ABSTRACT

Mining Frequent Itemsets is one of the most important concepts of Data Mining. Over two decades many research works have been done on Frequent Itemset Mining. But it becomes a very difficult task when they are applied to Big Data. Many efficient pattern mining algorithms have been discovered in the last two decades, yet most do not hold good for Big Dataset. Recent improvements in the field of parallel programming already provide good tools to tackle this problem. Hadoop is one such tool which is used to process the big data in parallel using MapReduce. Recent studies reveal that MapReduce programming model ensures significant performance gains in context of data mining algorithms.

In terms of memory as well as in execution speed, tree based Pattern growth algorithm is considered as most efficient than the other Frequent Itemset Mining (FIM) methods. Another important thing which is to be considered in Frequent Itemset Mining is that it often generates a very large number of itemsets, which reduces not only the efficiency but also the effectiveness of mining. So Constraint-based FIM has been proved to be effective in reducing the search space in the FIM task and thus improves the efficiency. In addition to this, in almost all Frequent Pattern Mining algorithms generates Frequent 1-itemsets inorder to find the support count(occurrences) of each item in the entire transactions. This task is itself a tedious task in generating Frequent Patterns when considering the hugeness of modern databases available. No explicit strategy has been outlined in these algorithms to perform the aforesaid task.
To overcome the above drawbacks an efficient algorithm called Modified FP Growth has been proposed to mine Frequent Itemsets from big data set. In this algorithm, Map Reduce concept is used to find Frequent Itemsets from Big Data set. In each Data Node Frequent 1-itemsets are generated using a new tree structure called support count tree. This tree can be easily be embedded into any of the existing algorithms aimed at FIM. With the help of this tree Frequent 1-Itemsets are found out quickly and efficiently which in-turn speeds up the generation of Frequent Itemsets of the entire database. In addition to this, to still more increase the efficiency of MapReduce task a cache has been included in the Map phase to maintain support count tree for calculating the Frequent-1 itemsets of each mapper. This reduces the total time of calculating Frequent-1 itemsets since it bypasses the sort and the combine task of each Mapper in the original MapReduce tasks. This in-turn reduces the total execution time of generating Frequent Itemsets of the entire database.

**Keywords:** Data Mining, Frequent Itemset, Constraints, Hadoop, Map Reduce, support count, Frequent 1-itemsets, patterns, cache