International Journal of Computer Science and Engineering Communications,
Volume.4, Issue.2 (2016): Page.1366-1371
www.scientistlink.com/ijcsec

Mining of Compact and Lossless High Utility Itemset Using Systolic Tree

R.T.Yamine[1], Dr. N.S.Nithya[2]
PG Scholar[1], Associate Professor[2]
Department of Computer Science and Engineering
K.S.R.College of Engineering,India
yaminerajan@gmail.com

Abstract--*Mining high utility itemsets from a transactional database refers to the discovery of itemsets with high utility like profits. Although a number of relevant algorithms have been proposed in recent years, they incur the problem of producing a large number of candidate itemsets for high utility itemsets. Such a large number of candidate itemsets degrades the mining performance in terms of execution time and space requirement. The Systolic tree structure improving the processing speed in proposed system. The systolic tree mechanism is used in the transaction database for extracting the frequent pattern itemsets. Systolic tree based rule mining scheme is combined with weighted association rule mining(WARM) process which is used to fetch the frequently accessed itemsets with its weight value. Based on the item request count and span time values, it estimates the weight value. The performance of the proposed systolic tree algorithm for high utility itemset mined results is compared with the earlier methods such as UP-Growth and FP-Growth methods in terms of the parameters like time, memory space, and runtime for each and every number of transaction and educational dataset.*

Keywords *- Association Rule Mining, Data Mining, Systolic tree mechanism, Utility-based mining*

1. INTRODUCTION

Data mining is the process of realizing useful knowledge from a group of data. Association Rule Mining (ARM) is an important data mining technique which is used to discover the rules/patterns among items in a huge database. Association rule mining with itemset frequencies are used to mine itemset interactions. Frequent pattern mining algorithms are designed to find frequently occurring sets in databases. Memory and run time requests are very high in frequent pattern mining algorithms. A transaction database consists of two features such as internal and external utility. Capacity of a product in a particular transaction is called the internal utility and the profit rate of a product is called external utility. The utility of itemset is well-defined as the product of external utility and internal utility.

Utility of Itemset (U) = external utility * internal utility

In many areas of business like inventory, retail, etc. decision making is very significant. In a transaction database each item is represented by a binary value, without allowing for its profit. In several applications like cross-marketing in retail stores, website click- stream analysis, online e-commerce management and finding the dynamic pattern in bio- medical applications High utility mining are widely used.

2. Related work

A parallel Apriori algorithm version is proposed in [7]. In [8], the FP-growth algorithm is implemented on PC clusters. Though parallel computing platforms can improve the processing speed of software algorithms, measured speedup has not scaled with the number of processors. In [8], scaling efficiency with number of processors is only 13.4/16 = 0.84 and 22.6/32 = 0.71, respectively. In [7], only 2 times speedup was achieved when using 8 nodes. An advantage FPGAs have over parallel computing platforms is the ability to parallelize algorithms at the instruction-level granularity, as opposed to a higher level module-level granularity. The poor scaling of software-based algorithms has sparked research into hardware-based methods for data mining. A parallel implementation of the Apriori algorithm on FPGAs was first done in [9]. Due to the processing time involved in reading the transactional database multiple times, the hardware implementation was only 4 times faster than the fastest software implementation. They further explored the parallelism and developed a bitmapped CAM architecture which provides 24 times performance gain over the software version [10]. Achieving a better speedup using Apriori will be difficult due to the nature of the algorithm which must read the entire database once for every item in the worst case. FPGrowth algorithms are an alternative to Apriori-based solutions. Conclusions from several independent evaluations indicate that FP-growth is the current optimal association rules mining algorithm [11, 2].

Table1. Transactional Database

ID	ITEMS
1	BCD
2	BC
3	ACD
4	ACD
5	ABC

While software solutions exist, we do not know of any existing hardware implementations of the FP-growth algorithms.

3. Data flow diagram

The following Diagram shows the sequence process of calculating and demonstrating a high utility Itemsets. From this Fig. , the comparison of frequent Itemset with given threshold value and by considering a profit, gives High utility Itemsets.

