Volume 6, Issue 12, December 2016 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Survey Paper on Frequent Itemset Mining Techniques

Jayant Kayastha*, **Prof. N. R. Wankhade**

Department of Computer Engineering, KCT's Late G.N. Sapkal College of Engineering,
Maharashtra, India

Abstract— High utility itemsets (HUIs) mining is an emerging topic in data mining, which refers to discovering all itemsets having a utility meeting a user-specified minimum utility threshold min_util. Frequent sets play a necessary role in several data processing tasks that attempt to realize fascinating patterns from databases, like association rules. The mining of association rules is one in all the foremost widespread issues of these. Compared to mining from a static dealings dataset, the streaming case has much more info to trace and much bigger complexness to manage. In-frequent things will become frequent afterward and hence cannot be unheeded. In this paper we present state of art for Frequent Itemset Mining algorithms. The survey clearly lists the disadvantages of the available algorithms for HUI and proposes alternatives to overcome the flaws.

Keywords— Utility mining, high utility itemset mining, top-k pattern mining, top-k high utility itemset mining..

I. INTRODUCTION

Frequent itemsets play an essential role in many data mining tasks that try to find interesting patterns from databases, such as association rules, correlations, sequences, episodes, classifiers, clusters and many more of which the mining of association rules is one of the most popular problems. The original motivation for searching association rules came from the need to analyze so called supermarket transaction data, that is, to examine customer behavior in terms of the purchased products. Association rules describe how often items are purchased together. For example, an association rule "beer) chips (80%)" states that four out of five customers that bought beer also bought chips. Such rules can be useful for decisions concerning product pricing, promotions, store layout and many others. Since their introduction in 1993 by Argawal et al. [1], the frequent itemset and association rule mining problems have received a great deal of attention. Within the past decade, hundreds of research papers have been published presenting new algorithms or improvements on existing algorithms to solve these mining problems more efficiently.

Frequent Itemset Mining (FIM) [2], [3], [4] is a fundamental research topic in data mining. However, the traditional FIM may discover a large amount of frequent but low-value itemsets and lose the information on valuable itemsets having low selling frequencies. Hence, it cannot satisfy the requirement of users who desire to discover itemsets with high utilities such as high profits. To address these issues, utility mining [5], [6], [7]emerges as an important topic in data mining and has received extensive attention in recent years. In utility mining, each item is associated with a utility (e.g. unit profit) and an occurrence count in each transaction (e.g. quantity). The utility of an itemset represents its importance, which can be measured in terms of weight, value, quantity or other information depending on the user specification. An itemset is called high utility itemset (HUI) if its utility is no less than a user-specified minimum utility threshold min_util. HUI mining is essential to many applications such as streaming analysis [8], [9], [10], market analysis, mobile computing and biomedicine. However, efficiently mining HUIs in databases is not an easy task because the downward closure property [2], [4] used in FIM does not hold for the utility of itemsets. In other words, pruning search space for HUI mining is difficult because a superset of a low utility itemset can be high utility.HUI mining is a gaining lot of interest and has developed a lot till date, in this paper we focuses on deep study of the existing mining algorithms and list out the techniques which are used to get the results efficiently and also analysis is done so as to improvise the existing algorithms to improve the results

The rest of the paper is organized as, section 2 provides literature study of the existing algorithms, whereas the flaws of the existing systems are shown in section 3 and the conclusion is draw in last section.

II. LITERATURE SURVEY

Existing In recent years, many algorithms for high utility itemset mining has been proposed In [11] author proposes a novel frequent pattern tree (FP-tree) structure, which is an extended prefix-tree structure for storing compressed, crucial information about frequent patterns, and develop an efficient FP-tree-based mining method, FP-growth, for mining *the complete set of frequent patterns* by pattern fragment growth. Efficiency of mining is achieved with three techniques: (1) a large database is compressed into a highly condensed, much smaller data structure, which avoids costly, repeated database scans, (2) our FP-tree-based mining adopts a pattern fragment growth method to avoid the costly generation of a large number of candidate sets, and (3) a partitioning-based, divide-and-conquer method is used to decompose the mining task into a set of smaller tasks for mining confined patterns in conditional databases, which dramatically reduces the search space.

