Generate Frequent Item Sets with Modified Top down Apriori Algorithm using Mapreduce

Jyoti Yadav  
M.E Student  
Sushila Devi Bansal College of Technology, Indore

Neha Sehta  
Asst. Prof.  
Sushila Devi Bansal College of Technology, Indore

ABSTRACT
As with the advancement of the IT technologies, the data is increasing day by day and it is difficult to manage the data and find out relevant information from it. There are many conventional data mining techniques present for generating frequent item sets. Association rule mining is one of the important task of descriptive technique. Apriori algorithm is the versatile algorithm for generating frequent item sets. Challenge is to improve efficiency by taking less time and produce better results. Hadoop Map reduce introduced by Google overcomes the problem of multiple re-execution of tasks. In this paper Hybrid approach of modified apriori with Hadoop MapReduce is proposed.

General Terms
Modified Algorithm, Frequent Item sets, Apriori Algorithm

Keywords
Hadoop, Map-Reduce, Apriori, Data mining, Support Association rules

1. INTRODUCTION
Data mining is the main part of Knowledge Discovery Database. Data mining normally involves four classes of task; classification, clustering, regression, and association rule learning [1]. Data mining refers to discover knowledge in enormous amounts of data. It is a precise discipline that is concerned with analyzing observational data sets with the objective of finding unsuspected relationships and produces a review of the data in novel ways that the owner can understand and use. Data mining main disciplines are statistics, machine learning and Data management databases. Association rule mining finds interesting associations or correlation relationships among large set of data items [2,3]. Hadoop an open source framework is an effective platform for implementing data mining method with the programming model called MapReduce.

The remainder of this paper is organized as follows. In Section 2, some related work on data mining techniques is discussed. Further in section 3 the Hadoop framework and its main components are explained. In section 4 Association rule mining is discussed with Apriori Algorithm. In section 5 Proposed algorithm is written. In section 6 demonstration of existing and modified algorithm by example is discussed and. Finally conclusion is given.

2. LITERATURE REVIEW
One of the most well known and popular data mining techniques is the Association rules or frequent item sets mining algorithm. AIS algorithm in [4] which generates candidate item sets on-the-fly during each pass of the database scan. Large item sets from preceding pass are checked if they were presented in the current transaction.

Therefore extending existing item sets created new item sets. This algorithm turns out to be ineffective because it generates too many candidate item sets. It requires more space and at the same time this algorithm requires too many passes over the whole database and also it generates rules with one consequent item.

J. Woo [6], presented a MapReduce algorithm based on Apriori algorithm that is a popular algorithm to collect the itemsets that occurred frequently. They have implemented and executed the Apriori-MapReduce algorithm on Hadoop framework. By focusing on the time complexity, their results showed that the proposed algorithm provides high performance computing when the map and reduce nodes are added, as compared to the normal Apriori algorithm. Further, the produced itemsets by the algorithm can be adopted to compute and produce an association rule for market analysis.

Ning. Li. et al [7], implemented a parallel Apriori algorithm based on MapReduce in order to provide a fast and efficient algorithm that can handle large volumes of data, which is becoming a challenging issue nowadays. Their experimental results demonstrated that the algorithm can effectively process large data sets on commodity hardware and it is scalable.

3. HADOOP-MAPREDUCE FRAMEWORK
Hadoop is a framework that provides a distributed file system and helps us to analyze and process large datasets, through MapReduce programming model [8]. The original Hadoop 1.0 consists of two main components called HDFS and MapReduce. Hadoop project is developed in Java includes mainly [12,13]:

Hadoop Common: The collection of normal utilities that support other Hadoop modules.
Hadoop Distributed File System (HDFS): A distributed file system that provides high storage capacity and access to required application data. 
Hadoop YARN: A structure as a central platform to deliver consistent operations, security, and data governance tools across Hadoop Clusters

3.1 HDFS Hadoop Distributed File System
HDFS is a distributed file system used to store files across a collection of servers in a Hadoop cluster. HDFS is implemented in Master/Slave architecture. Basically, there are two significant services running on an HDFS, named as NameNode and DataNodes. In every Hadoop cluster, there is a single NameNode, which runs on the master node [9].