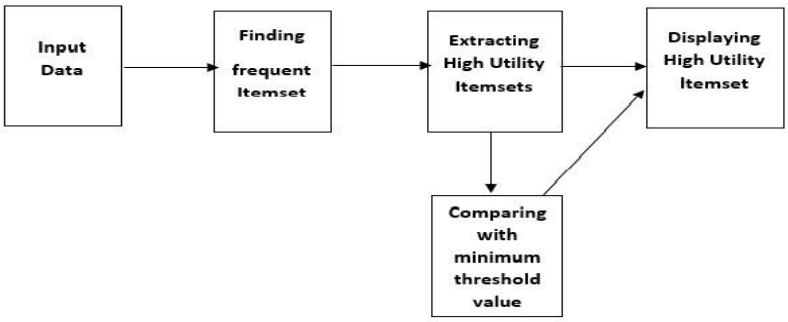


FIG.1 Data Flow Diagram

4. Association rule learning

In data mining, association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases. Piatetsky-Shapiro describes analysing and presenting strong rules discovered in databases using different measures of interestingness. Based on the concept of strong rules, Agrawal et al. Introduced association rules for discovering regularities between products in large scale transaction data recorded by point-of-sale (POS) systems in supermarkets. For example, the rule {Onions, Potatoes}=>{burger} found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, he or she is likely to also buy hamburger meat. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements. In addition to the above example from market basket analysis association rules are employed today in many application areas including Web usage mining, intrusion detection and bioinformatics.

There are two process in Association Rule Mining:
1. Find frequent pattern using Support

2. Discover rules from frequent pattern using Confidence

An association rule has two parts, an antecedent part (if) and a consequent part (then). An antecedent is a set of item found in the transaction. A consequent is a resultant item that is found in combination with the antecedent. Association rules are created by analysing data for frequent patterns and using the criteria support and confidence to identify the most important relationships. Support is an indication of how frequently the items appear in the database. Confidence indicates the number of times the if/then statements have been found to be true.

4.1 Support

The support value of X with respect to T is defined as the proportion of transactions in the database which contains the itemset X. In formula: supp(X) =(number of transaction containing the set)/(total number of transaction). In the example

database, the itemset {milk, bread, butter} has a support of 1/5=0.2 since it occurs in 20% of all transactions (1 out of 5 transactions). The argument of supp(X) is a set of preconditions, and thus becomes more restrictive as it grows.

4.2 Confidence

The confidence value of a rule, X=>Y, with respect to a set of transactions T, is the proportion the transactions that contains X which also contains Y. Confidence is defined as: Conf(X=>Y) = supp (XUY)/supp(X). For example, the rule {butter, bread}=>{milk} has a confidence of 0.2/0.2=1.0 in the database, which means that for 100% of the transactions containing butter and bread the rule is correct (100% of the times a customer buys butter and bread, milk is bought as well). Discovering useful patterns hidden in a database plays an essential role in several data mining tasks, such as frequent pattern mining, weighted frequent pattern mining, and high utility pattern mining.

5. Frequent pattern analysis

The repetition analysis is seen as finding frequently occurring patterns in the text. Frequent pattern is a pattern (in our case a word, or phrase) that occurs frequently in the data set (in our case the document set). Essentially frequent pattern analysis compares every item to every other item. Algorithms have been optimized to cut off this comparison when certain conditions are met. These frequent patterns can also be seen as an automated hypothesis generation. But the patterns need to be evaluated by the domain expert. Frequent pattern analysis generates lots and lots of patterns. Some are not interesting because they report known and/or common occurrences, but other times they may find a novel pattern.

Discovering useful patterns hidden in a database plays an essential role in several data mining tasks, such as frequent pattern mining, weighted frequent pattern mining, and high utility pattern mining. Among them, frequent pattern mining is a fundamental research topic that has been applied to different kinds of databases, such as transactional databases streaming databases and time series databases and various application domains, such as bioinformatics. Nevertheless, relative importance of each item is not considered in frequent pattern mining. To address this problem, weighted association rule mining was proposed. In this framework, weights of items, such as unit profits of items in transaction databases, are considered. With this concept, even if some items appear infrequently, they might still be found if they have high weights. However, in this framework, the quantities of items are not considered yet. Therefore, it cannot satisfy the requirements of users who are interested in discovering the itemsets with high sales profits, since the profits are composed of unit profits, i.e., weights, and purchased quantities.

Mining high utility itemsets from databases refers to finding the itemsets with high profits. Here, the meaning of itemset utility is interestingness, importance, or profitability of an item to users. Utility of items in a transaction database consists of two aspects: 1) the importance of distinct items, which is called external utility, and 2) the importance of items in transactions, which is called internal utility. Utility of an itemset is defined as the product of its external utility and its internal utility. An itemset is called a high utility itemset if its utility is no less than a user-specified minimum utility threshold; otherwise, it is called a low-utility itemset.

