Volume 4, Issue 3, March 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Survey of Various Association Rule Mining Approaches

Jyotsana Dixit
ME (CTA), SSGI
India

Abha Choubey
Associate Professor (CSE), SSGI
India

Abstract-Data mining technology has emerged as a means of identifying hidden patterns and trends from large volume of data. 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 (AR) mining is a data mining task that discovers interesting relations between variables in database. It is intended to identify strong rules discovered in databases using different measures of interestingness. This paper presents a survey of various association rule mining approaches. Many factors have been considered for making comparison and also description about various data is presented which will provide increased efficiency as well as accurate results.

Keywords-Data mining (DM), Association rule (AR), Apriori algorithm, FP growth algorithm, SETM, AIS, Genetic algorithm (GA), Particle Swarm Optimization (PSO).

I. INTRODUCTION

Data mining is automatically extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) patterns or knowledge from huge amount of data. Association rule is the most frequently implemented model of data mining. Association is mainly exploring the association among fields from enormous amount of data. There have been a lot of studies on association rule and it has been proven to be an effective method. Features of the method mainly include that the association rule can easily explain the rules generated by the data and is able to present the inter relationships among variables. However, the importance is setting the right criteria for filtering process, otherwise loose criteria can cause excessive and mixed- up results, to the contrary, if the criteria are too narrowed down, some interesting rarely-seen samples might be ignored. The data that association rule deals with are categorical data, if one intends to proceed with numerical data, performing data discretization will improve the accuracy of the rules. An association rule is an implication of the form:

$$X \rightarrow Y, \text{ where } X, Y \subset I, \text{ and } X \cap Y = \varnothing$$

There are two main terms associated with association rule mining:-
1. Support for an association rule A→B is the percentage of transactions in the database that contains AUB. i,e it is the percentage of transactions in which the item occurs.
2. Confidence for an association rule A→B is the ratio of the number of transactions that contain AUB to the number of transactions that contain A.

Mining of association rules from a database consists of finding all rules that meet the user-specified threshold support and confidence. The problem of mining association rules can be decomposed into two sub-problems [Agrawal1994] as shown in figure below:-

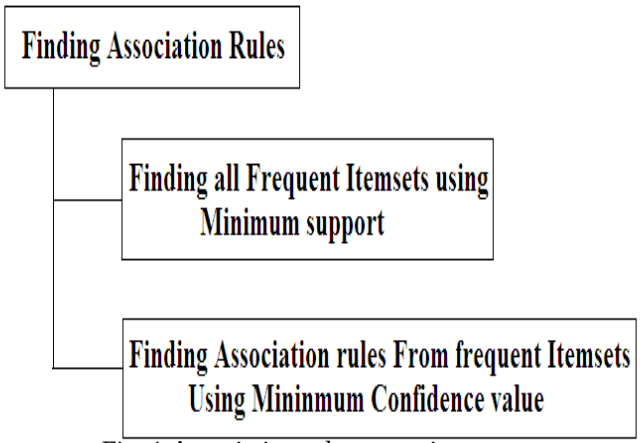

Fig. 1 Association rule generation

A. **Types of Data**

Depending on the types of data the mining techniques are applied to, data mining can be classified into different categories:-

(a) **Relational database**- Till date most data are stored in relational database and relational database is one of the biggest resources of mining objects. As it's well known that relational database is highly structured data repository, data are described by a set of attributes and stored in tables. With the use of well developed database query languages, data mining on relational database is not difficult.

(b) **Transactional database**- Transactional database refers to the set of transaction records, in most cases they are sales records. With the advent of computer and e-commerce, enormous transactional databases are available currently. Data mining on transactional database focuses on the mining of association rules, finding the association between items in the transaction records.

(c)**Spatial database**- Spatial databases typically include not only traditional data but also the location or geographic information regarding the corresponding data. Spatial association rules describe the association between one set of features and another set of features in a spatial database. For example most software companies in India are around City Bangalore, the spatial operations that used to describe the correlation can be within, near, next to, etc.

