ABSTRACT

The Data Mining trends have been developed and researched in terms of technologies and methodologies. Frequent pattern mining is one of the specific data mining tasks, particularly from retail data. The task is to discover all sequential patterns with a user-specified minimum support, where the support of a pattern is the number of data-sequences that contain the pattern. This thesis focuses on problems related to frequent data mining for knowledge based system. These problems have analyzed and the solutions have been made for the problems related to earlier process and new methods have been developed for mining frequent patterns. Initially this thesis work concentrates on past developments in the area of frequent pattern mining and then this research work initially proposed an AprioriAllHybrid algorithm for finding the frequent patterns. Normally the execution time of the algorithm to find sequential pattern depends on total no of candidates generated at each level and the time taken to scan the database. In this proposed method the scanning time is reduced and the number of candidate item sets generated at each step is also reduced since the database is read only for one time, afterwards an intermediate database is created at each iteration. Then the association rule generated by the Apriori algorithm is optimized using genetic algorithm.
To generate strong association rules, Genetic Algorithm operators like selection, crossover and mutation have been applied on association rule generated by Apriori algorithm. The parallel algorithm has been proposed to mine the frequent patterns with a user specified minimum support. The work is shared among n number of processors to compute frequent item sets. So there will be communication between the processors. The time taken to finish the task is very less when compared to other algorithms. The main drawback of this algorithm is cost, because no of processors used will be increased when the number of data items increased. To make it more efficient, partition algorithm have been developed, in this a separate partitioning is created for each sets of data items. To find the count of a particular itemsets, no need to scan the entire database, it is enough to refer only the particular partition. So the scan time has been reduced. Out of all the algorithms, partition algorithm will have better performance over the other existing algorithms.