6. Systolic Tree

It is not always practical or efficient to directly translate a software algorithm into a hardware architecture. Our approach is to build the tree based on the maximum node degree estimation. When the actual node degree at some point in the tree exceeds the estimated node degree, some frequent itemset will not be found. Suppose the number of items in the database is *n*, the estimated maximum node degree estimation is *K* and the estimated depth of the systolic tree is *W*. Each node in the static tree structure has *K* children. The total number of nodes in the tree is $KW + KW-1 + ... + K1 = K(KW-1) K-1$. When *K* is large, the number of children for each node is large which in turn requires each node to have a large number of interfaces.

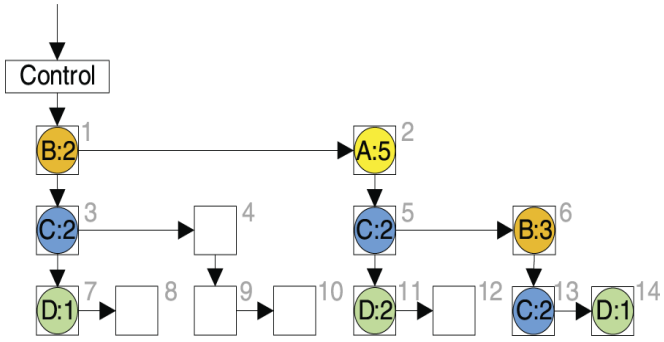


Fig. 2. The Systolic Tree Architecture

This will make the inner structure of each node very complex. To simplify the complexity of the node, we assign two instead of *K* interfaces to each node. One of the two interfaces is dedicated to the connection with its first child, the other

one is connected to its nearest sibling. To further illustrate the difference between FP-tree and systolic tree architecture, consider the FP-tree data structure. This FP-tree is different from the one defined in [2] in that it has a control node. The input and output of the whole systolic tree passes through the control node. The input can be data or control signals and the output is usually defined by the designer. The letter inside each node represents the item, and the number beside the node is the number of times that the item emerges in the transaction database. The dashed lines indicate the connections in the original FP-tree. The solid lines show the actual connections of the nodes in the systolic tree. Figure 2 shows the static systolic structure where K=2 and W=3. Each node in the systolic tree architecture is also referred to as a processing element (PE). Each PE has its local data structure and corresponding operations upon receiving signals from outside. There are three kinds of processing elements in this figure. The root PE is the control node discussed above. The PEs in the rightmost column are the counting nodes which are specifically used for frequent itemset dictation, which we will talk about later. The third kind of processing elements are the general PEs. Each node has a counterpart in the general processing elements in Fig. 1, but the converse does not hold. Each general PE has one input from its parent and two outputs to its child and siblings respectively. Each PE only has a connection with its leftmost child. If it has to send the data to its rightmost child, the data will be passed to its leftmost child and then through its siblings, passing through all children on the way. The general processing elements which do not contain any item are empty. If the items in one transaction are transferred into the architecture in an ascending order, any PE must contain a smaller item than that of its children. However this does not guarantee that the node item in the systolic tree is always greater than its left-side siblings.

7. Proposed system

Ecommerce-oriented Data mining will be a very promising area. It can automatically predict trends in customer spending, market trends which guide company to build personalized business intelligence web site, bring huge business profits. We give a high level overview of a proposed architecture for an e-commerce system with integrated data mining. To really take advantage of this domain, however, data mining must be integrated into the e-commerce systems with the appropriate data transformation bridges from the transaction processing system to the data warehouse and vice-versa. In data mining, association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases**.** A systolic tree algorithm is several times faster than the implementation of the FP-growth algorithm and UP-growth algorithm to implement in large dataset. We propose a system, designed to perform weighted rule mining for transaction dataset. Automatic weight estimation scheme is used in the system. Each item is assigned a weight value with reference to the request count and sequence. A systolic tree is an arrangement of pipelined processing elements (PE) in a multidimensional tree pattern. The role of the systolic tree as mapped in FPGA hardware is then similar to the FP-tree as used in software. The transaction items are updated into the systolic tree with candidate item matching and count update operations.