(d) **Temporal and time-series database**- In comparison from traditional transactional data, for each temporal data item the corresponding time related attribute is associated. Temporal association rules can be more useful and informative than basic association rules. Using this retailers can have a more insight view of the correlations hence can make more efficient strategies.

B. **Considerable factors for comparison**

The factors which can be used for making comparison between various association rule mining approaches are dataset,support counting ,rule generation,candidate generation,number of transactions,average size of transactions,number of items etc.

II. ASSOCIATION RULE MINING APPROACHES

This section presents a survey some of the approaches used to generate association rules. Many of the approaches had been proposed till date to solve rule mining problem. The algorithms used in these approaches are used to identify large item sets can be classified as either sequential or parallel. Often, it is assumed that the item sets are recognized and stored in lexicographic order (based on item name).This ordering provides a logical method in which item sets can be generated and counted. This is the normal approach with sequential algorithms. On the other hand, parallel algorithms focus on how to parallelize the task of finding large item sets. The following subsections will discuss the important features of previously proposed approaches along with their benefits and limitations.

A. **AIS**

The AIS algorithm was the first published algorithm developed to generate all large item sets in a transaction database [Agrawal1993]. It focused on the enhancement of databases with necessary functionality to process decision support queries. This algorithm was targeted to discover qualitative rules. This technique is limited to only one item in the consequent. That is, the association rules are in the form of $X \Rightarrow I_j \mid á$, where X is a set of items and I_j is a single item in the domain I, and á is the confidence of the rule.

AIS algorithm consists of two phases. The first phase constitutes the generation of the frequent item sets. This is followed by the generation of the confident and frequent association rules in the second phase. The drawback of the AIS algorithm is that it makes multiple passes over the database. Furthermore, it generate and counts too many candidate item sets that turn out to be small, which requires more space and waste much efforts that turned out to be useless.

Applying to sales data obtained from a large retailing company, the effectiveness of the AIS algorithm was measured in [Agrawal1993].There were a total of 46,873 customer transactions and 63 departments in the database. The algorithm was used to find if there was an association between departments in the customers' purchasing behaviour. The main problem of the AIS algorithm is that it generates too many candidates that later turn out to be small [Agrawal1994].Besides the single consequent in the rule, another drawback of the AIS algorithm is that the data structures required for maintaining large and candidate item sets were not specified [Agrawal1993].If there is a situation where a database has m items and all items appear in every transaction, there will be 2^m potentially large item sets. Therefore, this method exhibits complexity which is exponential in the order of m in the worst case.

B. **SETM**

The SETM algorithm was proposed in [Houtsma1995] and was motivated by the desire to use SQL to calculate large item sets [Srikant1996]. The SETM algorithm was motivated by the desire to use SQL to compute large item sets. Like AIS, In SETM algorithm candidate item sets are generated on the fly as the database is scanned but counted at the end of the pass. It thus generates and counts every candidate item set that the AIS algorithm generates. However, to use the standard SQL join operation for candidate generation, SETM separates candidate generation from counting. It saves a copy of the candidate item set together with the TID of the generating transaction in a sequential structure. At the end of the pass, the support count of candidate item sets is determined by sorting and

aggregating this sequential structure. Furthermore, [Sarawagi1998] mentioned that SETM is not efficient and there are no results reported on running it against a relational DBMS.

C. **FP-GROWTH**

FP-growth algorithm is proposed by Han et al.(2000). It works in a divide and conquer way. It requires two scans on the database .FP-growth first computes a list of frequent items sorted by frequency in descending order (F-List) during its first database scan. In its second scan, the database is compressed into a FP-tree. Then FP-growth starts to mine the FP-tree for each item whose support is larger than minimum support by recursively building its conditional FP-tree. The algorithm performs mining recursively on FP-tree. The problem of finding frequent item sets is converted to searching and constructing trees recursively.

