Abstract- An introduction to eclat algorithm is given along with it's map reduce enhancement. Further improvement to map reduce eclat algorithm is given using horizontal layout of database. This algorithm shows good performance in terms of complexity and time.

Keywords- Association rules, data mining, market basket analysis, eclat, mreclat, map reduce, horizontal database.

I. INTRODUCTION

Nowadays there is vast amount of data in database records of even small and medium scale enterprises. In order to process this data and find useful information out of it for profitability of any organization data mining is used. Identifying relationships or association between large sets of data is called association rule mining. The huge amount of data constantly becoming stored and manipulated in data warehouses, several industries are becoming more interested in using association rule mining. For example detecting association relationship in large quantities of transactional business data can thereby assist in catalog design, cross marketing, loss leader analysis and even varied decision making process. A typical and most frequently used example of ARM is market basket analysis wherein the buying patterns of customers is studied to find associations of the most frequent items brought together.

An association rule is of the form A->B, where A is the antecedent and B, the consequent. These are basically if then statements. The idea of support and confidence is used to identify important relationships. Support tells about how frequently the items appear in the database and confidence indicates number of times the if/then statements are found to be absolutely true.

Eclat is a ARM algorithm which makes use of vertical layout of data. This algorithm is highly beneficial as it scans the database only once. Only support is calculated in this algorithm.

Algorithm: Input: Fk = {I1, I2, ..., In} // cluster of frequent k-itemsets. Output: Frequent l-itemsets, l >k.

```
Bottom-Up(Fk){
1. for all li Fk
2. Fk +1 = ∅;
3. for all ljCFk, i < j
4. N = li ∩ lj;
5. if N .sup ≥ min_sup then
6. Fk +1 = Fk +1UN;
7. end
8. end
9. end
10. if Fk +1 ≠ ∅ then
11. Bottom-Up(Fk +1);
12. end
13. }
```

Here, in this algorithm items are stored as input in Fk. Output is a set of frequent itemsets. Firstly, Fk+1 is empty database. Then support of individual items is calculated. Now, the support is measured against the threshold value of support, that is, the minimum support. All the remaining items are taken and put in Fk+1, which was originally empty, so now it contains all the frequent items and is called the first frequent itemset. This process goes on until Fk+1 is empty database.

II. OBJECTIVES

Eclat algorithm is used to find frequent itemsets from large amount of database. But this algorithm has some problems. Like more time complexity is major problem. So our main objectives are:

1. To increase performance in term of time complexity and number of iterations.
2. Use horizontal Layout of data.
3. Compare the performance of old algorithm with new one based on complexity and time.

III. METHODOLOGY

In this algorithm, we will use the top down technique of searching like breadth for search and we use the horizontal dataset. In the traditional eclat algorithm they use the vertical database and bottom up technique for element searching.

We try to reduce time complexity by using top down technique. When we use bottom up approach then complexity is increased. We will use top down technique like breadth first search. When we use vertical dataset, we need to access more transactions for finding frequent itemsets we will use horizontal dataset so we need fewer transactions for generating frequent item sets.

We will take market basket analysis dataset as input. Next step is to preprocess the dataset. We will apply eclat algorithm. That will generate association rules which mean that we find those itemset which are frequently occurring in database and we are using horizontal form of data for storing the data. We will apply traditional data algorithm and enhanced algorithm then will compare the result of both algorithm in the term of performance. Performance can be measured in term of escape time and number of iterations. We will show result of both algorithms. We will prove that enhanced algorithm reduce the number of iterations. There...
may large amount of data in the database. In that database there are numbers of items along with their transaction ids. We will process all the items. Time to process these items will be reduced by enhanced algorithm. There will be less number of iterations required for process these items and escape time is also reduced. After that we will get frequent items. Output is set of frequent itemsets. We will use matlab for implementing enhanced algorithm.

IV. RESULTS

A new algorithm using horizontal database, by using the transpose function on original database is specified as follows:

Procedure: intersect TID sets (T,D)

Input: T is available as horizontal data base

Output: list of frequent item sets

1. Initialize: P= {<1j, t (1j)> for all 1 itemset in T
2. For each item set X with < X, t(X)> in P do PX<Q
3. For each item set < Y, t(Y)> in P such that Y is lexicographically > X do
4. Nxy = X U Y
5. T (Nxy) = t(X)  U t(Y)
6. If support of Nxy >= D
7. Then PX <- PX <- {<Nxy, t (Nxy)>}
8. End
9. End
10. Intersect TID sets (PX,D)
End

When transpose will be taken, it reduces number of transactions in the database to create association rules. This is because will one transaction, you can access many database elements. The second modification which can be made will be based on to replace bottom-up parsing with top-down parsing. This is because, the bottom-up technique had higher complexity as compared to top-up approach. When the higher complexity technique will be replaced by lower complexity technique, it will gave arise to lower processing time.

This algorithm is implemented in MATLAB (R2013a). The comparison of the old and new algorithms is shown in the table below:

<table>
<thead>
<tr>
<th>TABLE I: COMPLEXITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Itemset</td>
</tr>
<tr>
<td>L1</td>
</tr>
<tr>
<td>L2</td>
</tr>
<tr>
<td>L3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE II: TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Itemset</td>
</tr>
<tr>
<td>L1</td>
</tr>
</tbody>
</table>

V. CONCLUSIONS AND FUTURE WORK

The new algorithm comes out to be highly efficient in terms of time and complexity. This can further be extended in order to study scalability effect with this new algorithm. Moreover, map reduce can be used where scalability issues are there.

REFERENCES


