IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 11, November 2015

Survey of Efficient Algorithms in Data Mining For High Utility

V. Keerthy[1], Mrs. B. Buvaneswari[2]

M.E., Computer Science and Engineering, K.L.N., Madurai, India[1]

Associate Professor, CSE, K.L.N., Madurai, India[2]

Abstract: Data mining is a wide spreading research topic with its frequent applications in online e-business and web click stream analysis. Mining high utility itemsets from a transactional database relates to the discovery of itemsets with high utility like profits or gains. Efficient discovery of frequent itemsets in large datasets is a crucial task of data mining. From the past few years many methods have been proposed for generating high utility patterns, by this there are some problems as producing a large number of candidate itemsets for high utility itemsets. In high utility itemset mining, the profit values for every item are considered. Generating high utility itemsets from a set of transactions in horizontal data format is a common practice. We hereby present the study of issues related to the different structures used and algorithms for mining the high utility itemsets.

Keywords: Data mining; frequent itemset; high utility itemset; transactional database.

I. INTRODUCTION

Data mining is a booming area of research in today's era. Data mining helps to produce profitable conclusions from unstructured and structured data. It is concerned with examining of large volumes of data to automatically find interesting similarities or relations which is proportional to better understanding of the underlying processes. Data mining actions use combination of techniques from database like artificial intelligence, statistics, and technologies based machine learning. Data mining is symbolized as knowledge mining from data.

Utility Mining is among one of the most difficult data mining activity which is the mining of high utility itemsets efficiently. Discovering the itemset with high utilities is called as Utility Mining. A high utility itemset is an itemset which is used frequently and is a profitable itemset, also it is measured according to user preference utility or other expressions [1,5]. The researchers came up with the idea of utility based mining which involves a user to freely express his or her view for the usefulness of itemsets as utility values and among them find the high utility values greater than threshold due to the limitations of frequent and rare itemsets. The term utility is the quantitative measure of user preference that is user's view about the utility value of itemset.

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, Web click-stream analysis, and mobile environments. In view of this, utility mining emerges as an important topic in data mining field. 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 user [3, 4, 6]. Utility of items in a transaction database consists of two aspects:

- ✓ The importance of distinct items, which is called external utility.
- ✓ The importance of items in transactions, which is called internal utility.

Mining high utility itemsets from databases refers to discovering itemsets with high profits or high utility values. The meaning of itemset utility is profitability, characteristics or importance of an item in user's point of view or users need. A high utility itemset can be elaborated as: A bunch of itemsets in a transactional database. This itemset in a transactional database includes two concepts: Firstly, itemsets in a single transaction are called Internal utility and Secondly, itemsets in multiple transactions are called External utility.

High utility itemsets mining is growing with the more innovative mining techniques with wider applications are being developed. Mining high utility itemsets from transactional databases is very important and has wide range of applications like online e-commerce management, website click stream analysis, mobile commerce environment planning, business promotion in chain hypermarkets, cross marketing in retail stores[2]. Frequent itemsets are the itemsets that appear frequently in the transactions. Frequent itemsets are the itemsets that occur frequently in the transaction data set. The goal of frequent itemset mining is to identify all the itemsets in a transaction dataset. The itemsets which appear frequently in the transactions are called frequent itemsets.

In the early days, the size of the database and the generation of a reasonable amount of frequent itemsets were considered as the most costly aspects of frequent

itemset mining, and the most energy went into minimizing the number of scans through the database. However, if the minimal support threshold is set too low, or the data is highly correlated, the number of frequent itemsets itself can be prohibitively large. To overcome this problem, several proposals have been made to construct a concise representation based on lossless compression methods [1].

II. RELATED WORK

Vincent S. Tseng, Cheng-Wei Wu, Philippe Fournier-Viger, and Philip S. Yu [1] proposed a novel framework for mining closed high utility itemsets (CHUIs), which serves as a compact and lossless representation of HUIs. This paper proposed three efficient 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. To recover all HUIs from the set of CHUIs, authors proposed a method called DAHU (Derive All High Utility Itemsets) and that to without accessing the original database. Authors claimed that this technique achieves a massive reduction in the number of HUIs. AprioriHC-D and AprioriHC both algorithms can't perform well on dense databases when min_utility is low since they suffer from the problem of a large amount of candidates.

Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer, Byeong-Soo Jeong, and Young-Koo Lee [2] focused on incremental and interactive data mining because this provides the ability to use previous data structures. They proposed three tree structures and claimed that these structures efficiently perform incremental and interactive HUP (High Utility Pattern) mining. This reduces the calculations when a minimum threshold is changed or a database is updated. One of the tree structures, Incremental HUP Lexicographic Tree (IHUPL-Tree), is arranged according to an item's lexicographic order. It can capture the incremental data without any restructuring operation. The second tree structure is the IHUP Transaction Frequency Tree (IHUPTF-Tree). This is simple and easy to construct and handle. In this tree the items are arranged according to their transaction frequency. It does not require any restructuring operation even when the data base is incrementally updated. Authors have achieved the less memory consumption. The mining time is reduced by designing the IHUP-Transaction-Weighted Utilization Tree (IHUPTWU-Tree). This tree is based on the TWU value of items in descending order.

Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu [3] proposed two algorithms utility pattern growth (UP-Growth) and UP-Growth+, for mining high utility itemsets with a set of effective strategies for pruning candidate itemsets. A tree-based data structure named utility pattern tree (UP-Tree) is maintained for the information of high utility itemsets such that candidate itemsets are generated with only two scans of database. The authors then proposed two efficient algorithms named UP-Growth and UP-Growth+ for mining high utility itemsets from transaction databases. It is found that the runtime is improved especially when databases contain lots of long transactions. HUCtree to solve the problems of large number of candidates and multiple times of database scanning. This is followed by the algorithms to HUI mine, but these algorithms discover HTWUIs in a pattern-growth approach. Still, the problem of huge memory usage for constructing and visiting conditional trees is unavoidable.

Chun-Jung Chu, Vincent S. Tseng, Tyne Liang [**4**] considered the database where the utility values for the items could be negative. They have proposed the method HUINIV (High Utility Itemsets with Negative Item Values)-Mine and claimed that this method can effectively identify high utility itemsets by generating fewer high transaction-weighted utilization itemsets such that the execution time can be reduced substantially in mining the high utility itemsets. They also claimed that memory requirement less and there is less CPU I/O. This HUINIV-Mine algorithm is based on principle of the Two-Phase algorithm and augments negative item value.

Hua-Fu Li, Hsin-Yun Huang, Suh-Yin Lee[5] propose two efficient one pass algorithms MHUI-BIT and MHUI-TID for mining high utility itemsets from data streams within a transaction sensitive sliding window. To improve the efficiency of mining high utility itemsets two effective representations of an lexicographical tree-based summary data structure and itemset information were developed.

Sen Su, Shengzhi Xu, Xiang Cheng, Zhengyi Li, and Fangchun Ya[6] proposed a differentially private FIM algorithm based on the FP-growth algorithm, which is referred to as PFP-growth. The PFP-growth algorithm consists of a preprocessing phase and a mining phase. In the pre-processing phase, to improve the utility and privacy tradeoff, a novel smart splitting method is proposed to transform the database. For a given database, the preprocessing phase needs to be performed only once. In the mining phase, to offset the information loss caused by transaction splitting, we devise a run-time estimation method to estimate the actual support of itemsets in the original database. In addition, by leveraging the downward closure property, we put forward a dynamic reduction method to dynamically reduce the amount of noise added to guarantee privacy during the mining process. Through formal privacy analysis, we show that our PFP-growth algorithm is differentially private.

Vincent S. Tseng, Senior Member, Cheng-Wei Wu, Philippe Fournier-Viger, Philip S. Yu[7] Proposed a new framework for top-k high utility itemset mining, where k is the desired number of HUIs to be mined. Two types of efficient algorithms named TKU (mining Top-K Utility itemsets) and TKO (mining Top-K utility itemsets in one phase) are proposed for mining such itemsets without the need to set min_util. provide a structural comparison of the two algorithms with discussions on their advantages and limitations. Empirical evaluations on both real and synthetic datasets show that the performance of the proposed algorithms is close to that of the optimal case of state-of-the-art utility mining algorithms.where k is the desired number of high utility itemsets to be mined.

We present in the following table some prominent works in short along with our findings.