Authors in [12] propose a new mining task: mining top-k frequent closed patterns of length no less than min_/spl lscr/, where k is the desired number of frequent closed patterns to be mined, and min_/spl lscr/ is the minimal length of each pattern. An efficient algorithm, called TFP, is developed for mining such patterns without minimum support. Two methods, closed-node-count and descendant-sum are proposed to effectively raise support threshold and prune FP-tree both during and after the construction of FP-tree. During the mining process, a novel top-down and bottom-up combined FP-tree mining strategy is developed to speed-up support-raising and closed frequent pattern discovering. In addition, a fast hash-based closed pattern verification scheme has been employed to check efficiently if a potential closed pattern is really closed.

In recent years, high utility itemset mining has received lots of attention and many efficient algorithms have been proposed, authors in [13] present a Two-Phase algorithm to efficiently prune down the number of candidates and can precisely obtain the complete set of high utility itemsets. In the first phase, we propose a model that applies the ”transaction-weighted downward closure property” on the search space to expedite the identification of candidates. In the second phase, one extra database scan is performed to identify the high utility itemsets and they also parallelize our algorithm on shared memory multi-process architecture using Common Count Partitioned Database (CCPD) strategy.

In [14] authors propose three novel tree structures to efficiently perform incremental and interactive HUP mining. The first tree structure, 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), which obtains a compact size by arranging items according to their transaction frequency (descending order). To reduce the mining time, the third tree, IHUP-transaction-weighted utilization tree (IHUPTWU-Tree) is designed based on the TWU value of items in descending order. Authors in [15] proposes the Isolated Items Discarding Strategy (IIDS), which can be applied to any existing level-wise utility mining method to reduce candidates and to improve performance. The most efficient known models for share mining are ShFSM and DCG, which also work adequately for utility mining as well. By applying IIDS to ShFSM and DCG, the two methods FUM and DCG+ were implemented, respectively.

Authors in [16] we propose an efficient algorithm, namely UP-Growth (Utility Pattern Growth), for mining high utility itemsets with a set of techniques for pruning candidate itemsets. The information of high utility itemsets is maintained in a special data structure named UP-Tree (Utility Pattern Tree) such that the candidate itemsets can be generated efficiently with only two scans of the database. The performance of UPGrowth was evaluated in comparison with the state-of-the-art algorithms on different types of datasets. Authors in [17] proposes a high utility itemset growth approach that works in a single phase without generating candidates. Our basic approach is to enumerate itemsets by prefix extensions, to prune search space by utility upper bounding, and to maintain original utility information in the mining process by a novel data structure. Such a data structure enables us to compute a tight bound for powerful pruning and to directly identify high utility itemsets in an efficient and scalable way.

Authors in [18] we propose an algorithm, called HUI-Miner (High Utility Itemset Miner), for high utility itemset mining. HUI-Miner uses a novel structure, called utility-list, to store both the utility information about an itemset and the heuristic information for pruning the search space of HUI-Miner. By avoiding the costly generation and utility computation of numerous candidate itemsets, HUI-Miner can efficiently mine high utility itemsets from the utility-lists constructed from a mined database. These algorithms can be generally categorized into two types: twophase and one-phase algorithms. The main characteristic of two-phase algorithms is that they consist of two phases. In the first phase, they generate a set of candidates that are potential high utility itemsets. In the second phase, they calculate the exact utility of each candidate found in the first phase to identify high utility itemsets. Two-Phase, IHUP, IIDS and UP-Growth are two-phase based algorithms. UPGrowth is one of the state-of-the-art two-phase algorithms, which incorporates four effective strategies DGU, DGN, DLU and DLN for pruning candidates in the first phase. One the contrary, the main characteristic of one-phase algorithms is that they discover high utility itemsets using only one phase and produce no candidates. d2HUP and HUI-Miner are onephase algorithms. d2HUP transforms a horizontal database into a tree-based structure called CAUL [17] and adopts a patterngrowth strategy to directly discover high utility itemsets in databases. HUI-Miner considers a database of vertical format and transforms it into utility-lists [18]. The utility-list structure used in HUI-Miner allows directly computing the utility of generated itemsets in main memory without scanning the original database.

