS
- SETM
- Apriori
- AprioriTID
- Apriori hybrid
- Eclat
- Recursive Elimination
- FP-Growth

The paper is divided into several sections for ease of understanding. Section 2, Literature Review, represents the study of previous year papers written on association rule mining techniques. Section 3 represents the summary of all important algorithms written on association rule algorithm. Section 4 states the conclusion drawn from the comparison table.

2. LITERATURE REVIEW
Jochen Hipp *et al* provided several efficient algorithms that cope up with the popular and computationally expensive task of association rule mining with a comparison of these algorithms concerning efficiency [4]. He proposed that the algorithms show quite similar runtime behavior in their experiments.

Rakesh Aggarwal and Ramakrishnan Srikant presented two new algorithms, Apriori and AprioriTID, for discovering all

significant association rules between items in a large database of transactions and compared these algorithms to the previously known algorithms, the AIS and SETM algorithms [2]. They proposed that these algorithms always outperform AIS and SETM.

Daniel Hunyadi presented a comparative study of Apriori and FP Growth association rule algorithms [5]. He suggested that having its origin in the analysis of the marketing bucket, the exploration of association rules represents one of the main applications of data mining. Their popularity is based on an efficient data processing by means of algorithms. Being given a set of transactions of the clients, the purpose of the association rules is to find correlations between the sold articles. Knowing the associations between the offered products and services helps those who have to take decisions to implement successful marketing techniques.

Christian Borgelt provides efficient implementation of Apriori and Eclat algorithms [6]. Finding frequent item sets in a set of transactions is a popular method for so called market basket analysis, which aims at finding regularities in the shopping behavior of customers of supermarkets, mail-order companies, on-line shops etc. In particular, it tries to identify sets of products that are frequently bought together. The main problem of finding frequent item sets, i.e. item sets that are contained in a user-specified minimum number of transactions, is that there are so many possible sets, which render naive approaches infeasible due to their unacceptable execution time. Among the more sophisticated approaches two algorithms known under the names of Apriori and Eclat are most popular. Both rely on a top down search in the subset lattice of the items. He proposed for free item sets Eclat wins the competition with respect to execution time and it always wins with respect to memory usage. The data set in which it takes lead is for the lowest minimum support value tested, indicating that for lower minimum support values it is the method of choice, while for higher minimum support values its disadvantage is almost negligible. For closed item sets the more efficient filtering gives Apriori a clear edge with respect to execution time. For maximal item sets the picture is less clear. If the number of maximal item sets is high, Apriori wins due to its more efficient filtering, while Eclat wins for a lower number of maximal item sets due to its more efficient search.

Lars Schmidt-Thieme presented a paper on Eclat algorithm and proposed the following two conclusions:

1) At least for dense datasets, Eclat is faster than all its competitors [7].

2) For sparse datasets, Eclat is not suitable. The latter conclusion is not very surprising. Eclat is used as a basic algorithm and has a bundle of optional algorithmic features that are taken partly from other algorithms like Apriori.

Christian Borgelt presented a paper on Recursive Elimination algorithm [8]. He proposed that if a quick and straightforward implementation is desired, it could be the method of choice. Even though its underlying scheme—which is based on deleting items, recursive processing, and reassigning transactions—is very simple and works without complicated data structures, recursive elimination performs surprisingly well.

R.Divya and S.Vinod Kumar presented a paper in which various algorithms have been proposed for mining association rule but in every algorithm they found a common drawback of various scans over the database [9]. According to the paper association rule mining is to find out association rules that satisfy the predefined minimum support and confidence from a given database. The problem is usually decomposed into two sub problems. One is to find those item sets whose occurrences exceed a predefined threshold in the database; those item sets are called frequent or large item sets. The second problem is to generate association rules from those large item sets with the constraints of minimal confidence.

Komal Khurana and Mrs. Simple Sharma presented a paper[10]. This paper represents comparison of five association rule mining algorithms: AIS, SETM, Apriori, AprioriTID and Apriori Hybrid . The AprioriTID and Apriori Hybrid have been proposed to solve the problem of Apriori algorithm. From the comparison they conclude that the Apriori Hybrid is better than Apriori and AprioriTID because it reduces overall speed and improves the accuracy.

Ziauddin *et al* researched on association rule mining. They presented a survey of research work since its beginning [11]. He however proposed that association rule mining is still in a stage of exploration and development. There are still some essential issues that need to be studied for identifying useful association rules.

