IJCSE International Journal of Computer Sciences and Engineering Open Access

Review Paper Volume-6, Issue-4 E-ISSN: 2347-2693

Implementation of Apriori Algorithm in E-Commerce Application

Karan Kapoor[1*], Shana Parveen[2], Praveen Gupta [3], Abhay Gupta [4]

[1] Department of Information Technology, PSIT Kanpur, Dr. A.P.J. Abdul Kalam Technical Universty, Lucknow, India
[2] Department of Information Technology, PSIT Kanpur, Dr. A.P.J. Abdul Kalam Technical Universty, Lucknow, India
[3] Department of Information Technology, PSIT Kanpur, Dr. A.P.J. Abdul Kalam Technical Universty, Lucknow, India
[4] Department of Information Technology, PSIT Kanpur, Dr. A.P.J. Abdul Kalam Technical Universty, Lucknow, India

**Corresponding Author: kapoor.karan1708@gmail.com, Tel.: +91-9044505101*

Available online at: www.ijcseonline.org

Received: 23/Mar/2018, Revised: 31/Mar/2018, Accepted: 21/Apr/2018, Published: 30/Apr/2018

Abstract— Apriori algorithm is an influential algorithm for mining frequent item sets for Boolean association rules. It uses a bottom up approach where frequent subset is extended one item at a time. It starts with an analysis of e- commerce data and the driving forces behind the success of data mining in e-commerce.

Keywords—Data Mining, Data Clustering, Data Classification.

I. Introduction

Data Mining is a field of computer science that is concerned with extracting useful information from varied sources. In an era where information has become the inherent necessity of human beings, its increased relevance and usefulness has taken focus as need of the hour. Data mining is a kind of process for decision support [1]. It gets the potential and useful information and acknowledges from practical application data which is large, incomplete, noisy, ambiguous and random. Data mining relates to extracting a large of data from database, transforming, analyzing and modelling handling these data, and withdrawing the critical data to aid decision making. The most important part of this association rule mining is the mining of item sets that are frequent. Apriori algorithm is a widely used technique in order to find those frequent combinations of item sets. Mining association rules is an important issue in data mining [2].

However, when any of these frequent item sets increase in length, the algorithm needs to pass through many iterations and as a result, the performance drastically decreases.

In this paper, we propose a modification to the apriori algorithm by using a hash function which divides the frequent item sets into buckets. Further, we propose a novel technique to be used in conjunction with the apriori algorithm by eliminating infrequent item sets from the candidate set. In this top down approach, it finds the frequent item sets without going through several iterations, thus saving time and space. By discovering a large maximal frequent item set very early in the algorithm, all its subsets are also frequent hence we no longer need to scan them. Clearly, the proposed technique has an advantage over the existing Apriori algorithm when the most frequent item set's length is long.

The research paper is written at Pranveer Singh Institute of Technology.

II. Related Work

Our research paper aims at improving recommendation in E-Commerce websites. For that there will be User Interface development which acts as a kind of input module to the project. The existing E-Commerce websites aims at providing recommendation based on the transaction history of the user and sometimes they recommend latest products and also highly rated products. In our project we aim at recommending products to the user based on the transaction history of other users who has same characteristics as this user. So this requires the data mining techniques like clustering

III. Apriori Algorithm

Apriori algorithm is the classic algorithm of association rules, which enumerate all of the frequent item sets [3]. When this algorithm encountered dense data due to the large number of long patterns emerge, this algorithm's performance declined dramatically.The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database: this has applications in domains such as market basket analysis.

Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation) [4].Following are the steps used in its implementation:-

- Text preprocessing
- Mining of frequent itemsets
- Partitioning the text documents based on frequent item sets
- Clustering of text documents within the partition.

Other algorithms are designed for finding association rules in data having no transactions, or having no timestamps (DNA sequencing). Each transaction is seen as a set of items. Given a threshold **C**, the Apriori algorithm identifies the item sets which are subsets of at least **C** transactions in the database.

Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time, and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.

Apriori algorithm is the algorithm used to find association among the items which come together in a transaction [5]. It takes the transaction database as input and gives frequent item set which occur together as output.

Apriori uses breadth-first search and a Hash tree structure to count candidate item sets efficiently. It generates candidate item sets of length K from item sets of length K-1. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.
The pseudo code for the algorithm is given below for a transaction database , and a support threshold of T. Usual set theoretic notation is implied; though note that T is a multiset. C_k is the candidate set for level k. At each step, the algorithm is assumed to generate the candidate sets from the large item sets of the preceding level, heeding the downward closure lemma.Count[c] accesses a field of the data structure that represents candidate set c, which is initially assumed to be zero. Many details are omitted below, usually the most important part of the implementation is the data structure used for storing the candidate sets, and counting their frequencies.

IV. Pseudo Code

```
Apriori(T, $\epsilon$)
$L_1 \leftarrow$ {large 1 – item sets}
K $\leftarrow$ 2
While $L_{k-1} \neq \emptyset$
$C_k \leftarrow \{ a \cup \{b\} \mid a \in L_{k-1} \wedge b \notin a \} -$
      $\{ c \mid \{ s \mid s \subseteq c \wedge |s| = k-1 \} \nsubseteq L_{k-1} \}$
For transactions t $\epsilon$ T
$C_t \leftarrow \{ c \mid c \epsilon C_k \wedge c \subseteq t \}$
For candidates c $\epsilon$ $C_t$
count[c] $\leftarrow$ count[c]+1
$L_k \leftarrow \{ c \mid c \epsilon C_k \wedge \text{count}[c] \geq \epsilon \}$
k $\leftarrow$ k + 1
return $\cup_k L$
```

V. Example

Assume that a large supermarket tracks sales data by stock-keeping unit (SKU) for each item: each item, such as "butter" or "bread", is identified by a numerical SKU. The supermarket has a database of transactions where each transaction is a set of SKUs that were bought together. Let the database of transactions consist of following item sets:

Item sets
{1,2,3,4}
{1,2,4}
{1,2}
{2,3,4}
{2,3}
{3,4}
{2,4}

We will use Apriori to determine the frequent item sets of this database. To do this, we will say that an item set is frequent if it appears in at least 3 transactions of the database: the value 3 is the *support threshold*.