Systolic tree is used to arrange candidate sets with frequency values. Due to the limited size of the systolic tree, a transactional database must be projected into smaller ones each of which can be mined in efficiently. A high performance projection algorithm which fully utilizes the advantage of FP-growth is used. It reduces the mining time by partitioning the tree into dense and spare parts. Systolic tree based rule mining scheme is enhanced for weighted rule mining process. In this work the weight values of the items is generated using optimization algorithm, If the counts value of the items is high is considered as most important items for pattern mining with high utility and less number of scans only required for each itemset in the transactional database ,the running time of the Systolic Tree Algorithm takes less time when compare to existing tree algorithm .The performance accuracy of the proposed system is high with less number of memory space occupied and more faster. The proposed work presents a systolic tree algorithm to improve the accuracy results of the frequent itemset mining results when compare to existing frequent itemset mining results. It calculates the weight values automatically based on the request counts and the sequence value for each item in the transactional database. Automatic assignment of the weight values becomes more challenge in the existing work in order to overcome these problems in this work presents an optimization algorithm to calculate weight values for each item in the transaction database.

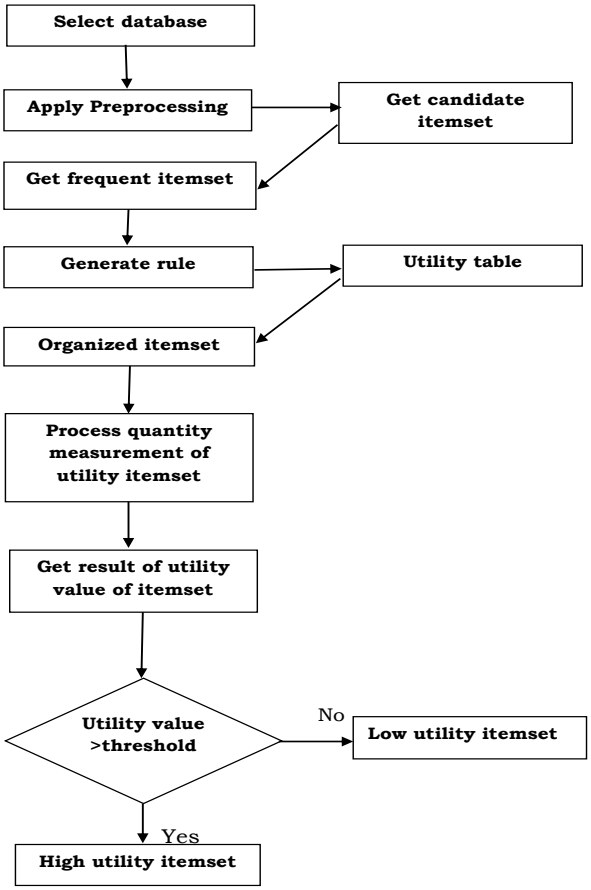
7.1 Objective

Traditional Association rule learning (ARM) model treat all the items in the database equally by only considering if an item is present in a transaction or not. Frequent itemsets may only contribute a small portion of the overall profit, whereas non-frequent itemsets may contribute a large portion of the profit. Frequency is not sufficient to answer questions, such as whether an itemset is highly profitable, or whether an itemset has a strong impact.

7.2 Scope of the project

Utility mining is used to find the high utility itemsets from databases. The aim of the project is to find the high utility itemset with the faster process for large transaction dataset using systolic tree algorithm.

7.3 System flow diagram



8. Performance analysis

The mining time for FP-growth grows larger than the systolic tree algorithm with the decrease of support threshold in different dataset. The advantage of the systolic tree over FP-growth becomes dramatic with long frequent patterns, which is challenging to the algorithms that mine the complete set of frequent patterns.

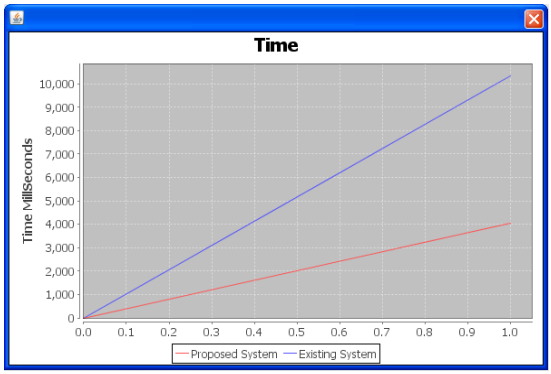


Fig.3 Comparison of proposed and existing system

In retrospect, the systolic tree algorithm removes the most frequent items and mines them in the projected databases. Compared with the original FP-growth algorithm, the systolic tree algorithm reduces the runtime by mining the dense part of the FP-tree in hardware and the sparse part in software simultaneously. However, it introduces the overhead of database projection. If the overhead is not amortized by the runtime reduction, the systolic tree algorithm is slower than the original FP-growth algorithm. One can expect that the mining time will decrease with more frequent patterns mined in hardware when the size of the systolic tree is fixed. , the systolic tree algorithm removes the most frequent items and mines them in the projected databases. Hence more patterns will be mined in the systolic tree. It will be more efficient than the FP algorithm that can be used for the mining. Also the systolic tree algorithm reduces the runtime.