3. INTRODUCTION TO ALGORITHMS

3.1 AIS ALGORITHM

This algorithm is used to detect frequent item sets. It uses candidate generation in order to detect them. The candidates are generated on the fly and they are then compared with the already generated frequent item sets. One of the disadvantage of this algorithm includes the generation and counting of too many candidate item sets that turn out to be small. This was the first algorithm to introduce the problem of generation of association rules**.** The drawback of the AIS algorithm is that it makes multiple passes over the database. Furthermore, it generates and counts too many candidate itemsets that turn out to be small, which requires more space and waste much efforts that turned out to be useless [10].

3.2 SETM ALGORITHM

Just like the AIS algorithm, this algorithm also does on the fly counting. It is based on the transaction read from the database. But SETM was created for SQL and uses relational operations. Whereas SETM uses standard SQL join operation for the generation of candidates and then separates candidate generation from counting. First the candidates are generated using equi-joins and then sorted and then the ones that don't meet the minimum support are removed.

3.3 APRIORI ALGORITHM

This algorithm has been very frequently used for mining of frequent item sets and to discover associations. The major difference in Apriori is the less candidate itemsets it generates for testing in every database pass. The search for association rules is guided by two parameters: support and confidence. Apriori returns an association rule if its support and confidence values are above user defined threshold values. The output is ordered by confidence. If several rules have the same confidence then they are ordered by support. Thus Apriori favors more confident rules and characterises these rules as more interesting [10].This algorithm uses a breadth first search approach. It counts support of item sets. It has a candidate generation function which makes use of the downward closure property of the support. Apriori implementation uses a data structure that directly represents a

prefix tree. The tree grows top-down level by level, removing those branches that cannot contain a frequent item set.

3.4 APRIORI TID ALGORITHM

Just like the Apriori algorithm, AprioriTID algorithm uses the generation function in order to determine the candidate item sets. The only difference between the two algorithms is that, in AprioriTID algorithm the database is not referred for counting support after the first pass itself. Here a set of candidate item sets is used for this purpose for k>1. When a transaction doesn't have a candidate k-item set in such a case the set of candidate item sets will not have any entry for that transaction. This will decrease the number of transaction in the set containing the candidate item sets when compared to the database. As value of k increases every entry will become smaller than the corresponding transactions as the number of candidates in the transactions will keep on decreasing. Apriori only performs better than AprioriTID in the initial passes but more passes are given AprioriTID certainly has better performance than Apriori.

3.5 APRIORI HYBRID ALGORITHM

Apriori and AprioriTID use the same candidate generation procedure and therefore count the same item sets. Apriori examines every transaction in the database. On the other hand, rather than scanning the database, AprioriTID scans candidate item sets used in the previous pass for obtaining support counts[10]. Apriori Hybrid uses Apriori in the initial passes and switches to AprioriTid when it expects that the candidate item sets at the end of the pass will be in memory.

3.6 ECLAT ALGORITHM

Eclat implementation represents the set of transactions as a bit matrix and intersects rows give the support of item sets. It follows a depth first traversal of a prefix tree.

3.6.1 Bit Matrices

A convenient way to represent the transactions for the Eclat algorithm is a bit matrix, in which each row corresponds to an item, each column to a transaction (or the other way round). A bit is set in this matrix if the item corresponding to the row is contained in the transaction corresponding to the column, otherwise it is cleared. There are basically two ways in which such a bit matrix can be represented: Either 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. (The latter representation is equivalent to using a list of transaction identifiers for each item.)

3.6.2 Search Tree Traversal

As already mentioned, Eclat searches a prefix tree in depth first order. The change of a node to its first child consists in making a new bit matrix by intersecting the first row with all following rows. For the second child the second row is intersected with all following rows and so on. The item corresponding to the row that is intersected with the following rows thus is added to form the common prefix of the item sets processed in the corresponding child node. Of course, rows corresponding to infrequent item sets should be discarded from the constructed matrix, which can be done most conveniently if we store with each row the corresponding item identifier. Intersecting two rows can be done by a simple logical and on a fixed length integer vector if worked with a true bit matrix

3.7 RECURSIVE ELIMINATION ALGORITHM

Recursive elimination is based on a step by step elimination of items from the transaction database together with a recursive processing of transaction subsets.

STEPS:

1. Load transactions (in memory)
2. Count item frequencies
3. Delete all rare items from the transactions
4. Sort each transaction according the items frequency
5. Create recursive elimination data structure.

