2nd International Conference on "Innovative Trends in Science, Engineering and Management"
YMCA, Connaught Place, New Delhi ICITSEM-16
05 November 2016, www.conferenceworld.in ISBN : 978-93-86171-10-8

# ANOTHER APPROACH TO IMPROVING THE APRIORI ALGORITHM

**Sarthak Arora[1], Dr. Dharmveer Singh Rajput[2]**

*[1,2]Jaypee Institute of Information Technology, Noida-62, (India)*

## ABSTRACT

*Apriori algorithm is a well-known algorithm that is used to perform association rule mining. Apriori, all though extremely popular and efficient has certain shortcomings. One of these is that it involves the repeated scanning of data. Moreover it also tends to generate item sets that are not required. Another drawback of the Apriori algorithm is that it becomes increasingly difficult to identify unusual events. In this paper we try to improve the Apriori algorithm. We do so by performing an additional pruning step at the initial level.*

***Keywords: Data Mining; Apriori Algorithm; Prinning; Data; Big Data***

## I. INTRODUCTION

Several methods have been proposed to try and improve the Apriori algorithm. There are many techniques by which this is possible. Some of the methods that can be used to improve the Apriori algorithm are portioning, sampling, transaction reduction etc. Hash based item set counting as well as dynamic item set counting can also be used for this purpose [2]. The technique / method that we will be following in this paper is pruning.

## II. BACKGROUND

One approach to pruning is that, before the candidate item sets Ck are produced, Lk-1 is further pruned, the no of times all items occurred in Lk-1 are counted, and then those item sets which have this number less than k-1 in Lk-1 are deleted, In this way, the number of connecting items sets decreases, so that the number of candidate items will ultimately reduce [1].

Our approach is that at the initial level of generating candidate sets we will take only those item sets whose support is greater than the minimum support. This is different from Apriori in the sense that we are not considering those itemsets that have support equal to the minimum support. However this approach is only followed while generating L1 from C1.

## III. PROPOSED WORK

L1 = {frequent items} such that candidates in C1 with (> min_support) greater than min_support

For (k= 1; Lk !=∅; k++) do begin

Ck+1= candidates generated from LK;

For each transaction in database do

Increment the count of all candidates in Ck+1 that are contained in t

Lk+1= candidates in Ck+1with min_support end return ∪KLK;

**IV. EXAMPLE ON DATA SET**

Let us consider the following data set

| Transaction Id | Item sets |
|---|---|
| T1 | 1,2,3,5,6 |
| T2 | 1,2,3 |
| T3 | 1,4,5,7 |
| T4 | 2,3,4,7 |
| T5 | 1,2,4,5 |
| T6 | 1,2,3,4,5,7 |
| T7 | 1,3,6 |
| T8 | 1,2,3 |

| Itemset | Support |
|---|---|
| 1 | 7 |
| 2 | 6 |
| 3 | 6 |
| 4 | 4 |
| 5 | 4 |
| 6 | 2 |
| 7 | 3 |

Applying Apriori Algorithm

On applying apriori we'll include those itemsets in L1 whose support is greater than or equal to minimum support.

**L1**

| Itemset | Support |
|---|---|
| 1 | 7 |
| 2 | 6 |
| 3 | 6 |
| 4 | 4 |
| 5 | 4 |

**C2**

| Itemset | Support |
|---|---|
| 1,2 | 5 |
| 1,3 | 5 |
| 1,4 | 3 |
| 1,5 | 4 |
| 2,3 | 5 |
| 2,4 | 3 |
| 2,5 | 3 |
| 3,4 | 2 |
| 3,5 | 2 |
| 4,5 | 3 |

**L2**

| Itemset | Support |
|---|---|
| 1,2 | 5 |
| 1,3 | 5 |
| 1,5 | 4 |
| 2,3 | 5 |

**C3**

| Itemset | Support |
|---|---|
| 1,2,3 | 4 |

**L3**

| Itemset | Support |
|---|---|
| 1,2,3 | 4 |

Applying our approach on the same data set:

**L1**

| Itemset | Support |
|---|---|
| 1 | 7 |
| 2 | 6 |
| 3 | 6 |

**C2**

| Itemset | Support |
|---|---|
| 1,2 | 5 |
| 1,3 | 5 |
| 2,3 | 5 |

**L2**

| Itemset | Support |
|---|---|
| 1,2 | 5 |
| 1,3 | 5 |
| 2,3 | 5 |

**C3**

| Itemset | Support |
|---|---|
| 1,2,3 | 4 |

**L3**

| Itemset | Support |
|---|---|
| 1,2,3 | 4 |

## V. RESULT

On applying our algorithm to the data set we observed that the number of scans of the database reduced significantly. Because of an additional pruning condition, the no of item sets generated in L1 were less than those generated by the Apriori algorithm. As a result, less transaction pairs were generated in the consequent candidate set table c2. Less transaction pairs implies reduced scans of the database as compared to the original Apriori algorithm. However this pruning is only done while generating L1.

## VI. CONCLUSION

In this paper, we have introduced a new approach to improving the Apriori algorithm. We have applied it on multiple data sets. In this paper we have presented our algorithm on an example data set and compared the results with those of the original Apriori algorithm.

## VII. ACKNOWLEDGMENT

This research paper is made possible through the help and support from everyone, including: parents, teachers, family, friends, and in essence, all sentient beings. Especially, please allow us to dedicate our acknowledgment of gratitude towards Dr. Dharmveer Singh Rajput for his support and encouragement.

## REFERENCES

[1] International Journal of Computer and Communication Engineering, Vol. 2, No. 1, January 2013
[2] Research of an Improved Apriori Algorithm in Data Mining Association Rules
[3] Professor Anita Wasilewska www3.cs.stonybrook.edu/~cse634/lecture_notes/07apriori.pdf