



COMPARISON OF ASSOCIATION RULE MINING ALGORITHMS

¹. K. Shankar, ².T. Dinesh Kumar

^{1&2}. Asst.Professor Department of Computer Science,

Srimad Andavan Arts and Science College (Autonomous), Trichy -5.

1.Dineshsaran10@gmail.com

ABSTRACT

In General review of four various association rule mining algorithms Apriori, AprioriTid, Apriori hybrid and tertius algorithms and their drawbacks which would be helpful to find new solution for the problems that are found in these algorithms is presented in a comparison between various association mining algorithms. The Association rule mining is the one of the most important technique of the data mining. Its aim is to extract interesting correlations, frequent patterns and association among set of items in the transaction database.

KEYWORDS:

Data mining, Association rule algorithms, Apriori, AprioriTid, Apriori hybrid and Tertius algorithms

INTRODUCTION

The science of extracting useful information from large data sets or databases is named as data mining. Though data mining concepts have an extensive history, the term “**Data Mining**”, is introduced relatively new, in mid 90’s. Data mining covers areas of statistics, machine learning, data management and databases, pattern recognition, artificial intelligence, and other areas.

ASSOCIATION RULE MINING

In data mining, association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using different measures of interestingness. Based. A typical and widely-used example of association rule mining is Market Basket Analysis. The problem is to generate all association rules that have support and confidence greater than the user-specified minimum support and minimum confidence.

Definition: Association Rule

• **Association Rule**

- An implication expression of the form $X \rightarrow Y$, where X and Y are itemsets
- Example:
 $\{Milk, Diaper\} \rightarrow \{Beer\}$

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

• **Rule Evaluation Metrics**

- Support (s)
 - Fraction of transactions that contain both X and Y
- Confidence (c)
 - Measures how often items in Y appear in transactions that contain X

Example:

$$\{Milk, Diaper\} \Rightarrow Beer$$

$$s = \frac{\sigma(Milk, Diaper, Beer)}{|T|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(Milk, Diaper, Beer)}{\sigma(Milk, Diaper)} = \frac{2}{3} = 0.67$$

Support(S)

Support(S) of an association rule is defined as the percentage/fraction of records that contain AUB to the total number of records in the database. Suppose the support of an item is 0.1%, it means only 0.1percent of the transaction contain purchasing of this item.

$$\text{Support (AB)} = \text{Support count of (AB)} / \text{Total number of transaction in D}$$

Confidence(C)

Confidence(C) of an association rule is defined as the percentage/fraction of the number of transactions that contain AUB to the total number of records that contain A. Confidence is a measure of strength of the association rules, suppose the confidence of the association rule $A \Rightarrow B$ is 80%, it means that 80% of the transactions that contain A also contain B together.

$$\text{Confidence (A|B)} = \text{Support (AB)} / \text{Support (A)}$$

A. Association Rules Goals

- Find all sets of items (item-sets) that have support (number of transactions) greater than the minimum support (large item-sets).
- Use the large item-sets to generate the desired rules that have confidence greater than the minimum confidence.

B. General AR Algorithm

- In the first pass, the support of each individual item is counted, and the large ones are determined.
- In each subsequent pass, the large item-sets determined in the previous pass is used to generate new item-sets called candidate item-sets.
- The support of each candidate item-setis counted, and the large ones are determined.
- This process continues until no new large item-sets are found.

APRIORI ALGORITHM

The **APRIORI** algorithm finds the frequent set L in the database D. It makes use of the downward closure property. The algorithm is a bottom search, moving upward level-wise in the lattice. However, before reading the database at every level, it prunes many of the sets which are unlikely to be frequent sets, thus saving any extra efforts.

Candidate Generation : Given the set of all frequent (k-1) item-sets. We want to generate superset of the set of all frequent k-item-sets. The intuition behind the aprioricandidates generation procedure is that if an item-set A has minimum support, so do all subsets of A. after all the (l+1)-candidate sequences have been generated, a new scan of the transactions is started (they are read one-by-one) and the support of these new candidates is determined.

Pruning : The pruning step eliminates the extensions of (k-1) item-sets which are not found to be frequent, from being considered for counting support. For each transaction t, the algorithm checks which candidates are contained in t and after the last transaction are processed; those with support less than the minimum support are discarded.

Discovering Large Item-sets

- Multiple passes over the data
- First pass – count the support of individual items.
- Subsequent pass
 1. Generate Candidates using previous pass's large item-set.
 2. Go over the data and check the actual support of the candidates.
- Stop when no new large item-sets are found.

Any subset of large item-set is large, therefore to find large k-item-set

- Create candidates by combining large k-1 item-sets.
- Delete those that contain any subset that is not large[1].

AR –Statement of Problem

- $I = \{ i_1, i_2, \dots, i_m \}$ is a set of items.
- D is a set of transactions T.
- Each transaction T is a set of items.
- TID is a unique identifier that is associated with each transaction.

