www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issue 4 April 2016, Page No. 16276-16280

Efficient Mining Algorithm For Reducing Number Of Itemsets Using Chud

***Hari.D*[1], *Suganya.R*[2], *Parameshwari.M*[3]**

[1] Assistant Professor, Department of Computer Science and Engineering, Dr.Mahalingam College of Engineering and Technology,
Pollachi, Tamil Nadu, India
hari@gmail.com

[2] UG Scholar, Department of Computer Science and Engineering, Dr.Mahalingam College of Engineering and Technology,
Pollachi, Tamil Nadu, India
suganyamcet2907@gmail.com

[3] UG Scholar, Department of Computer Science and Engineering, Dr.Mahalingam College of Engineering and Technology,
Pollachi, Tamil Nadu, India
paramesh4768@gmail.com

***Abstract:** Data Mining plays an essential role for mining useful pattern hidden in large databases. Apriori algorithm is used to find frequent itemsets in large databases. Apriori is a Bottom-up generation of Frequent item set combinations. Frequent itemset mining may discover the large amount of frequent but low revenue itemsets and lose the information on the valuable itemsets having low selling frequencies. High Utility Itemset mining identifies itemsets whose utility satisfies the given threshold. It allows the users to quantify the usefulness or preferences of items using different values. A High Utility Itemset which is not included in another itemset having the same support is called Closed High Utility Itemset. Mining High utility itemsets uses Apriori algorithm which takes the Input as Frequent Itemsets from the Transactional database, profit, and price and gives the High Utility Itemsets as the Output. To mine the Closed High Utility Itemsets the system addresses an efficient Depth-First search algorithm named CHUD.*

Keywords: ***Frequent item set, High Utility Itemset, Closed High Utility Itemset, CHUD.***

1. Introduction

Data mining and knowledge discovery from the data bases has received much attention in recent years. Knowledge Discovery in Databases is the nontrivial process of identifying valid, previously unknown and potentially useful pattern in data. Data mining, the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge-driven decisions. The automated, prospective analyses offered by data mining move beyond the analyses of past events provided by retrospective tools typical of decision support systems. Data mining tools can answer business questions that traditionally were too time consuming to resolve. Data mining tools can answer business questions that traditionally were too time consuming to resolve. Frequent pattern mining has been the focus of great interest among data mining researchers and practitioners. It is today widely accepted to be one of the key problems in the data mining fields. Frequent pattern mining is to find the item sets that appear frequently from plenty of data, an itemset is any subset of the set of all items. Once a set of frequent itemsets has been found, association rules can be generated. Association rules are of the form A→B, and could be read as “A implies B”. Each association rule has support (how common the precondition is in the dataset), confidence (how often the precondition leads to the consequence in the dataset). To mine the frequent itemsets the system addresses the use of Apriori algorithm.Apriori algorithm [1] is used for large transactional databases and it is very influential algorithm over other algorithms as other algorithms are derived from this algorithm. Apriori is a Bottom-up generation of Frequent item set combinations. The problem of frequent itemset mining is popular. But it has some important limitations when it comes to analyzing customer transactions. An important limitation is that purchase quantities are not taken into account. Thus, an item may only appear once or zero time in a transaction. Thus, if a customer has bought five breads, ten breads or twenty breads, it is viewed as the same. A second important limitation is that all items are viewed as having the same importance, utility of weight.

High utility itemset mining is useful in decision making process of many applications, such as retail marketing, web services. The problem of high-utility itemset mining is to find the itemsets (group of items) that generate a high profit in a database when they are sold together. The basic meaning of the utility is the interestedness or importance or profitability of the items to the users.

An itemset is called closed high utility itemset if its utility is no less than a user specified minimum utility threshold and it has no proper supersets having the same support. Although the set of closed HUIs is a compact representation of HUIs, it is not lossless. To tackle this problem, each closed HUI is annotated with a special structure called utility unit array such that the resulting itemset is called a Closed High Utility Itemset. The idea of utility unit array makes the set of CHUIs lossless because HUIs and their utilities can be derived from this set without accessing the original database. Besides, it was shown that the set of CHUIs can be several orders of magnitude smaller than the set of all HUIs, especially for dense databases. Moreover, they proposed the first algorithm named CHUD to discover CHUIs in databases.

The rest of this paper is organized as follows. Important related work is introduced in section 2, Existing system is introduced in section 3, Proposed system is introduced in section 4, Experiment result is explained in section 5, Conclusion and future work is explained in section 6.

