Frequent Item Set using mining Technique

S.V.Subramanyam

Abstract— Frequent pattern mining has been a focused theme in data mining research for over a decade. Abundant literature has been dedicated to this research and tremendous progress has been made, ranging from efficient and scalable algorithms for frequent itemsets mining in transaction databases to numerous research frontiers, such as sequential pattern mining, structured pattern mining, correlation mining, associative classification, and frequent pattern-based clustering, as well as their broad applications. In this paper, we develop a new technique for more efficient pattern mining. Our method find frequent 1-itemset and then uses the heap tree sorting we are generating frequent patterns, so that many. We present efficient techniques to implement the new approach.

Keywords— Data mining; Frequent Pattern mining; Support; Min Heap; Data structure

I.INTRODUCTION

Data mining is the process of analysing the data from different perspectives and going over the useful information – information that can be used to increase revenue, cuts costs, or both [2]. Precisely data mining is the process of finding correlation or patterns among dozens of fields in large relational database. One of the most important data mining applications is that of mining association rules. Data mining has many virtues and vices which involves in many fields. Some of the examples are (1) Bank – to identify patterns that help be used to decide result for loan application to the customer, (2) Satellite research – to identify potential undetected natural resources or to identify disaster situations, (3) Medical fields – to protect the patients from infectious diseases, (4) Market strategy – to predict the profit and loss in purchase.

Agrawal et al. (1993) was the first man who has proposed frequent pattern mining for market basket analysis in the form of association rule mining [1]. In this he analyses customers shopping basket to finding buying habits by finding associations between the different purchased items by customers. In this paper, we perform a high-level overview of frequent pattern mining methods, extensions and applications. We proposed new method in the place of FP-Growth tree using MINHEAP tree, in this tree we are trying to find most frequent item which is already sorted and present in root of heap tree.

II.FP – GROWTH METHOD FOR MINING FREQUENT PATTERNS

An FP-growth (frequent pattern growth) uses an extended prefix-tree (FP-tree) structure to store the database in a compressed form. FP-growth adopts a divide-and-conquer approach to decompose both the mining tasks and the databases. It uses a pattern fragment growth method to avoid the costly process of candidate generation and testing used by Apriori [2].

FP-Growth adopts a divide-and-conquer strategy as follows. First, it compresses the database representing frequent items into a frequent-pattern tree or FP-tree, which retains the item sets association information. It then divides the compressed database into a set of conditional databases. Each associated with one frequent item and mines each such database separately [3].

Definition 1 (FP-tree) A frequent pattern tree is a tree structure defined below.

a. It consists of one root labelled as “root”, a set of item prefix sub-trees as the children of the root, and a frequent-item header table.

b. Each node in the item prefix sub-tree consists of three fields: item-name, count, and node-link, where item-name registers which item this node represents, count registers the number of transactions represented by the portion of the path reaching this node, and node-link links to the next node in the FP-tree carrying the same item-name, or null if there is none.

c. Each entry in the frequent-item header table consists of two fields, (1) item-name and (2) head of node-link, which points to the first node in the FP-tree carrying the item-name.[3]

TABLE I AN EXAMPLE PREPROCESSED TRANSACTIONAL DATABASE

Tid	List of Items
100	{Pen, Ink, Rubber}
200	{ Pen, Ink, Pencil }
300	{Pencil, Rubber}
400	{Ink, Pencil}
500	{Register, Ink, Pen}

III.DATA STRUCTURES

The data structures are user defined data types specifically created for the manipulation of data in a predefined manner. There are two types of data structure Linear and Non Linear. Array, stacks, queues and Linked List are Linear Data structure whereas trees and graphs are non linear data structure.

A. Tree

Trees are very flexible, versatile and powerful data structures .A tree is a non linear data structure in which items are arranged in a sorted sequence. It is used to represent Hierarchical relationship existing amongst several data structures.

There are different types of trees like Binary tree, B-Tree, B+Tree ,AVL Tree, Extended Tree , Heap Tree etc.

All trees manage data in different types. Here we will explain the functioning of Heap Tree.

B. Heap

The Heap is used in an elegant sorting algorithm called heapsort. Suppose H is a complete binary tree with n elements is maintain in memory using the sequential representation of H. Then H is called the Heap.