Many studies are planned to mine completely different sorts of top-k patterns, like top-k frequent item sets [19], [20], [21], top-k frequent closed itemsets [19], [22], top-k closed sequent patterns [23], top-k association rules [25], top-k sequent rules [24], top-k correlation patterns [26], [27], [28] and top-k circular function similarity attention-grabbing pairs [29]. What distinguishes every top-k pattern mining algorithmic program is that the form of patterns discovered, as well because the knowledge structures and search methods that area unit employed. for instance, some algorithms [24], [25] use a rule expansion strategy for locating patterns, whereas others believe a pattern-growth search victimization structures like FP-Tree [20], [21], [22]. The selection of knowledge structures and search strategy has an effect on the potency of a top-k pattern mining algorithm in terms of each memory and execution time. However, the higher than algorithms discover top-k patterns according to ancient lives rather than the utility measure. As a consequence, they will miss patterns yielding high utility.

III. DISCUSSION

Although many studies have been devoted to HUI mining, it is difficult for users to choose an appropriate minimum utility threshold in practice. Depending on the threshold, the output size can be very small or very large.

Besides, the choice of the threshold greatly influences the performance of the algorithms. If the threshold is set too low, too many HUIs will be presented to the users and it is difficult for the users to comprehend the results. A large number of HUIs also causes the mining algorithms to become inefficient or even run out of memory, because the more HUIs the algorithms generate, the more resources they consume. On the contrary, if the threshold is set too high, no HUI will be found. To find an appropriate value for the min_util threshold, users need to try different thresholds by guessing and re-executing the algorithms over and over until being satisfied with the results. This process is both inconvenient and time-consuming. To solve the issues of the existing system we proposed a new architecture which is show in fig 1.