DOI: 10.18535/Ijecs/v5i4.44

2. Related Work

Based on our literature there is no published work describing the use of online communities like forum for mining frequent itemsets. Existing work on frequent itemsets is mainly based on transactional dataset. Their approaches are helpful to construct commonsense knowledge for frequent itemsets. Notably, there is some work on knowledge extraction from web online communities to support and summarization. These researchers approaches utilize the characteristics of the best fit for their specific tasks. Frequent itemset mining is fundamental in data mining task.

3. Methods

3.1 Existing System

A naïve approach would be to first mine the complete set of frequent itemsets and then remove every frequent itemset that is a proper subset of, and carries the same support as, an existing frequent itemset. However, this is quite costly. One of its popular application is market basket analysis, which refers to the discovery of set of items that are frequently purchased by the customers. Frequent itemset mining [1] may discover the large amount of frequent but low revenue itemsets and lose the information on the valuable itemsets having low selling frequencies. These problems are caused by the following.

- FIM treats all the itemsets as having the same importance/unit profit/weight.
- It assumes that in every time in the transaction appear in the binary form i.e., an item can be either present or absent in the transaction, which does not indicates its purchase quantity in the transactions.
- Hence it does not satisfy the requirements of the users who desire to discover itemsets with high utilities such as high profits

The search space for mining HUIs cannot be directly reduced as it is done in FIM because a superset of a low utility itemset can be a high utility itemset. A very large number of high utility itemsets makes it difficult for the users to comprehend the results. It may also cause the algorithms to become inefficient in terms of time and memory requirement, or even run out of memory. It encompasses predictive and descriptive methods for data mining. These representation successfully reduce the number of itemsets found, but they are developed for frequent itemset mining instead of HUI mining. The main objective of high utility itemset mining is to find all itemsets having greater or equal to user defined minimum utility threshold.

Apriori algorithm generates lots of candidate sets and scans the database every time. When a new transaction is added to the database, then it should rescan the entire database again. Generation of Candidate itemsets is expensive. Existing methods often generate a huge set of high utility itemsets and their mining performance is degraded consequently. The huge number of HUIs form a challenging problem to the mining performance since the more HUIs the algorithm generates, the higher processing time it consumes. Needs several iterations of Data and difficult to find rarely occurring events. FP -growth is not usable to find the high utility itemsets.

3.2 Proposed System

Mining high utility itemsets from databases is an important data mining task, which refers to the discovery of itemsets with high utilities (e.g. high profits). However, it may present too many HUIs to users, which also degrades the efficiency of the mining process. If the number of frequent sets for a given database is large it could become infeasible to generate them all. Moreover, if the database is dense, or the minimal support threshold is set too low, then there could exist a lot of very large frequent sets, which would make sending them all to the output infeasible to begin with. Since the collection of all frequent sets is downward closed, it can be represented by its maximal elements, the so called maximal frequent sets. Another very popular concise representation of all frequent sets are the so called closed frequent sets. A set is called closed if its support is different from the supports of its supersets. Most algorithms that have been proposed to find the maximal frequent itemsets rely on the same general structure as the Apriori algorithm. To achieve high efficiency for the mining task and provide a concise mining to the users, the system goes for mining closed high utility itemsets, which serves as a compact and lossless representation of HUIs. In the proposed system, a condensed and meaningful representation of HUIs named Closed High Utility Itemsets is mined using the AprioriHC-D (Apriori-based algorithm for mining High utility Closed itemset) and CHUD (Closed High Utility itemset Discovery), which integrates the concept of closed itemsets into high utility itemset mining. The system incorporate the concept of closed itemset with high utility itemset mining to develop the closed high utility itemsets. CHUD method is adapted for mining CHUIs and include several effective strategies for reducing the number of candidates generated. The CHUD generates candidates in a recursive manner. There are several possibilities to incorporate the closed constraint into high utility itemset mining. One possibility is to define the closure on the supports of itemsets. In this case, there are two possible definitions depending on the join order between the closed constraint and the utility constraints they can be explained. To present representative HUIs to users, some concise representations of HUIs were proposed. A HUI is said to be maximal if it is not a subset of any other HUI. Although this representation reduces the number of extracted HUIs, it is not lossless. The reason is that the utilities of the subsets of a maximal HUI cannot be known without scanning the database. Besides, recovering all HUIs from maximal HUIs can be very inefficient because many subsets of a maximal HUI can be low utility.

