Efficient Algorithms for Closed and HUI Itemsets Mining

Mahendra M. Kapadnis, Prof. P. B. Koli

1Department of Computer Engineering, Late G. N. Sapkal COE, Nashik, mk250387@gmail.com
2Department of Computer Engineering, Late G. N. Sapkal COE, Nashik, prashant.b.koli@gmail.com

Abstract- High utility itemsets refer to set of items which has high utility like profit or importance in a database, and efficient mining of high utility itemsets plays an important role in many real-life applications. Mining high utility itemsets (HUIs) from large amount of databases is finding of itemsets with high utilities or values (e.g. high profits). Exist too many HUIs to users reduces the efficiency of the mining process. To accomplish high efficiency for the mining job and to provide a summarizing mining result to users from large amount of databases, we propose a new innovative framework in this paper for mining closed high utility itemsets (CHUIs), which functions as a lossless and compact representation of HUIs. Where we propose three capable algorithms named AprioriCH (Apriori-based algorithm for mining High Utility Closed itemsets), AprioriHC-D (AprioriHC algorithm with Discarding unpromising and isolated items) and CHUD (Closed High Utility Itemset Discovery) to find this representation. Later on, a scheme called DAHU (Derive All High Utility Itemsets) is proposed to recuperate all HUIs from the set of CHUIs without accessing the original database. Results which obtain from real and synthetic datasets indicates that the proposed algorithms are very efficient and that the methodologies reach a massive decrease in the number of HUIs. In addition, when all HUIs can be recuperated by DAHU, the combination of CHUD (Closed High Utility Itemset Discovery) and DAHU (Derive All High Utility Itemsets) overtakes the state-of-the-art algorithms for mining HUIs.

Keywords- Frequent itemset, high utility itemset, concise and lossless representation, utility mining, data mining.

I. INTRODUCTION

In data mining Frequent itemset mining is a very important research issue. Market basket analysis is the mostly use one of its widely held applications, in which sets of items (itemsets) discovers that are frequently purchased collected by customers. The traditional model of Frequent itemset mining may finds a huge quantity of frequent itemsets with less profit and miss the data on valuable itemsets having less retailing frequencies. These problems are occurring due to Frequent itemset mining considers all items those having the same importance/unit profit/weight and it imagines that every item in a transaction looks in a binary form, i.e., an item doesn’t indicate its purchase quantity in the transaction it may be in the transaction or maybe not. Hence, Frequent itemset mining cannot fulfil the necessity of users who want to find itemsets with high values such as high profits.

Utility mining plays as an important role in data mining to solve the above issues. In utility mining, each item has its own a weight (e.g. unit profit) and may appear more than once in each operation (e.g. purchase quantity). The value of an itemset shows its importance, as it measured in terms of weight, profit, cost, quantity or other data depending on the user’s first choice. An itemset is named a high utility itemset (abbreviated as HUI) when its utility is no less than a user stated bottom utility threshold. A wide range of applications utility mining has such as website click stream analysis, cross-marketing analysis and biomedical areas.

II. LITERATURE SURVEY
In Dec. 2009, C. F. Ahmed, B. S. Jeong, S. K. Tanbeer, and Y. K. Lee, proposes an efficient tree structures for high utility pattern mining in incremental databases [2] which has three novel tree structures to efficiently accomplish incremental and interactive High Utility Pattern (HUP) mining. The first tree structure, Incremental HUP Lexicographic Tree (IHUPL-Tree), can capture the incremental data without any rearrangement process. The next one tree structure is the IHUP transaction frequency tree (IHUP-TF-Tree), which gains a compressed size by ordering items as per their transaction frequency (descending order). The third tree, IHUP-transaction-weighted utilization tree (IHUPTWU-Tree) is built to decrease the mining period, grounded on the TWU cost of items in descending order.

In 2008, K. Chuang, J. Huang, and M. Chen proposed a Mining top-k frequent patterns in the presence of the memory constraint [5]. This paper explores a practically remarkable mining job to retrieve top-k (closed) itemsets in the occurrence of the memory constraint. To obey with the top bound of the memory intake, two efficient algorithms, called MTK and MTK_Close, are invented for mining frequent itemsets and closed itemsets, in rehearsal, it is pretty challenging to constrain the memory consumption while also efficiently obtaining top-k itemsets. To effectively obtain this, MTK and MTK_Close are invented as level-wise finding algorithms, where the number of candidates are to be generated-and-tested in every database scan will be limited.

In 2003, R. Chan, Q. Yang, and Y. Shen, proposed mining high utility itemsets where mining high utility itemsets from a transactional database [6] refers to the finding of itemsets with high utility like profits. This paper proposes two algorithms, viz. utility pattern growth (UP-Growth) and UP-Growth+, for mining high utility itemsets which has a set of effective policies for pruning candidate itemsets. The data of high utility itemsets is collected in a tree-based data structure named utility pattern tree (UP-Tree) like that candidate itemsets may be generated efficiently by only two scans of database.

In 1994, R. Agrawal and R. Srikant, proposed Fast algorithms for mining association rules in which they consider the problem of determining association rules among items in a huge database of sales transactions. Paper provide two new algorithms for solving the problem that are basically different from the known algorithms. Experimental assessment shows that these algorithms outperform the well-known algorithms by factors ranging from three for minor problems to more than an order of magnitude for huge problems. It also displays how the best features of the two proposed algorithms can be joined into a hybrid algorithm, called as Apriori Hybrid. Scale-up research show that Apriori Hybrid scales linearly with the number of transactions. An Apriori Hybrid also has outstanding scale-up assets in view of transaction size and the number of items in the database.