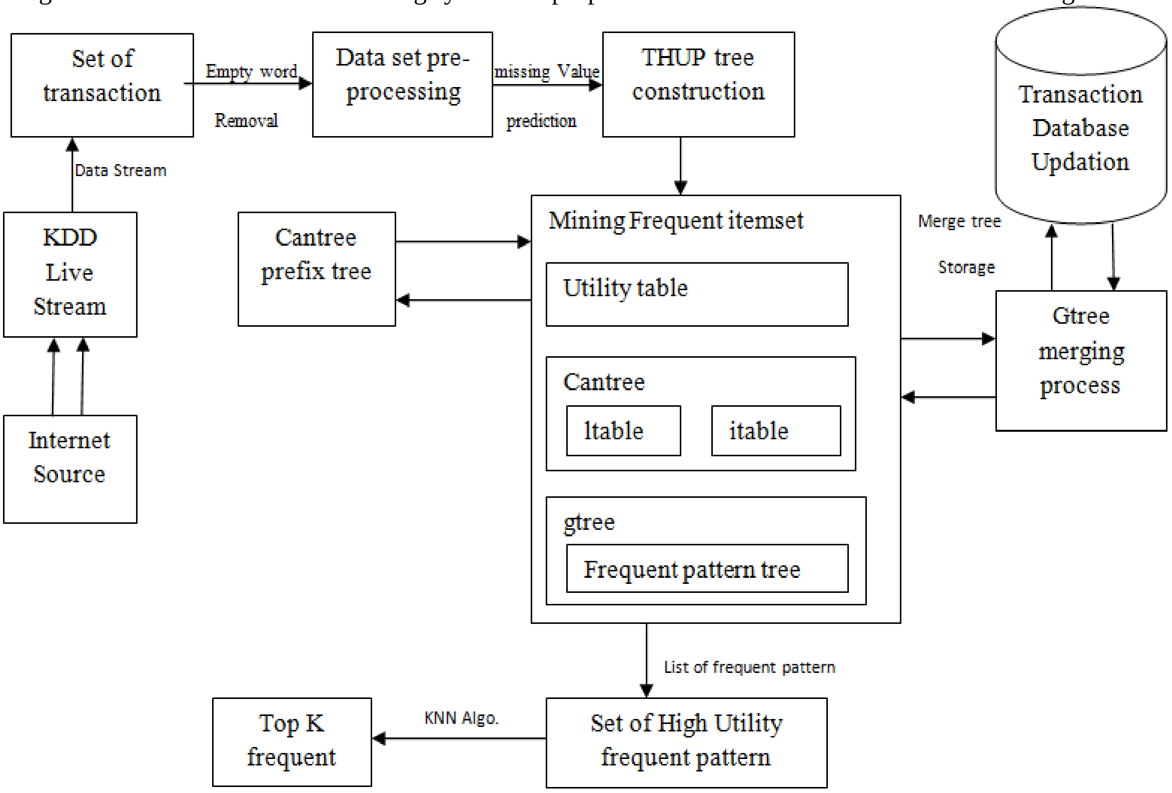

Fig1. Proposed System Architecture

IV. CONCLUSIONS

Throughout the last decade, a lot of people have implemented and compared several algorithms that try to solve the frequent itemset mining problem as efficiently as possible. Unfortunately, only a very small selection of researchers put the source codes of their algorithms publicly available such that fair empirical evaluations and comparisons of their algorithms become very difficult. Moreover, we experienced that different implementations of the same algorithms could still result in significantly different performance results. In this survey, we presented an in depth analysis of a lot of algorithms which made a significant contribution to improve the efficiency of frequent itemset mining and we also proposed our approach.

ACKNOWLEDGMENT

My sincere thanks go to KCTs Late G.N. Sapkal College of Engineering for providing a strong platform to develop my skill and capabilities. We would like to thanks all those who directly or indirectly help us in presenting the paper. I hereby take this opportunity to express our heartfelt gratitude towards the people whose help is very useful to complete our project. I would like to express our heartfelt thanks to my guide Prof. N. R. Wankhade whose experienced guidance became very valuable for me.

REFERENCES

[1] R. Agrawal, T. Imielinski, and A.N. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, volume 22(2) of SIGMOD Record, pages 207–216. ACM Press, 1993.

[2] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in Proc. Int. Conf. Very Large Data Bases, 1994, pp. 487– 499.

[3] K. Chuang, J. Huang, and M. Chen, "Mining top-k frequent patterns in the presence of the memory constraint," VLDB J., vol. 17, pp. 1321–1344, 2008.

[4] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in Proc. ACM SIGMOD Int. Conf. Manag. Data,2000, pp. 1–12.

[5] V. S. Tseng, C. Wu, B. Shie, and P. S. Yu, "UP-Growth: An efficient algorithm for high utility itemset mining," in Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2010, pp. 253–262.

[6] V. S. Tseng, C. Wu, P. Fournier-Viger, and P. S. Yu, “Efficient algorithms for mining the concise and lossless representation of high utility itemsets,” IEEE Trans. Knowl. Data Eng., vol. 27, no. 3, pp. 726–739, Mar. 1, 2015.

[7] C. Wu, P. Fournier-Viger, P. S. Yu, and V. S. Tseng, “Efficient mining of a concise and lossless representation of high utility itemsets,” in Proc. IEEE Int. Conf. Data Mining, 2011, pp. 824–833.

[8] C. Ahmed, S. Tanbeer, B. Jeong, and Y. Lee, “Efficient tree structures for high-utility pattern mining in incremental databases,” IEEE Trans. Knowl. Data Eng., vol. 21, no. 12, pp. 1708–1721, Dec. 2009.

[9] C. Lin, T. Hong, G. Lan, J. Wong, and W. Lin, “Efficient updating of discovered high-utility itemsets for transaction deletion in dynamic databases,” Adv. Eng. Informat., vol. 29, no. 1, pp. 16–27, 2015.