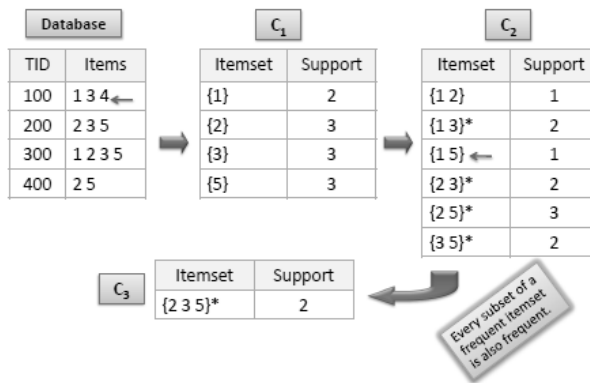
DRAWBACKS

The main drawbacks of Apriori algorithm are

- a) It takes more time, space and memory for candidate generation process.
- b) To generate the candidate set it requires multiple scan over the database.

APRIORITID ALGORITHM

- Apriori performs better than AprioriTid in the initial passes but in the later passes AprioriTid has better performance than Apriori. Due to this reason we can use another algorithm called Apriori Hybrid algorithm.
- In which Apriori is used in the initial passes but we switch to AprioriTid in the later passes. The switch takes time, but it is still better in most cases.
- Estimate the size of C'



DRAWBACKS

- a) An extra cost is incurred when shifting from Apriori to AprioriTid.
- b) Suppose at the end of K th pass we decide to switch from Apriori to AprioriTid. Then in the (k+1) pass, after having generated the candidate sets we also have to add the Tids to C'^{k+1}.

TERTIUS ALGORITHM

- This algorithm finds the rule according to the confirmation measures. It uses first order logic representation. It includes various option like class Index, classification, confirmation Threshold, confirmation Values, frequency Threshold, horn Clauses, missing Values, negation, noise Threshold, number Literals, repeat Literals, roc Analysis, values Output etc.

DRAWBACK

- a) Tertius is its relatively long runtime, which is largely dependent on the number of literals in the rules. Increasing the allowed number of literals increases the runtime exponentially, so we want to keep the maximum to three. Even with an allowed maximum of three literals, the

runtime is still quite long - running Tertius can take up to several hours for some of our larger tests.

Comparison of Apriori, AprioriTid, AprioriHybrid and Tertius

Sl.no	Properties	Apriori	Aprioritid	AprioriHybrid	Tertius
01.	Candidate Generation.	Candidate Item – sets are Generated using only the large item sets of the previous pass without considering the transactions in the database.	The database is not used at all for counting the support of candidate item-set after the first pass.	Hybrid algorithm can be designed that uses apriori in the initial passes and switches to apriori tid in the latter passes .	The tertius algorithm builds a rules out of the attribute pair value in the training data .
02.	Methodology used	Use join and prune step.	Use join and prune as well as tids .	Users apriori and apriori tid.	Use first order logic representation .
03.	Database scan	Multiple scan all over the database .	Use the database only once .	Users apriori and apriori tid.	It is depends on the number of rules and regulations.
04.	Memory usage	It takes more space and memory for candidate generation process .	Apriori-id needs memory for large and additional cost is incurred if it cannot completely fit in to the memory.	An extra cost is incurred when shiffiting form apriori to apriori tid.	When the program runs out of memory, the best rules found and printed, and a message indicate that the search was interrupted.
05.	Execution time	It take more execution time for candidate generation process.	For small problem its better than apriori but it takes more time for large problem	It's better than apriori and apriori tid.	It's relatively long run time. Tertius can take up to several hours for some of our larger tests.

CONCLUSION

In this Article provided an overview on four different association rule mining algorithms Apriori, AprioriTid, Apriori hybrid and tertius algorithms and their drawbacks which would be helpful to find new solution for the Problems found in these algorithms and also presents a comparison between various association mining algorithms.

REFERENCES

- [1] Introduction to Data mining with case studies – Gupta GK – PHI ' 2006
- [2] R. Agrawal, T. Imielinski, and A. Swami, “Mining association rules between sets of items in large databases”. In Proc. of the ACM SIGMOD international Conference on Management of Data - SIGMOD '93. p. 207 Washington, D.C., May 1993.
- [3] J. Han, M. Kamber, “Data Mining Concepts and Techniques”, Morgan Kaufmann Publishers, San Francisco, USA, 2001, ISBN 1558604898.
- [4] R. Agrawal, T. Imielinski, and A. Swami, “ Database mining: A performance perspective,” IEEE Transactions on Knowledge and Data Engineering, 5(6):914{925, December 1993. Special Issue on Learning and Discovery in Knowledge Based Databases.
- [5] Flach, P. A., &Lachiche, N. (2001), “Confirmation-Guided Discovery of First-Order Rules with Tertius,”Mach. Learn., 42(1-2), 61-95.
- [6] J. Han, Y. Cai, and N. Cercone, “ Knowledge discovery in databases: An attribute oriented approach,” In Proc. of the VLDB Conference, pages 547{559, Vancouver, British Columbia, Canada, 1992.
- [7] Margaret H. Dunham, “Data Mining Introductory and Advanced Topics,” Publisher, Person education india, 2006, Pages 328
- [8] http://en.wikipedia.org/wiki/Association_rule_learning