To incorporate the closed constraint into high utility itemset mining. There are several possibilities. First, we can define the closure on the utility of itemsets. In this case, a high utility itemset is said to be closed if it has no proper superset having the same utility. However, this definition is unlikely to achieve a high reduction of the number of extracted itemsets since not many itemsets have exactly the same utility as their supersets in real datasets. A second possibility is to define the closure on the supports of itemsets. In this case, there are two possible definitions depending on the join order between the closed constraint and the utility constraint:

- Mining all the high utility itemsets first and then apply the closed constraint.
- Mining all the closed itemsets first and then apply the utility constraints.

The proposed representations is lossless due to a new structure named utility unit array that allows recovering all HUIs and their utilities efficiently. The proposed representation is also compact. It reduces the number of itemsets by several orders of magnitude, especially for datasets containing long high utility itemsets.

3.3 Process Flow Diagram

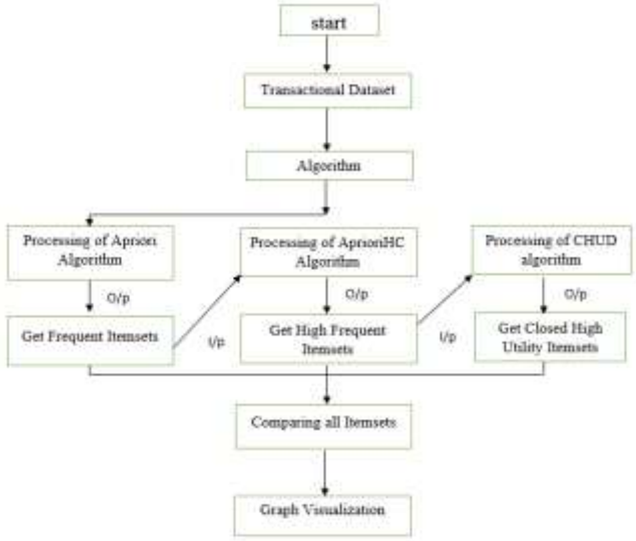
start
Transactional Dataset
Algorithm
Processing of Apriori Algorithm
Processing of AprioriHC Algorithm
Processing of CHUD algorithm
O/p
O/p
O/p
Get Frequent Itemsets
I/p
Get High Frequent Itemsets
I/p
Get Closed High Utility Itemsets
Comparing all Itemsets
Graph Visualization

Figure 1: Block diagram of the system

3.4 Module Description

A Module is a part of a program. The modules are given: Frequent itemset mining acting as an necessary role in the theory and practice of many important data mining task, like mining association rules, long patterns, emerging patterns and dependency rules. It has been useful in the meadow of telecommunications, census analysis and text analysis. Frequent itemset mining leads to the discovery of associations and correlations among items in large transactional correlational datasets. With massive amounts of data continuously being collected and stored, many industries are becoming interested in mining such patterns from their databases. The discovery of interesting correlation relationships among huge amounts of business transaction records can help in many business decision-making processes. The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent itemset properties. Apriori employs an iterative approach known as a level-wise search, where k-itemsets are used to explore (k+1)-itemsets. In Utility Mining, each item has a weight (eg. unit profit) and can appear more than once in each transactions (e.g. purchase quantity). The utility of an itemset represents its importance, which can be measured in terms of weight, profit, cost, quantity or other information depending on the user preferences. Mining High utility itemsets uses AprioriHC algorithm which takes the Input as Frequent Itemsets from the Transactional database, profit, and price and gives the High utility Itemsets as the Output. Apriori property that all nonempty subsets of a frequent itemset must also be frequent. At the k-th iteration (for k≥2),it forms frequent k-itemset candidates based on the frequent(k−1)-itemsets, and scans the database once to find the complete set of frequent k-itemsets, variations involving hashing and transaction reduction can be used to make the procedure more efficient. Other variations include partitioning the data (mining on each partition and then combining the results) and sampling the data (mining on a data subset). These variations can reduce the number of data scans required to as little as two or even one. It has certain limitations. The algorithm explains only the presence and absence of an item in transactional databases.

CHUD algorithm considers vertical database and mines CHUIs in a depth-first search. It performs a database scan to compute the transaction utility of each transaction. At the same time, Total weighted Utility of each item is computed. Each item having a total weighted utility no less than abs_min_utility is added to the set of high transaction weighted utility itemset. CHUD algorithm considers vertical database and mines CHUIs in a depth-first search. It performs a database scan to compute the transaction utility of each transaction. Proposed technique adopts a Compact representation to maintain the utility information of itemset in databases with several effective strategies integrated to prune the search space. To reduce the number of joins that are performed, we propose a novel pruning strategy named EUCP (Estimated Utility Co-occurrence Pruning) that can prune itemsets without having to perform joins. This strategy is easy to implement and very effective. Thus techniques to prune the search space developed in FIM cannot be directly applied in HUI. We have presented a novel algorithm for high-utility itemset mining named CHUD. This algorithm integrates a novel strategy named EUCP (Estimated Utility Co-occurrence Pruning) to reduce the number of joins operations when mining high-utility itemset using the utility list data structure.

