**International Journal of Technical Research and Applications e-**ISSN: 2320-8163,
www.ijtra.com Volume 3, Issue 3 (May-June 2015), PP. 238-240

# IMPROVED APRIORI ALGORITHM FOR ASSOCIATION RULES

**Shikha Bhardwaj[1], Preeti Chhikara[2], Satender Vinayak[2], Nishant Pai[2], Kuldeep Meena[2]**
[1]Faculty, Department of Computer Science & Engineering,
Bharati Vidyapeeth's College of Engineering, New Delhi
[2]Final Year Students, Department of Computer Science & Engineering,
Bharati Vidyapeeth's College of Engineering, New Delhi
shikha.bhardwaj@bharatividyapeeth.edu

**Abstract— Association rules are the main techniques to determine the frequent item set in data mining. Apriori algorithm is the classic algorithm of association rules, which enumerate all of the frequent item sets. If database is large, it takes too much time to scan the database. The improved algorithm is verified, the results show that the improved algorithm is reasonable and effective, and can extract more valuable information.**

**Keywords- Apriori, Improved Apriori, Association Rule, Data Mining**

## I. INTRODUCTION

Data mining also known as Knowledge Discovery in Database (KDD). The purpose of data mining is to abstract interesting knowledge from the large database.

Apriori is a classic algorithm for learning association rules. Apriori is designed to operate on databases containing transactions. The Apriori algorithm is used for association rule mining. Apriori is the best-known basic algorithm for mining frequent item sets in a set of transactions. Apriori algorithm represents the candidate generation approach. It generates candidate (k+1) item sets based on frequent k-item sets.

The Apriori-based algorithms finds frequent item sets based upon an iterative bottom-up approach to generate candidate item sets. Apriori is a Breadth First Search Algorithm (BFS). Now, a method to obtain frequent item-set by using a different approach to classical Apriori algorithm, by making a matrix of given example by considering row as transactions and columns as items. By reducing rows and columns from matrix, we will finally produce a frequent item set without scanning database repeatedly. So this method will increase the efficiency and reduce the time to generate the frequent item-sets.

In Apriori, each set of data has a number of items and is called a transaction. The output of Apriori is sets of rules that tell us how often items are contained in sets of data.
Frequent item set mining and association rule induction are powerful methods for so-called market basket analysis, which aims at finding regularities in the shopping behavior of customers of supermarkets, mail-order companies, online shops etc.This algorithm only needs to scan the database one time and also greatly reduces the number of candidates of frequent item sets.

## II. RELATED WORK

In year 2010, Yongge Shi, Yiqun Zhou have purposed some association rules to increase the efficiency of the Apriori algorithm which can improve the speed of data mining effectively, enhance the ability of ADSL line quality's analysis and solving[1].

In year 2010, Libing Wu, KuiGong, Fuliang Guo, XiaohuaGe have given a C# code which achieves the improved algorithm confirmed by many experiments, this algorithm is better than traditional algorithms in time consumed and reduces the frequency with which we scan the database and reduce the unnecessary duplication of effort[2].

Rehab H. Alwa and Anasuya V. Patil (2013) described a novel approach to improve the Apriori algorithm through the creation of Matrix- File [3].

Mohammed Al- Maolegi and Bassam Arkok (2013) indicated the limitation of the original Apriori algorithm of wasting time for scanning the whole database searching on the frequent item-sets, and presented an improvement on Apriori by reducing that wasted time depending on scanning only some transactions[4].

## III. METHODS TO IMPROVE APRIORI EFFICIENCY

- Hash-based item set counting: A k-item set whose corresponding hashing bucket count is below the threshold cannot be frequent.
- Transaction reduction: A transaction that does not contain any frequent k-item set is useless in subsequent scans.
- Partitioning: Any item set that is potentially frequent in DB must be frequent in at least one of the partitions of DB.
- Sampling: mining on a subset of given data, lower support threshold + a method to determine the completeness.
- Dynamic itemset counting: add new candidate item sets only when all of their subsets are estimated to be frequent.

## IV. PSEUDO CODE

### A. *Apriori Algorithm*

The name of algorithm is based on the fact that the algorithm uses prior knowledge of frequent item set properties. It uses a breadth-first search strategy for counting the support of item sets. It also 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. Apriori algorithm is the original algorithm of Boolean Association rules of mining frequent item sets, raised by R.

Apriori first scans the database and searches for frequent item sets of size 1 by accumulating the count for each item and collecting those that satisfy the minimum support requirement.

- Join Step: $C_k$ is generated by joining $L_{k-1}$with itself
- Prune Step: Any (k-1)-item set that is not frequent cannot be a subset of a frequent k-item set
- Pseudo-code:

```
$C_k$: Candidate item set of size k
$L_k$: frequent item set 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** $C_kL_k$;
```

Our hash based Apriori implementation, uses a data structure that directly represents a hash table. This algorithm proposes overcoming some of the weaknesses of the Apriori algorithm by reducing the number of candidate k-itemsets. In particular the 2-itemsets, since that is the key to improving performance. This algorithm uses a hash based technique to reduce the number of candidate itemsets in the first pass. It is claimed that the number of itemsets in C2 generated using hashing can be reduced, so that the scan required and efficiency become better.