3.2 MapReduce
MapReduce is a parallel programming framework that provides a parallel and distributed platform in order to simplify the difficulties encountered while processing and analyzing large data sets. MapReduce processes the large
The amount of structured and unstructured data by using map and reduce functions. Based on this definition the map and reduce functions are formalized as follows[11,12]:

Map function: map: (key1, value1) => list(key2, value2)
Reduce function: reduce: (key2, list (value2)) =>(key3,value3)

4. ASSOCIATION RULE MINING

The techniques for discovering association rules from the data have conventionally focused on identifying relationships between items telling me feature of human behavior, usually trade behavior for determining items that customers buy together. All rules of this type describe a particular local pattern.

Transaction database DB, I={i1, i2, i3…in} is a set of items with n different itemsets in DB, each transaction T in DB is a set of item (i.e. itemsets), so T I[9]. Support: The support of an itemset is the count of that itemset in total number of transactions.

Definition 1: Let I={i1, i2, …, in} be a set of items. D is a transactional database. Where (k=1,2,3…n) is an item. Tid is the exclusive identifier of transaction T in transactional database.

The association rule x→y has support s in D if the probability of a transaction in D contains both X and Y is s.

Definition 2: If the support of item-sets X is greater than or equal to minimum support threshold, X is called frequent item-sets. If the support of item-sets X is smaller than the be decomposed into two-step process [14]:

Step1: Find all frequent itemsets. By definition, each of these itemsets will occur at least as frequently as a pre-determined minimum support count.

Step2: Generate strong association rules from the frequent itemsets. By definition, these rules must satisfy minimum support and minimum confidence.

4.1 Apriori Algorithm

Apriori employs an iterative approach known as a level-wise search, where k-itemsets are used to explore (k+1)-itemsets.[5]. First, the set of frequent 1-itemsets is found. This set is denoted L1. L1 is used to find L2, the set of frequent 2-itemsets, which is used to find L3, and so on, until no more frequent k-itemsets can be found. The finding of each Lk requires one full scan of the database. In order to find all the frequent itemsets, the algorithm adopted the recursive method. The main idea is as follows [7]:

For Existing Approach C, is A,B,D
For Modified Approach C, is A,B,C,D


<table>
<thead>
<tr>
<th>Table 3. Frequent 2-Itemsets</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Existing Approach</strong></td>
</tr>
<tr>
<td>L₂</td>
</tr>
<tr>
<td>A,B</td>
</tr>
<tr>
<td>A,D</td>
</tr>
<tr>
<td>B,D</td>
</tr>
<tr>
<td>A,B,D</td>
</tr>
</tbody>
</table>

For Existing Approach C₂ is \{A,B\},\{A,D\},\{B,D\}.
For Modified Approach C₂ is\{A,B,D\}.

<table>
<thead>
<tr>
<th>Table 4. Frequent 2-Itemsets</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Existing Approach</strong></td>
</tr>
<tr>
<td>L₂</td>
</tr>
<tr>
<td>A,B,D</td>
</tr>
</tbody>
</table>

So, the frequent itemsets are \{A,B,D\}. Generate association rules, A->BD, AB->D, AD->B, B->AD.

If someone buys items AB together then they likely to buy D at the same time. Similarly, with item A, D B item is also brought. This shows A, B, D are the items which are likely to be purchased together.

```
Algorithm Map_Reduce(count[],D[][],) {
    i=1;
    while(i<No of Transaction) {
        MAPER(no_of_transactions/2)
        MAPER(no_of_transaction/2+1,no_of_transaction)
        REDUCER(i,no_of_transactions/2)
        REDUCER(no_of_transaction/2+1,no_of_transaction)
    }
    return Association Rule
}
```

The Proposed algorithm uses Mapper,Reducer functions to avoid generation of unnecessary patterns.

6. PROPOSED ALGORITHM

```
Algorithm Apriori_MapReduce_Partitioning(D[], sup) {
    // D[][] —Input dataset
    // supp -- Minimum support
    no_transaction = calculate_transaction(D)
    no_item = calculate_item(D);
    for i=1 to no_of_transaction do
        for j=1 to no_of_items do
            if D[i][j]==1 then
                count++,
        }
    }
    for i=1 to no_of_item do
}
```

7. CONCLUSION

In this paper, an algorithm is proposed which is a hybrid approach of modified apriori with hadoop map reduce for generating frequent itemsets. It will overcome the deficiency of classical Apriori algorithm. As proposed algorithm uses top down approach of apriori which reduces the number of database scans and it is useful for large amount of database scans.

8. REFERENCES