The FP-Tree algorithm is of the efficient rule mining algorithms because of three reasons. First the FP-Tree is a compressed representation of the original database because only those frequent items are used to construct the tree, other irrelevant information are pruned. Also by ordering the items according to their supports the overlapping parts appear only once with different support count. Secondly this algorithm only scans the database twice. Thirdly, FP-Tree uses a divide and conquer method that considerably reduced the size of the subsequent conditional FP-Tree.

The limitation of FP-Tree is that it is difficult to be used in an interactive mining system. During the interactive mining process, users may alter the threshold of support according to the rules. However for FP-Tree the changing of support may lead to repetition of the whole mining process. Also FP-Tree is not suitable for incremental mining because with changing time new datasets may be inserted into the database which may lead to a repetition of the whole process using this algorithm.

D. **APRIORI**

The Apriori algorithm developed by [Agrawal1994] is a great achievement in the history of mining association rules. It is by far the most well-known association rule algorithm. The fundamental difference of this algorithm from AIS and SETM are the way of generating candidate item sets and the selection of candidate item sets for counting. In Apriori, the first phase is for frequent item-set generation. Frequent item-sets are detected from all-possible item-sets by using a measure called support count (SUP) and a user-defined parameter called minimum support. Support count of an item set is defined by the number of records in the database that contain all the items of that set. If the value of minimum support is too high, the number of frequent item sets generated will be less, and there by resulting in generation of a few rules. Again, if the value is too small, then almost all possible item sets will become frequent and thus a huge number of rules may be generated. Selecting better rules from them may be another problem. After detecting the frequent item-sets in the first phase, the second phase generates the rules using another user-defined parameter called minimum confidence. The Apriori generates the candidate item sets by joining the large item sets of the previous pass and deleting those subsets which are small in previous pass without considering the transactions in the database. By only considering large item sets of the previous pass, the number of candidate large item sets is considerably reduced. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation, and groups of candidates are tested against the data).The algorithm terminates when no further successful extensions are found.

Apriori Pseudo Code

C_k: Candidate itemset of size k
L_k : frequent itemset of size k
L_1 = {frequent items};
for (k = 1; L_k !=∅; k++) **do begin**
  C_{k+1} = candidates generated from L_k;
  for each transaction t in database do
    increment the count of all candidates in C_{k+1} that are contained in t
  L_{k+1} = candidates in C_{k+1} with min_support
  end
return $\cup_k L_k$;

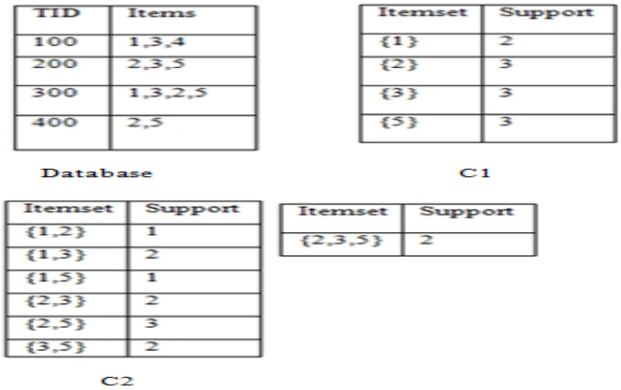

Fig. 2 Apriori Example

Instead of many advantages this algorithm has some limitations also that is if the value of minimum support is too high, the number of frequent item sets generated will be less and thereby resulting in generation of a few rules. Again if the value is too small, then almost all possible item sets will become frequent and thus a huge number of rules may be generated. Selecting better rules from them may be another problem.

