iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** XI **Month of publication:** November 2017

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue XI November 2017- Available at www.ijraset.com

A Comparative Analysis of Frequent Pattern Mining Algorithms

Heli Vyas[1], Dr. Sanjay M. Shah[2]
Assistant Professor[1], Director[2], Chaudhari Technical Institute, Gandhinagar, Kadi Sarva Vishva Vidhyalaya, Kadi[2]

Abstract: Data mining research is a vast field; several researchers have done in depth research on mining of frequent patterns in transaction databases, time-series databases, and many other kinds of databases. As a matter of fact, candidate set generation-and-test methodology in most of the previous researches is found to be similar to Apriority. However, this can be a costly affair especially when long patterns exist. In this study, efficiency of mining is achieved with a new technique where a large database is compressed into a highly condensed, much smaller data structure, which helps in avoiding costly reruns. Our performance study shows that the new algorithm is efficient and scalable for mining both long and short frequent patterns, and also a bit faster than Apriority and some of the new found frequent pattern mining methods.
Keywords: Pattern mining, proposed algorithm, comparative analysis, large database

I. INTRODUCTION

Frequent pattern mining plays an essential role in mining associations, correlations, sequential patterns, episodes, multi-dimensional patterns, max-patterns, partial periodicity, emerging patterns, and many other important data mining tasks.
Due to its popularity and robust design all most all of the studies of the past have adopted an approach similar to Apriority. Here the central idea is to iteratively generate the set of candidate patterns of length (k+1) from the set of frequent patterns of length k (for k $\geq$ 1), and check their corresponding occurrence frequencies in the database. There are some shortfalls in this algorithm like frequently access database, problem in large database and time consuming.
After some careful examinations, we could relate the problem of Apriority algorithm to the candidate set generation and test i.e. if one can bypass the generation of a huge set of candidate patterns, the performance of frequent pattern mining mechanism can be substantially improved. The solution to this, as we have proposed in our algorithm, is to arrange the data arrange in a specific format so a repeated access of database is not required.

II. APRIORI ALGORITHM: DESIGN AND CONSTRUCTION

Apriority is a algorithm for finding frequent item sets using candidate generation. Apriori is the most classical and famous algorithm for mining frequent patterns [1].
The algorithm works on specific attributes and employs bottom up strategy, wherever candidate generation are extended one item at a time and tested against the data. Apriori algorithm generates frequent item sets. If item satisfies a specific minimum support and minimum confidence then it's thought about as a frequent item.

A. Steps of Apriori algorithm

1) Find all large 1-itemsets. (scan the DB, and count the number of times that item appears in a basket)
2) Initialize count of candidate with zero
3) Find out frequency of each candidate.
4) Discard candidates whose frequency count < minimum support.
5) Repeat process until large-item set is empty.

B. Like every coin has two sides, an Apriori-like algorithm has a few downsides too, like

1) It can get way too expensive to handle a large data set, due to the manner in which it works. For example, if a data contains 10^4 frequent 1-itemsets, the Apriori algorithm will need to generate more than 10^7 length-2 candidates and accumulate and test their occurrence frequencies, no to talk of data sets that run in millions.
2) Due to its iterative nature, it becomes tedious and time consuming to repeatedly scan the database and check a large set of candidates by pattern matching, which is especially the case in mining of long patterns [2].

©IJRASET (UGC Approved Journal): All Rights are Reserved

C. Proposed algorithm: Design and Construction

Our proposed algorithm will arrange the data in a specific pattern and on the basis of this pattern it will decide whether to save data on local server or push it on to the cloud. But unlike Apriori here the mechanism will be much less iterative and repetitive. This Pattern generation process is the most important part in our proposed algorithm. Here are the steps of the proposed algorithm.

Steps to generate pattern:

1) Arrange candidates of each transaction in either alphabetical or ascending order of their occurrence.
2) Find out frequency of each candidate.
3) Discard candidates whose frequency count < minimum support.
4) Rearrange candidates in ascending order (candidates whose frequency count is maximum will come first) in each transaction.
5) Find frequency of 2-3 combination of candidates.
6) Discard combination candidates whose frequency count < minimum support.

This algorithm does not need to access the dataset every time. It first arranges the candidates in ascending or descending order of their occurrence and then finds frequency of each candidate. Now it singles out those candidates whose frequency is less than the predefined or user defined minimum support. Such candidates are discarded from the database at this level itself so the next process is less tedious and consumes less time. Remaining candidates again get rearranged in ascending order of frequency. Once again it makesa combination of candidates and finds their frequency. If frequency is less than minimum support they are discarded, as in the first step. This process is repeated till the user will getsan answer.

III. COMPARATIVE ANALYSIS

The apriori algorithm generates candidate item sets by repeatedly accessing dataset while the proposed algorithm does not need to access data set repeatedly.

The apriori algorithm has the following bottlenecks: i.) Difficult to handle huge number of candidate item sets also the candidate generation can be very costly with the increasing size of database. ii.) It is tedious to repeatedly scan huge databases.[2] iii)In case of large dataset, Apriori algorithm produces large number of candidate item sets. The algorithm scans database repeatedly for searching frequent item sets, so more the number of scans more is the time and resource requirement, making it inefficient in large datasets. [3]

The proposed algorithm functions in a different way. It works by generating a combination of candidates and finding their relative frequency in the first scan, hence it doesn't need to scan the database multiple times. So processing time and cost is also reduced. The main drawback of the Apriori algorithm is eliminated with this approach in the proposed algorithm.