3.5 Graph Visualization

To help the user understand what they are seeing, developers often turn to bar and pie charts. A graph is a picture designed to express words, particularly the connection between two or more quantities. Graphs make information easier to see. It is a pictorial way of representing relationships between various quantities, parameters, or measurable variables in nature. It basically summarizes how one quantity changes if another quantity that is related to it also changes. This is especially true when two or more sets of numbers are related in some way. But that only works for discrete data when at the links between data other tools come into play. Graph Visualization is a way of representing structural information as diagrams of abstract graphs and networks. It has important applications in networking, bioinformatics, software engineering, database and web design, machine learning, and in visual interfaces for other technical domains. Graphical Visualization is to analyze different patterns of mining the itemsets of the Transactional database. For a single Transactional database it shows what are all the frequent itemsets, high utility itemsets, closed high utility itemsets. This system uses different algorithm for mining the various itemsets like Apriori algorithm for mining frequent itemsets, AprioriHC for mining high utility itemsets, and CHUD algorithm for mining closed high utility itemsets. With these different algorithms the various results has been produced. The obtained results from the algorithm has different number of itemsets and variation in time consumption. In the obtained graph the X-axis represents time, count, cycles, whereas the Y-axis represents the score obtained by different algorithm. From the result, by comparing all the algorithms, it is found that CHUD is the efficient method for reducing the number of itemsets.

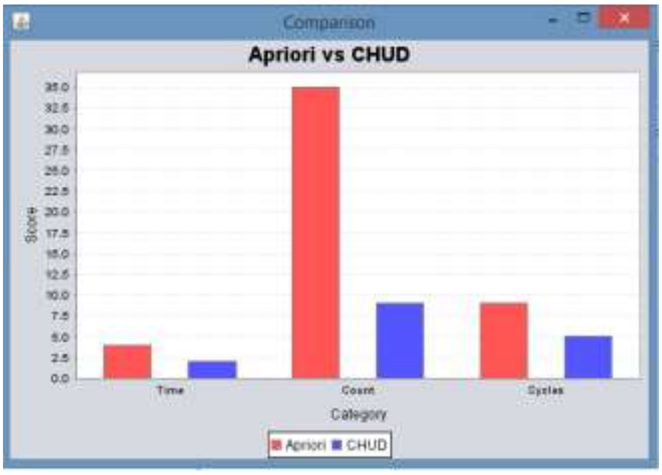

Figure 2: Snapshot of Graphical Visualization

3.6 Implementation

The closure property of Apriori to restrict the number of itemset to be examined, a heuristics is used to predict whether an itemset should be added to the candidate set However, the prediction usually overestimates, especially at the beginning stages, where the number of candidates approaches the number of all the combinations of items. The examination of all the combinations is impractical, either in computation cost or in memory space cost, whenever the number of items is large or the utility threshold is low. Classical algorithm was first used to obtain the frequent item sets without weights. After weight assigning approach, attributes with weighted support less than minimum weighted support were removed. Weighted approach with the basic Apriori was introduced to address the problem of using single minimum support for selecting the frequent item sets. There are two theorems to improve the Apriori algorithm to reduce the times of scanning frequency itemset.

Theorem1:- suppose X and Y are two subsets of transaction T and X is subset of Y. if Y is frequent itemset then X must be frequent itemset.

Theorem 2:- suppose X and Y are two subsets of transaction T and X is subset of Y. if Y is not frequent itemset then X must not be frequent itemset.

The working of APRIORI algorithm is based on the rule of sufficient repeat occurrence of a particular item in a transactional data set, enough to maintain the minimum support count requirement. The fundamental pattern of Apriori working is described as follows.

STEP 1: For the given transactional dataset the algorithm checks for the number of items in singularity and the frequency of occurrence for the same.

STEP 2: The ITEMS are identified and the first candidate set is made that lists down its frequency of occurrence.

STEP 3: This candidate set is then checked with the minimum support count given to identify the ITEMS that are above the support count.

STEP 4: After eliminating the ITEMS that fallout from the requirement the remaining ITEMS are then clubbed to create another candidate set that will check their occurrence in a group of TWO, THREE, and FOUR and so on as far as possible.