E. **GENETIC ALGORITHM**

Genetic Algorithm (GA) is based on the theory of natural selection and evolution. A genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection and crossover. GAs are one of the best ways to solve a problem for which little is known. They are a very general algorithm and so will work well in any search space. The Genetic Algorithm was developed by John Holland in 1970. Standard GA apply genetic operators such selection, crossover and mutation on an initially random population in order to compute a whole generation of new strings. GA runs to generate solutions for successive generations. The probability of an individual reproducing is proportional to the goodness of the solution it represents. Hence the quality of the solutions in successive generations improves. The process is terminated when an acceptable or optimum solution is found. Since association rule mining is an optimization problem hence genetic algorithm can be used to solve this problem. The functions of genetic operators are as follows:-

1) Selection: Selection deals with the probabilistic survival of the fittest, in that, more fit chromosomes are chosen to survive. Where fitness is a comparable measure of how well a chromosome solves the problem at hand.

2) Crossover: This operation is performed by selecting a random gene along the length of the chromosomes and swapping all the genes after that point.

3) Mutation: Alters the new solutions so as to add stochasticity in the search for better solutions. This is the chance that a bit within a chromosome will be flipped (0 becomes 1, 1 becomes 0).

The work of Anandhavalli M, Suraj Kumar Sudhanshu, Ayush Kumar and Ghose M.K et. al. is to find all the possible optimized rules from given data set using genetic algorithm. The rule generated by association rule mining algorithms like priori, partition, pincer-search, incremental, border algorithm etc, does not consider negation occurrence of the attribute in them and also these rules have only one attribute in the consequent part. By using Genetic Algorithm (GAs) the system can predict the rules which contain negative attributes in the generated rules along with more than one attribute in consequent part. The major advantage of using GAs in the discovery of prediction rules is that they perform global search and its complexity is less compared to other algorithms as the genetic algorithm is based on the greedy approach. But using genetic approach has certain limitations also like the genetic algorithm cannot assure constant optimisation response times. There is no absolute assurance that a genetic algorithm will find a global optimum solution.

F. **Particle Swarm Optimization**

PSO is originally attributed to Kennedy, Eberhart and Shi (1995) and was first intended for simulating social behaviour as a stylized representation of the movement of organisms in a bird flock or fish school. Particle Swarm Optimization is an approach to problems whose solutions can be represented as a point in an n-dimensional solution space. A number of *particles* are randomly set into motion through this space. At each iteration, they observe the "fitness" of themselves and their neighbours and "emulate" successful neighbours (those whose current position represents a better solution to the problem than theirs) by moving towards them.

PSO Pseudo Code

```
For each particle
   Initialize particle
END
Do
   For each particle
      Calculate fitness value
      If the fitness value is better than the best fitness value (pBest) in history
         set current value as the new pBest
   End
   Choose the particle with the best fitness value of all the particles as the gBest
   For each particle
      Calculate particle velocity according equation (a)
      Update particle position according equation (b)
   End
While maximum iterations or minimum error criteria is not attained
```

The updation of the particle's position & velocity can be mathematically modeled according the following equation:-

$$vid = \omega * vid + \eta * xrand() * (pid - xid) + \eta * 2Rand() * (pgd - xid) \quad \text{.....(a)}$$

$$xid = xid + vid \quad \text{.....(b)}$$

where *rand()* and *Rand()* are two random numbers independently generated within the range [0,1] and η1 and η2 are two learning factors which control the influence of the social and cognitive components.

Kuo et al. (2011) have demonstrated that using the PSO algorithm one can solve the problem of association rule mining in a more efficient manner and performance was tested for large databases by applying the Foodmart2000 database. Later on Gupta (2012) used weighted PSO for finding suitable threshold values for minimum support and confidence. Hence PSO has reduced the number of rules generated without compromising the quality of rules.

III. COMPARISON OF RULE MINING APPROACHES

In this section we will be comparing the various approaches based upon several metrics like the data structure, Data base, accuracy and applicability.