No	Author Name	Basic Concept	Claims by Author	Remarks
1	Vincent S. Tseng, Cheng- Wei Wu, Philippe Fournier-Viger, and Philip S. Yu, 2015	Closed high utility itemset, lossless and concise representation	High utility itemset can be compacted after pruning the database.	AprioriHC-D and AprioriHC both algorithms can't perform well on dense databases when min_utility is low since they suffer from the problem of a large amount of candidates.
2	Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer, Byeong-Soo Jeong, and Young-Koo Lee, 2009	Incremental mining, Interactive mining	Three new structures with the "build once mine many" is suitable incremental database	Authors used pattern growth approach, which avoids the problem of level wise candidate generation
3	Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu, 2013	Utility mining, External utility and Internal utility	Tree based data structures can be used to store the candidate itemsets.	Improvement in the run time especially when database contains lots of long transactions.
4	Chun-Jung Chu a, Vincent S. Tseng b, Tyne Liang, 2009	Negative values for utilities if itemsets are considered	Mining for negative item values is also utility mining	The critical requirements of temporal and spatial efficiency for mining high utility itemsets with negative item values are met. High scalability in dealing with large databases is achieved.
5	Hua-Fu Li, Hsin-Yun Huang, Suh-Yin Lee, 2010	Mining High-Utility Itemsets based on BIT vector	Adapted approaches of algorithms MHUI-BIT and MHUI-TID are developed to discover high-utility itemsets with negative item profits from data streams.	To improve the efficiency of mining high utility itemsets two effective representations of an extended lexicographical tree-based summary data structure and itemset information were developed
6	Sen Su, Shengzhi Xu, Xiang Cheng, Zhengyi Li, and Fangchun Ya,2015	Differentially private FIM algorithm	PFP-growth algorithm consists of pre-processing phase and a mining phase.	A novel smart splitting method is proposed to transform the database. For a given database, the pre-processing phase needs to be performed only once.
7	Vincent S. Tseng, Cheng-Wei Wu, Viger, Philip S. Yu, 2015	Framework for top-k high utility itemset mining	Two types of efficient algorithms named TKU (mining Top-K Utility itemsets) and TKO (mining Top-K utility itemsets in one phase)	Empirical evaluations on both real and synthetic datasets show that the performance of the proposed algorithms is close to that of the optimal case of state-of-the-art utility mining algorithms.where k is the desired number of high utility itemsets to be mined.

III. CONCLUSION

This paper demonstrates a survey on different High Utility Itemset mining algorithms that were proposed by researchers earlier for better development in the field of Data Mining. The multiple algorithms discussed above will be of great use for developing a new improved technique for mining high utility item sets which is efficient and effective. In future we will be developing an algorithm for Mining Maximal High Utility Itemsets from Transactional Databases

REFERENCES

[1] Vincent S. Tseng, Cheng-Wei Wu, Philippe Fournier-Viger, and Philip S. Yu, "Efficient Algorithms for Mining the Concise and Lossless Representation of High Utility Itemsets", IEEE Transactins on Knowledge and Data Engineering, Vol. 27, No. 3, 2015.

[2] Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer, Byeong-Soo Jeong, and Young-Koo Lee, "Efficient Tree Structures for High Utility Pattern Mining in Incremental Databases", IEEE Transactions on Knowledge and Data Engineering, Vol.21, No 12, December 2009, pp 1708-1721

[3] Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu, "Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases", IEEE Transactions on Knowledge and Data Engineering, Vol.25, No. 8, AUGUST 2013, pp 1772-1786.

[4] Chun-Jung Chu, Vincent S. Tseng, Tyne Liang, "An efficient algorithm for mining high utility itemsets with negative item values in large databases", Elsevier, 2009. doi:10.1016/j.amc.2009.05.066.

[5] Hua-Fu Li, Hsin-Yun Huang, Suh-Yin Lee, "Fast and memory efficient mining of high-utility itemsets from data streams: with and without negative item profits", Springer, 2010. DOI 10.1007.

[6] Sen Su, Shengzhi Xu, Xiang Cheng, Zhengyi Li, and Fangchun Ya, "Differentially Private Frequent Itemset Mining via Transaction Splitting", IEEE Transactions on Knowledge and Data Engineering, Vol.27, No 7, July 2015.

[7] Vincent S. Tseng, Cheng-Wei Wu, Viger, Philip S. Yu,, " Efficient Algorithms for Mining Top-K High Utility Itemsets", IEEE Transactions on Knowledge and Data Engineering, DOI 10.1109/TKDE.2015.