The first step of Apriori is to count up the number of occurrences, called the support, of each member item separately. By scanning the database for the first time, we obtain the following result

Item	Support
{1}	3
{2}	6
{3}	4
{4}	5

All the item sets of size 1 have a support of at least 3, so they are all frequent.

The next step is to generate a list of all pairs of the frequent items.

For example, regarding the pair {1,2}: the first table of Example 2 shows items 1 and 2 appearing together in three of the item sets; therefore, we say item {1,2} has support of three.

Item	Support
{1,2}	3
{1,3}	1
{1,4}	2
{2,3}	3
{2,4}	4
{3,4}	3

The pairs {1,2}, {2,3}, {2,4}, and {3,4} all meet or exceed the minimum support of 3, so they are frequent. The pairs {1,3} and {1,4} are not. Now, because {1,3} and {1,4} are not frequent, any larger set which contains {1,3} or {1,4} cannot be frequent. In this way, we can *prune* sets: we will now look for frequent triples in the database, but we can already exclude all the triples that contain one of these two pairs:

Item	Support
{2,3,4}	2

in the example, there are no frequent triplets. {2,3,4} is below the minimal threshold, and the other triplets were excluded because they were super sets of pairs that were already below the threshold.

We have thus determined the frequent sets of items in the database, and illustrated how some items were not counted because one of their subsets was already known to be below the threshold.

VI . Methods to Improve Apriori's Efficiency

There are several methods to improve the efficiency of Apriori algorithm[6] :

- Hash-based itemset counting: A k-itemset whose corresponding hashing bucket count is below the threshold cannot be frequent.

- Transaction reduction: A transaction that does not contain any frequent k-itemset is useless in subsequent scans.

- Partitioning: Any itemset 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 itemsets only when all of their subsets are estimated to be frequent.

VII. AdvantageS

Apriori algorithm is a classical Data Mining Algorithm which has varied advantages. Firstly, it is the only algorithm that operates on databases with a lot of transactions. Secondly, it uses large item set property thereby avoiding inconsistencies and redundancies. Thirdly, apriori algorithm can be easily paralyzed. Lastly, it is very easy to implement and use.

APRIORI-IMPROVE algorithm presents optimizations on 2-items generation, transactions compression and uses hash structure to generate L2, uses an efficient

VIII. LIMITATIONS

Apriori algorithm, suffers from a number of inefficiencies and trade-offs, which have spawned other algorithms.

Apriori algorithm requires many database scans which makes it very tedious and time consuming. Moreover it assumes that the transaction database is memory resident and is at times very slow. Large numbers of in-frequent itemsets are generated and thus increase the space complexity [7]. More search space is required and I/O cost will be increased. An Improved Apriori Algorithm called APRIORI-IMPROVE is proposed based on the limitations of Apriori algorithm [8].

IX. CONCULSION

In this paper, an improved Apriori is proposed through reducing the time consumed in transactions scanning for candidate item sets. Whenever the **k** of k-item set increases, the gap between our improved Apriori and the original Apriori increases from view of time consumed, and whenever the value of minimum support increases, the gap between our improved Apriori and the original Apriori decreases from view of time consumed. The time consumed to generate candidate support count in our improved Apriori is less than the time consumed in the original Apriori algorithm.

ACKNOWLEDGMENT

We are really grateful to all the people who directly or indirectly helped us in this research work. We are also thankful to Mr. Arunendra Singh (Assistant Professor, Department of Information Technology, PSIT Kanpur) under whose guidance this research paper has been a success.

REFERENCES

[1] Scalable and Efficient Improved Apriori Algorithm, Miss. Nutan Dhange, Prof. Sheetal Dhande Dept. of CE, SCOET,Amravati Univercity, Amravati,India Dept. of CSE, SCOET,Amravati Univercity,Amravati, India.

[2] David Hand,Heikki Mannila,Padhraic Smyth. Principles of Data Mining. translater Yinkun Zhang. Beijing: Mechanical Industry Press. 2003: 272-284.

[3] Research of an Improved Apriori Algorithm in Data Mining Association Rules Jiao Yabing

[4] Association Rule Mining based on Apriori Algorithm in Minimizing Candidate Generation, Sheila A. Abaya 7, July-2012

[5] Recommendation of Books Using Improved Apriori Algorithm ,Nilkamal More (Assistant Professor, Department of Information Technology).

[6] APRIORI Algorithm by Professor Anita Wasilewska

[7] Study of various Improved Apriori Algorithms Deepali Bhende, Usha kosarke , Mnisha Gedam

[8] Rui Chang, Zhiyi Liu, “An Improved Apriori Algorithm”, 2011 International Conference on Electronics and Optoelectronics (ICEOE 2011)

Authors Profile

Karan Kapoor pursuing Btech in Information Technology from Pranveer Singh Institute of Technology, Kanpur.

Shana Parveen pursuing Btech in Information Technology from Pranveer Singh Institute of Technology, Kanpur .

Praveen Gupta is working as an Assistant Professor in Department of Information Technology at PSIT, Kanpur.