STEP 5: Each time a candidate set is compared to the support count, it is checked that there are ITEMS or Group of ITEMS that surpass the minimum threshold value in order to make the new candidate set, once this cannot be achieved the generation of candidate steps is terminated and the Association rules are derived from the last generated candidate set and the first ITEM set.

STEP 6: Identification of strong and weak association rules, an association rule that is generated from the ITEM Sets derived from the above process is then compared to the threshold value of the particular transactional dataset. For example, if there is a Rule that says that 8 of 10 people who bought Milk also purchased Bread states that if Milk and Bread rule has to be made and since Bread occurred in 8 Transactions of Milk it is 80% Confidence and given threshold is 70% thus, it is affirm that this rule is a STRONG rule. On the other hand if the number of people purchasing Bread would have been 6 the confidence level of the transaction would have been 60% which is below the given threshold level and thus the rule stands to be a WEAK Rule.

4. Conclusion and Results

To mine the frequent itemsets the system addresses the use of Apriori algorithm. It is used to mine all frequent item sets in database. It uses knowledge from the previous iteration phase to produce the frequent itemsets. In the first, the algorithm scans the database to find frequency of 1-itemsets that contains only one item by counting each item in the database. High utility itemset mining identifies itemsets whose utility satisfies the given threshold. It allows users to quantify the usefulness or preferences of items using different values. The utility of an itemset represents its importance, which can be measured in terms of weight, profit, cost, or other information depending on the user preferences. The system uses AprioriHC algorithm to find the High Utility Itemsets, whereas, it takes the Input as Frequent Itemsets from the Transactional database, profit, price, and quantity and gives the High Utility Itemsets as the Output. It have certain limitations. This limitations are overcome by mining closed high utility itemsets in phase 2 and implemented graphical visualization of the mined itemsets. CHUD is 50 times faster than Apriori algorithm and it generates a much smaller number of candidates and results than Apriori algorithm. For the dense dataset the proposed representation can achieve a massive reduction in the number of extracted patterns and for the sparse datasets it achieves less reduction.

5. References

[1] M.-S. Chen, J.-S. Park, and P.S. Yu, "Efficient Data Mining for Path Traversal patterns", IEEE Trans. Knowledge and Data Eng.,vol. 10,no. 2,pp. 209-221,Mar. 1998.

[2] P. Kanikar, "Comparitive study of apriori algorithm performance on different datasets", in Proceedings of 20th international Conference on Very Large Databases , 2010, pp.38-43.

[3] Karandeep Singh Talwar, Abishek K Oraganti "Recommendation system using apriori algorithm," in IJSRD,Vol-3 2009.

[4] Nilesh S.Korde, prof.Shailendra W Shende, "Parallel Implementation of apriori algorithm", in: IEEE Transactions on Knowledge and Data Engineering 21(12) (2009),pp01-04

[5] Sudip Bhattacharya, Deepty Dubey, "High utility itemset Mining":Proceeedings of the 3rd International Conference on Knowledge Discovery and Data Mining, Newport Beach, California, 2002, pp. 115-118

[6] http://www.kdnuggets.com/datasets/

[7] Data Mining Concepts and Techniques 2nd Ed By Kamber pages 234-242.

[8] http://archive.ics.uci.edu/ml/

[9] Jiawei Han and Micheline Kamber, "Data Mining concepts and Techniques" 2nd edition Morgan Kaufmann Publishers, San Francisco2006.

[10] R.Nandhini, Dr.N.Suguna, "Shrewd Technique for mining High Utility Itemset", in IJCSIT, vol. 6(6).

[11] Jiawei Han, Jian Pei, and Yiwen Yin, "Mining Frequent patterns without Candidate Generation", Proceedings of the 2000 ACM SIGMOD international conference vol. 29, 2000.

[12] Cheng-Wei Wu, Philippe Fournier-Viger, Jia-Yuan Gu, Vincent S. Tseng, "Mining Closed High Utility Itemsets without Candidate Generation".

[13] Vincent S. Tseng, Chun-Jung Chu, Tyne liang "Efficient mining of Temporal High utility itemsets from Data streams", Proceedings of Second International Workshop on Utility-Based Data Mining, August 20, 2006

[14] L. Szathmary, A. Napoli, P. Valtchev, “Towards Rare Itemset Mining” Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence, 2007, Volume 1, Pages: 305-312, ISBN ~ ISSN:1082-3409 , 0-76953015-X.

[15] http://data-mining.philippe-fournier-viger.com/introduction-high-utility-itemset-mining/