Heap is of Two types Max Heap and Min Heap. If root is greater than or equal to their left and right child then it is called Max Heap.If root is less than or equal to their left and right child then it is called Min Heap.

1) Operations on Min Heap:

We can perform different types of operations on Min Heap i.e. Insertion, Deletion, Sorting, Traversing and Searching.

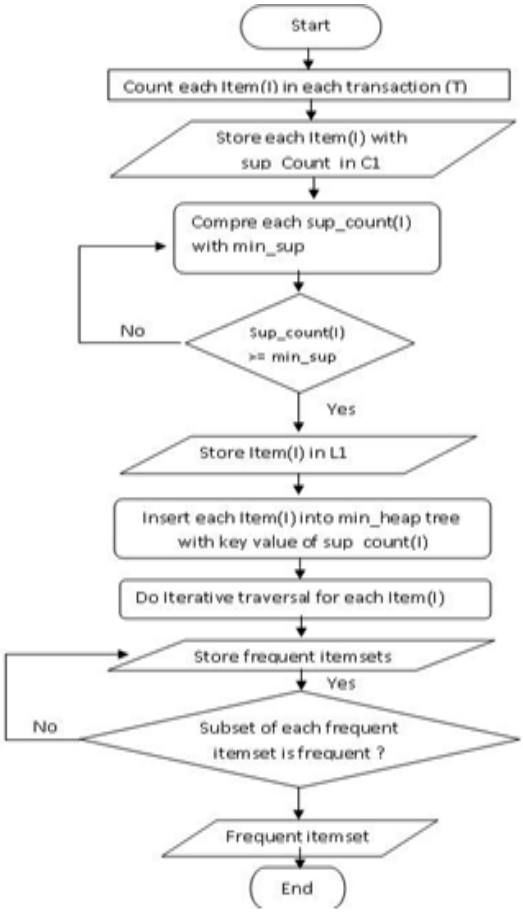
Once a heap has been built, Heap sort can simply remove the minimum value (root node) and create the output, sorted array one item at a time. This process is somewhat akin to the Selection Sort but much more efficient.

When the root node in a heap is removed to become part of the final, ordered data set, the last item on the heap is promoted to fill the vacancy at the root position. Clearly, in many cases, this last item will now be out of place. To ensure that the modified heap retains the min-heap property it becomes necessary to ``push down" the newly promoted root item until it is back in the right place. This pushing down process entails examining the node's key value and comparing it with the key value of the node's smallest child. If the node's smaller child is small in value than the node itself, a swap is performed. The process repeats, following the node from the root.

IV. PROPOSED METHOD

We are introducing a new method for finding the frequent pattern mining which is based on MINHEAP.

A.Flowchart



V.CONCLUSIONS

In this paper, we present the FP-Growth tree. We are trying to exploring the new approach on frequent pattern mining. Hopefully, this short overview may

give rough outline of our recent work and give just idea of general view of MINHEAP tree by using comparison based technique. In general, we feel as a young research field in data mining, frequent pattern mining has achieved tremendous progress and claimed a good set of applications.

However, in-depth research is still needed on several critical issues so that the field may have its long lasting and deep impact in data mining applications.

REFERENCE

[1] Luca Cagliero and Paolo Garza, "Infrequent Weighted Itemset Mining using Frequent Pattern Growth", IEEE Transactions on Knowledge and Data Engineering,2013.

[2] Johannes K. Chiang and Sheng-Yin Huang, "Multidimensional Data Mining for Healthcare Service Portfolio Management", IEEE 2013.

[3] Mohammad Al.Maolegi, Bassam Arkok,An improved apriori algorithm for association rules.

[4] Charu C.Aggarwal,Philip S.Yu.Anew framework for itemset generation in IBM T J Watson Research Centre,1998.

[5] Shweta Sharma and Ritika Pandhi, Proposed algorithm for frequent item sets patterns, vol.3, no.6, june2012

[6] Priyanka Asthana, Anju Singh and Divakar Singh;proposed a survey on association rule mining using apriori based algorithm and hash based methods,vol.3,issue7,july2013.