[10] U. Yun and H. Ryang, “Incremental high utility pattern mining with static and dynamic databases,” Appl. Intell., vol. 42, no. 2, pp. 323–352, 2015.

[11] Han, Jiawei, Jian Pei, and Yiwen Yin. "Mining frequent patterns without candidate generation." *ACM Sigmod Record*. Vol. 29. No. 2. ACM, 2000.

[12] Han, Jiawei, et al. "Mining top-k frequent closed patterns without minimum support." *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*. IEEE, 2002.

[13] Liu, Ying, Wei-keng Liao, and Alok Choudhary. “A fast high utility itemsets mining algorithm.” Proceedings of the 1st international workshop on Utility-based data mining. ACM, 2005.

[14] Ahmed, Chowdhury Farhan, et al. “Efficient tree structures for high utility pattern mining in incremental databases.” IEEE Transactions on Knowledge and Data Engineering 21.12 (2009): 1708-1721.

[15] Li, Yu-Chiang, Jieh-Shan Yeh, and Chin-Chen Chang. “Isolated items discarding strategy for discovering high utility itemsets.” Data & Knowledge Engineering 64.1 (2008): 198-217.

[16] Tseng, Vincent S., et al. “UP-Growth: an efficient algorithm for high utility itemset mining.” Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2010.

[17] Liu, Junqiang, Ke Wang, and Benjamin CM Fung. “Direct discovery of high utility itemsets without candidate generation.” 2012 IEEE 12th International Conference on Data Mining. IEEE, 2012.

[18] Liu, Mengchi, and Junfeng Qu. “Mining high utility itemsets without candidate generation.” Proceedings of the 21st ACM international conference on Information and knowledge management. ACM, 2012.

[19] K. Chuang, J. Huang, and M. Chen, “Mining top-k frequent patterns in the presence of the memory constraint,” VLDB J., vol. 17, pp. 13211344, 2008.

[20] G. Pyun and U. Yun, “Mining top-k frequent patterns with combination reducing techniques,” Appl. Intell., vol. 41, no. 1, pp. 7698, 2014.

[21] T. Quang, S. Oyanagi, and K. Yamazaki, “ExMiner: An efficient algorithm for mining top-k frequent patterns,” in Proc. Int. Conf. Adv. Data Mining Appl., 2006, pp. 436 447.

[22] J. Wang and J. Han, “TFP: An efficient algorithm for mining top-k frequent closed itemsets,” IEEE Trans. Knowl. Data Eng., vol. 17, no. 5, pp. 652663, May 2005.

[23] P. Tzvetkov, X. Yan, and J. Han, “TSP: Mining top-k closed sequential patterns,” Knowl. Inf. Syst., vol. 7, no. 4, pp. 438457, 2005

[24] P. Fournier-Viger and V. S. Tseng, “Mining top-k sequential rules,” in Proc. Int. Conf. Adv. Data Mining Appl., 2011, pp. 180194.

[25] P. Fournier-Viger, C. Wu, and V. S. Tseng, “Mining top-k association rules,” in Proc. Int. Conf. Can. Conf. Adv. Artif. Intell., 2012, pp. 6173.

[26] H. Xiong, M. Brodie, and S. Ma, “TOP-COP: Mining TOP-K strongly correlated pairs in large databases,” in Proc. IEEE Int. Conf. Data Mining, 2006, pp. 11621166.

[27] H. Xiong, P. Tan, and V. Kumar, “Mining strong affinity association patterns in data sets with skewed support distribution, in Proc. IEEE Int. Conf. Data Mining, 2003, pp. 387394.

[28] H. Xiong, P. Tan, and V. Kumar, “Hyperclique pattern discovery,” Data Mining Knowl. Discovery, vol. 13, no. 2, pp. 219242, 2006

[29] S. Zhu, J. Wu, H. Xiong, and G. Xia, “Scaling up top-k cosine similarity search,” Data Knowl. Eng., vol. 70, no. 1, pp. 6083, 2011.