6. CONCLUSION

The Mining high utility presents a novel efficient frequent itemset mining based on the creation of systolic tree with automatic weight generation .The automatic weight generation process uses frequency and count values of the items in the transaction. The weight values are automatically generated for each items using optimization algorithm for mining the high utility of the itemset for each number of the items in the transaction database. The systolic tree structure, the items values of each information are stored in the data structure for each frequent itemset, threshold are determined using existing algorithm and thus improves the itemset mining results in the larger transactional database. The proposed system overcomes the problem of the overestimation and underestimation utility problem for each utility mining.

References

[1] Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Database. In: ACM SIGMOD International Conference on Management of Data (1993) .
[2] Yao, H., Hamilton, H.J., Buzz, C. J., "A Foundational Approach to Mining Itemset Utilities from Databases", In: 4th SIAM International Conference on Data Mining, Florida USA (2004).
[3] "A Two-Phase Algorithm for Fast Discovery of High Utility Itemsets", Ying Liu, Wei-Keng Liao, and Alok Choudhary, Northwestern University, Evans
[4] "CTU-Mine: An Efficient High Utility Itemset Mining Algorithm Using the Pattern Growth Approach" In: Seventh International Conference on Computer and Information Technology (2007).
[5] J.Hu, A. Mojsilovic, ―High utility pattern mining: A method for discovery of high utility itemsets‖, in: pattern recognition. PP: 3317-3324, 2007.
[6] A.Erwin, R.P. Gopalan, and N.R. Achuthan, "Efficient Mining of High Utility Itemsets from Large Datasets", T. Washio et al. (Eds.): PAKDD2008, LNAI 5012, pp. 554–561, 2008. © Springer-Verlag Berlin Heidelberg 2008.
[7] Y.-C. Li, j,-s. Yeh, and C.-C. Chang, ―Isolated Items Discarding Strategy for Discovering High Utility Itemsets, ‖ Data and Knowledge engg. pp: 198-217, 2008.
[8] Liu Jian-Ping, Wang Ying Fan-Ding, ‖Incremental Mining algorithm Pre-FP in Association Rule Based on FP-tree‖, Networking and Distributed Computing, International Conference, pp: 199-203, 2010.
[9] Ahmed CF,Tanbeer SK,Jeong B-S, Lee Y-K (2011) ―HUC-Prune: An Efficient Candidate Pruning Technique to mine high utility patterns‖ Appl Intell PP: 181–198, 2011.
[10] Shih-Sheng Chen, Tony Cheng-Kui Huang, Zhe-Min Lin, ―New and efficient knowledge discovery of partial periodic patterns with multiple minimum supports‖, The Journal of Systems and Software 84, pp. 1638–1651, 2011, ELSEVIER.
[11] Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer, Byeong-Soo Jeong, Young-Koo Lee a,Ho-Jin Choi(2012) ―Single-pass incremental and interactive mining for weighted frequent patterns‖, Expert Systems with Applications 39 pp.7976– 7994, ELSEVIER 2012.
[12] "UP-Growth: An Efficient Algorithm or High Utility Itemset Mining", Vincent S. Tseng, Cheng-Wei Wu, Bai-En Shie, and Philip S. Yu. University of Illinois at Chicago, Chicago, Illinois, USA, 2010.
[13] Mengchi Liu Junfeng Qu, "Mining High Utility Itemsets without Candidate Generation", 2012.
[14] "FHM: Faster High-Utility Itemset Mining using Estimated Utility Co-occurrence Pruning", Philippe Fournier-Viger1, Cheng-Wei Wu 2014.
[15] Smita R. Londhe,, Rupali A. Mahajan,, Bhagyashree J. Bhoyar,"Overview on Methods for Mining High Utility Itemset from Transactional Database", International Journal of Scientific Engineering and Research (IJSER), Volume 1 Issue.4,December2013