S.no.	Approaches	Data structure	Database	Accuracy	Application
1.	AIS	Not precise	Transaction database	Very Less	Market-Basket analysis
2.	SETM	Not precise	SQL Compatible	Less	Market-Basket analysis
3.	FP-GROWTH	Hash table & tree	Transaction database	Less	Market-Basket analysis
4.	APRIORI	Hash table & tree	Transaction database	Less	Market-Basket analysis
5.	GA	Not precise	Transaction database	More Accurate	Optimization Problems
6.	PSO	Not precise	Transaction database	More Accurate	Optimization Problems

IV. CONCLUSION

This paper reviewed the progress of research on association rule mining implementation. The paper presents a comparison of six association rule mining approaches: AIS, SETM, FP-GROWTH, APRIORI, GA, and PSO. Out of all the six approaches PSO has been proposed for solving association rule mining problem. The proposed research work will be further carried out using Particle Swarm Optimization due to its diverse nature and many advantages as mentioned previously.

REFERENCES

[1] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami, Mining Association Rules Between Sets of Items in Large Databases, Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207-216, Washington, D.C., May 1993.

[2] Rakesh Agrawal and Ramakrishnan Srikant, Fast Algorithms for Mining Association Rules in Large Databases, Proceedings of the Twentieth International Conference on Very Large Databases, pp. 487-499, Santiago, Chile, 1994.

[3] Ming-Syan Chen, Jiawei Han and Philip S. Yu, Data Mining: An Overview from a Database Perspective, IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No. 6, pp. 866-883, 1996.

[4] David W. Cheung, Jiawei Han, Vincent T. Ng and C. Y. Wong, Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique, Proceedings of the Twelfth International Conference on Data Engineering, February 26 - March 1, 1996, New Orleans, Louisiana, pp. 106-114.

[5] C.Gyorodi, R.Gyorodi Mining association rules in large databases, Proceedings of the Oradea EMES02:45-50,*2002*.

[6] Jiawei Han and Yongjian Fu, Discovery of Multiple-Level Association Rules from Large Databases, Proceedings of the 21nd International Conference on Very Large Databases, pp. 420-431, Zurich, Swizerland, 1995.

[7] Eui-Hong Han, George Karypis, and Vipin Kumar, Scalable Parallel Data Mining For Association Rules, Proceedings of the ACM SIGMOD Conference, pp. 277-288, 1997.

[8] Komal Khurana, Simple Sharma,A Comparative analysis of association rules mining algorithms,*IJSRP,VOLUME 3,ISSUE 5*,2013.

[9] Margaret H.Dunham, Yongqiao Xiao,Le gruenwald,Zahid hussain, A Survey on association rule mining,Texas.

[10] M.Saggar. Agrawal.A.K. A.Lad. 2004 .Optimization of association rule mining using improved genetic algorithms. *In Proceeding of the IEEE International Conference on Systems Manand Cybernetics*.vol.4.pp.3725–3729.

[10] Sarawagi Sunita, Thomas Shiby, and Agrawal Rakesh, Integrating Mining with Relational Database Systems: Alternatives and Implications, Proceedings ACM SIGMOD International Conference on Management of Data, SIGMOD 1998, June 2-4, 1998, Seattle, Washington, USA .

[11] Sarath.K.N.V.D. Vadlaman. 2013. Association rule mining using binary particle swarm optimization. *Engineering Applications of Artificial Intelligence*. 1832 – 1840.

[12] Shing .Wang.Her. Chang. Yeh. Wei. Chiao. Huang. Pei. Wen.Wei.2009. Using association rules and particle swarm optimization approach for part change. *Expert Systems with Applications* 36 (2009). 8178–8184.

[12] Ramakrishnan Srikant and Rakesh Agrawal, Mining Generalized Association Rules, Proceedings of the 21nd International Conference on Very Large Databases, pp. 407-419, Zurich, Swizerland, 1995.

[13] Mohammed Javeed Zaki, Parallel and Distributed Association Mining: A Survey, IEEE Concurrency, October-December 1999.