III. PROPOSED SYSTEM

The Existing system is Frequent itemset mining (FIM) is a basic research topic in data mining. The market basket analysis is one of its popular applications, which related to the discovery of itemsets that are frequently purchased jointly by customers. Where, in this application, the old regular model of FIM may discover a huge amount of frequent but less revenue itemsets and miss the data on valuable itemsets having less retailing frequencies. That type of problems are occur because of the facts that (1) FIM rates all items as having the same importance/unit profit/weight and (2) it consider that every item in a transaction present in a binary form, i.e., an item can be either present or absent in a transaction, which does not shows its purchase quantity in the transaction. Hence, FIM cannot fulfill the requirement of users who desire to find itemsets having high utilities such as high profits. HUI mining is not an easy job as the downward closure property in FIM does not grasp in utility mining. In different way we can say that, the search space for mining HUIs cannot be directly reduced as it is done in FIM because a superset of a less utility itemset can be a high utility itemset. Were proposed for mining HUIs, but they also present a huge number of high utility itemsets to users. A very huge number of high utility itemsets makes it problematic for that users to understand
the results. It may also root the algorithms in way to make it inefficient in terms of time and memory necessity, or may run it out of memory. It is broadly acknowledged that the more high utility itemsets the algorithms create, the more processing they consume. The performance of the mining job reduces significantly for less minimum utility thresholds or when dealing with condensed databases.

![Figure 1. Block diagram of the proposed system](image)

**IV. MATHEMATICAL MODEL**

**4.1 Problem description**

1) Input
2) Push Closed Property
3) Apriori HC Algorithm
4) Apriori HC D Algorithm
5) Recovery of HUI

Let the system be described by S,

S= \{D, I, PCP, HC, HCD, R\} Where,

S: is a System.
D: is the set of Dataset.
I: Input.
PCP: Push Closed Property
HC: Apriori HC Algorithm
HCD: Apriori HC D Algorithm
R: Recovery of HUI

**4.2 Vein Diagram**

![Vein Diagram](image)

Where,
D: is the set of Dataset.
I: Input.
PCP: Push Closed Property.
HC: Apriori HC Algorithm.
HCD: Apriori HC D Algorithm.
R: Recovery of HUI.

4.3 Functional Dependencies

<table>
<thead>
<tr>
<th>Fn1</th>
<th>Fn2</th>
<th>Fn3</th>
<th>Fn4</th>
<th>Fn5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fn1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fn2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fn3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Fn4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Fn5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Where,
Fn1: input
Fn2: push closed property
Fn3: apriorihc algorithm
Fn4: Apriori HC D Algorithm
Fn5: Recovery

In the proposed method giving input database to AprioriHC algorithm to calculate frequent itemset and also it calculates the high closed itemset that is frequent and high closed itemsets mining is done using following algorithm.

Initially, a variable k is set to 1. The algorithm performs a database scan to compute the transaction utility of each transaction. At the same time, the TWU of each item is computed. Each item having a TWU no less than abs min utility is added to the set of 1 HTWUIs Ck. Then the algorithm proceeds recursively to generate itemsets having a length greater than k. During the kth iteration, the set of k HTWUIs Lk is used to generate (k + 1) candidates C(k+1) by using the Apriori-gen function [1]. Then the algorithm computes TWUs of itemsets in C(k+1) by scanning the database D once. Each itemset having a TWU no less than abs min utility is added to the set of (k +1) HTWUIs L(k+1). After that, the algorithm removes non-closed itemsets in L(k+1) by the following process. For each candidate X in L(k+1), the algorithm checks if there exists a subset Y subset of X such that Y is subset of L and SC(X) = SC(Y). If true, Y is deleted from L because Y is not a closed high utility itemset. If false, Y is kept and marked as closed because it may be a closed high utility itemset. The phase I of AprioriHC terminates when no candidate is generated. Then, the algorithm performs Phase II. In phase II, the algorithm scans the database once and calculates the utilities of HTWUIs that are marked as closed to identify the set of closed high utility itemsets.

V. RESULTS

As the first step is to calculate frequent itemset mining with high closed itemset given a dataset as input. After giving a dataset as input it calculate a frequent high closed itemset using apriori algorithm. Following figure shows the input dataset module and after that the result calculated by the aprioriHC algorithm.
CONCLUSION

The difficulty of redundancy in high utility itemset mining by proposing a lossless and compact representation termed closed high utility itemsets is solved. To do the mining of this representation, three capable algorithms called AprioriHC (Apriori-based approach for mining High Utility Closed itemset), AprioriHC-D (AprioriHC algorithm with Discarding unpromising and isolated items) and CHUID (Closed High Utility itemset Discovery). AprioriHC-D is an improved version of AprioriHC. Next step is to perform CHUD depth-first search for mining closed high utility itemsets from vertical database. To efficiently recover all high utility itemsets from closed high utility itemsets, we proposed an efficient method termed DAHU (Derive All High Utility itemsets). The combination of CHUD and DAHU is also faster than UP-Growth when DAHU could be applied.

REFERENCES