For example, when scanning each transaction in the database to generate the frequent 1-itemsets,L1, from the candidate 1-itemsets in C1, we can generate all of the 2-itemsets for each transaction, hash map them into the different buckets of a hash table structure, and increase the corresponding bucket counts. A 2-itemset whose corresponding bucket count in the hash table is below the support threshold cannot be frequent and thus should be removed from the candidate set. Such a hash based Apriori may substantially reduce the number of the candidate k-itemsets examined.

Steps:-

1. Scan all the transaction. Create possible 2-itemsets.
2. Let the Hash table of size 6.
3. For each bucket assign an candidate pairs using the ASCII values of the itemsets.
4. Each bucket has a count, which is increased by 1 each item, an item set is hashed to that bucket.
5. If the bucket count is equal or above the minimum support count, the bit vector is set to 1. Otherwise it is set to 0.
6. The candidate pairs that hash to locations where the bit vector bit is not set are removed.
7. Modify the transaction database to include only these candidate pairs.

In this, each transaction counting all the 1-itemsets. At the same time all the possible 2-itemsets in the current transaction are hashed to a hash table. It uses a hash table to reduce the number if candidate itemsets. When the support count is established the steps the frequent itemsets. It generates the candidate itemsets as like the Apriori algorithm.

## V. METHODOLOGY

To improve the performance of Apriori algorithm we are using the Hashing Data structure. We report experimental results on accident dataset. We have taken accident database as a Text file. In this data set, the average maximal potentially frequent itemset size is set to 14, while the number of transactions in the dataset is set to 25. Apriori and Hash based Apriori were executed for different minimum support level to generate the candidate 2-itemsets. The performance of Apriori and Hash based Apriori algorithms are evaluated for different minimum support levels.

## VI. SIMULATION RESULTS

The following table presents the test results of the implementations of Apriori and the Hash based Apriori on the dataset of supermarket for different minimum support level.

**Table 1. Memory Usage of Apriori and Hash Based Apriori**

| Miniumm support level | **Size of candidate 2-itemsets** | |
|---|---|---|
| | **Apriori** | **Hash Based Apriori** |
| 1 | 126 | 41 |
| 2 | 68 | 25 |
| 3 | 35 | 19 |
| 4 | 35 | 15 |
| 5 | 18 | 6 |
| 6 | 11 | 6 |

As a result, when comparing with Apriori algorithm the size of candidate 2 Itemsets of Hash based Apriori algorithm is reduced. For minimum support level of 1, the size of candidate 2-Itemsets is 126 while using Apriori. But it is reduced to 41 while using Hash based Apriori.

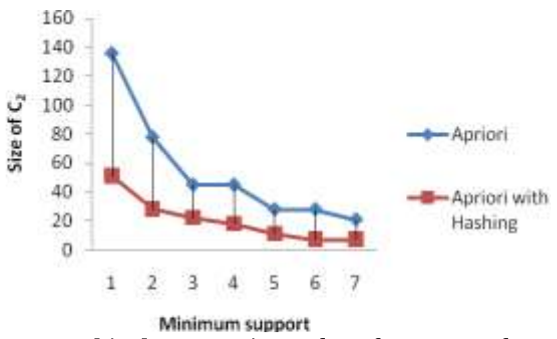

**Fig. 3. Graphical Comparison of Performance of Apriori and Hash Based Apriori**

In these graphs, we see that the memory usage of candidate 2-itemset for both algorithms increases exponentially as the minimum support is reduced. Applying Hashing data structure in Apriori reduce the size of candidate 2-itemsets when comparing with Apriori.

## VII. CONCLUSION

Determining frequent objects is one of the most important fields in data mining. This algorithm can achieve a smaller memory usage than the Apriori algorithm. It is well known that the way candidates are defined has great effect on running time and memory need. Hash based Apriori is most efficient for generating the frequent itemset than Apriori.

## VIII. ACKNOWLEDGMENT

The authors would like to thank everyone, just everyone!

## REFERENCES

[1] J. Han and M. Kamber, *Conception and Technology of Data Mining*, Beijing: China Machine Press, 2007.

[2] S. Rao, R. Gupta, “Implementing Improved Algorithm Over APRIORI Data Mining Association Rule Algorithm”, *International Journal of Computer Science And Technology*, pp. 489-493, Mar. 2012

[3] “Data Mining - concepts and techniques” by Jiawei Han and MichelineKamber

[4] “Introduction to data mining and its applications” S. Sumathi, S. N. Sivanandam.

[5] P. Keleher, A. L. Cox, and W. Zwaenepoel, ―Lazy Release Consistency for Software Distributed Shared Memory‖. In Proc. of the 19th Annual Int‘l Symposium on Computer Architecture,1992, pp. 13-21.

[6] J. Han and M. Kamber,"Data mining concepts and techniques‖, Elsevier, 2nd Edition. Chapter 5.

[7] C. Lucchese, S. Orlando, R. Perego, and F. Silvestri, ―Webdocs: a real-life huge transactional dataset‖. In Jr. et. al.