3.8 FP GROWTH ALGORITHM

FP-Tree frequent pattern mining is used in the development of association rule mining. FP-Tree algorithm overcomes the problem found in Apriori algorithm. By avoiding the candidate generation process and less passes over the database, FP-Tree was found to be faster than the Apriori algorithm [9]. It adopts a divide and conquer strategy. Firstly it compresses the database representing frequent items into a frequent –pattern tree or FP-tree. It retains the item set association information and compressed databases are divided into a set of conditional databases, each one associated with a frequent item. It takes the help of prefix tree representation of the given database of transactions (called FP tree), which saves considerable amount of memory for storing the transactions. An FP-Tree is a prefix tree for transactions. Every node in the tree represents one item and each path represents the set of transactions that involve with the particular item. All nodes referring to the same item are linked together in a list, so that all the transactions that containing the same item can be easily found and counted. Large databases are compressed into compact FP tree structure. FP tree structure stores necessary information about frequent item sets in a database.

4. COMPARISON TABLE

Table 1.1 shows the comparative study of various association rule mining algorithms. The classification is based on the features such as the uses, merits and demerits of each individual algorithm. It is beneficial for the user to go through the table as it will help in analyzing and concluding which algorithm will yield better results in which case. The table clearly summarizes the essential information of all the algorithms discussed in the paper. The main purpose of the table is to highlight the application of all the above stated algorithms.

Table1. Represents a comparative study of algorithms

SNO	Algorithm Name	Application	Merits	Demerits	Year	References
1.	AIS	Not frequently used, but when used is used for small problems.	1. Better than SETM. 2. Easy to use	1. Candidate sets generated on the fly. 2. Size of candidate set large.	1994	[2]
2.	SETM	Not frequently used.	1. Separates generation from counting.	1. Very large execution time. 2. Size of candidate set large.	1994	[2]
3.	Apriori	Best for closed item sets.	1. Fast 2. Less candidate sets. 3. Generates candidate sets from only those items that were found large.	1. Takes a lot of memory.	2003	[6]
4.	AprioriTID	Used for smaller problems.	1. Doesn't use whole database to count candidate sets. 2. Better than SETM. 3. Better than Apriori for small databases. 4. Time saving.	—	2013	[10]
5.	Apriori Hybrid	Used where Apriori and AprioriTID used.	Better than both Apriori and AprioriTID.	—	2013	[10]
6.	Eclat	Best used for free item sets.	1. Less memory usage. 2. Lower minimum support.	1. Apriori wins in cases where candidate sets are more.	2004	[7]
7.	Recursive Elimination	-	1. Better than Apriori in all cases.	1. Less than éclat in all cases.	2005	[8]
8.	FP Growth	Used in cases of large problems as it doesn't require generation of candidate sets.	1. Only 2 passes of dataset. 2. Compresses data set. 3. No candidate set generation required so better than éclat, Apriori.	Using tree structure creates complexity.	2003	[5] [6]

5. CONCLUSION

In this paper, algorithmic aspects of association rule mining are dealt with. From a broad variety of efficient algorithms the most important ones are compared. The algorithms are systemized and their performance is analyzed based on runtime and theoretical considerations. Despite the identified fundamental differences concerning employed strategies, runtime shown by algorithms is almost similar. The comparison table shows that the Apriori algorithm outperforms other algorithms in cases of closed item sets whereas Eclat takes the lead in free item sets. Recursive Elimination was better than Apriori in all the cases but lacked in comparison to Eclat in all the cases. FP growth displayed better performance in all the cases leaving Eclat and Apriori behind by making only 2 passes to the data sets and abolishing the concept of candidate generation. The paper would give a basic idea to the company's data mining team about the algorithm which would yield better results.

6. REFERENCES

[1] Seminar-*Reports/008/67999308-datamining-intro.pdf*

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

[3] 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.

[4] Hipp, J., G¨untzer, U., and Nakhaeizadeh, G. "Algorithms for Association Rule Mining – A General Survey and Comparison", SIGKDD Explorations ACM, JULY 2000.

[5] 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.

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

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

[8] Borgelt, C. "Finding frequent itemsets by Recursive Elimination algorithm". Workshop Open Source Data Mining Software (OSDM'05, Chicago, IL), 1-5 ACM Press, New York, NY, USA 2005.

[9] Divya, R., and Kumar, V.S. "Survey on AIS, Apriori and FP-Tree algorithm". International Journal of Computer Science and Management Research.Volume 1 Issue 2 September-2012 ISSN 2278-733X.

[10] 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.

[11] Ziauddin, Kammal, S., Khan, Z.K., and Khan, I.M., "Research on association rule mining". ACMA Volume 2, no. 1, 2012, ISSN 2167-6356. World science Publisher, United States, 2012.