The table depicts the comparison of two algorithms studied here based on different parameters

Parameters	Apriori	Proposed algorithm
Large datasets	Difficult to process	Easy to process
Number of scans	Database is scanned at every level	Lesser number of database scans
Time	Execution time is large	Execution time is smaller

IV. CONCLUSION

In recent times the size of most databases has been increasing by leaps and bounds. Therefore we require a system to handle such huge amounts of data. In this paper, algorithmic aspects of association mining rules are dealt with. From a broad variety of efficient algorithms the most important ones are compared. The algorithms are systemized and their performance is analysed based on runtime and theoretical considerations. The comparison table shows that the Apriori algorithm outperforms other algorithms in cases of closed item sets whereas proposed algorithm displayed better performance in all the cases. The undermining goal of this frequent item set mining process helps to form association rules for future use.

©IJRASET (UGC Approved Journal): All Rights are Reserved

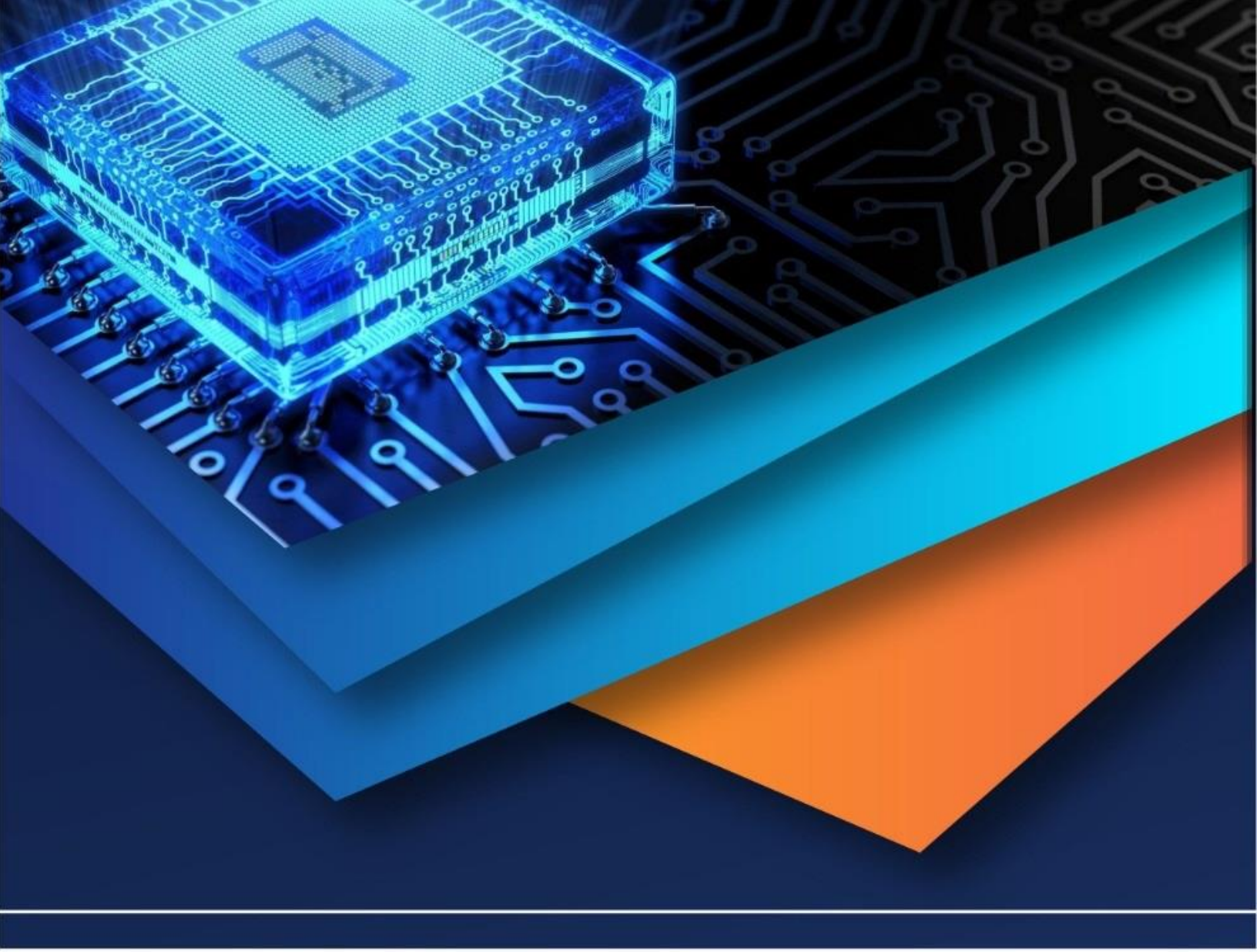
REFERENCES

[1] RupinderKaur, Rajeev Kumar Bedi, Sunil Kumar Gupta," REVIEW OF ASSOCIATION RULE MINING", International Journal of Advanced Technology & Engineering Research, ISSN No: 2250-3536 Volume 4, Issue 2, March 2014.

[2] Ahilandeeswari.G.,Dr. R. ManickaChezian," A comparative study of Frequent pattern mining Algorithms: Apriori and FP Growth on Apache Hadoop", International Journal of Innovations & Advancement in Computer Science, ISSN 2347 – 8616, March 2015.

[3] SuhaniNagpal, "Improved Apriori Algorithm Using Logrithmic Decoding and Pruning", In: Proceding of International Journal of Engineering Research and Applications, ISSN 2248-9622, Vol. 2, Issue 3, pp. 2569-2572, May-June 2012.

©IJRASET (UGC Approved Journal): All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)