Survey on Mining High Utility Itemsets without Candidate Generation

[1]Geeta Popalghat, [2]Prof. S. B. Kothari

[1,2]G. H. Raisoni College of Engineering and Management, Ahmednagar, Maharashtra, India.

[1]*geetarpopalghat@gmail.com*, [2]*s.kothari@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. Earlier work shows this on two phase candidate generation. This approach suffers from scalability issue due to the huge number of candidates. Our paper presents the efficient approach where we can generate high utility patterns in one phase without generating candidates. Here we have take experiments on linear data structure, our pattern growth approach is to search a reverse set enumeration tree and to prune search space by utility upper bounding. Also high utility patterns are identified by a closure property and singleton property. In this project we are presenting new approach which is extending these algorithms to overcome the limitations using the MapReduce framework on Hadoop. Experimental results show that the proposed algorithms, not only reduce the number of candidates effectively but also outperform other algorithms substantially in terms of runtime, especially when databases contain lots of long transactions.

Keywords: Data mining, utility mining, high utility patterns, frequent patterns, pattern mining

I. INTRODUCTION

FINDING interesting patterns has been an important data mining task, and has a variety of applications, for example, genome analysis, condition monitoring, cross marketing, and inventory prediction, where interestingness measures [17][5][7] play an important role. With frequent pattern mining [2], [3], [18] a pattern is regarded as interesting if its occurrence frequency exceeds a user specified threshold. For example, mining frequent patterns from a shopping transaction database refers to the discovery of sets of products that are frequently purchased together by customers. However, a user's interest may relate to many factors that are not necessarily expressed in terms of the occurrence frequency. For example, a supermarket manager may be interested in discovering combinations of products with high profits or revenues, which relates to the unit profits and purchased quantities of products that are not considered in frequent pattern mining. Utility mining emerged recently to address the limitation of frequent pattern mining by considering the user's expectation or goal as well as the raw data. Utility mining with the itemset share framework [19] for example, discovering combinations of products with high profits or revenues, is much harder than other categories of utility mining problems, for example, weighted itemset mining [10] and objective-oriented utility-based association mining [11]. Concretely, the interestingness measures in the latter categories observe an anti-monotonicity property, that is, a superset of an uninteresting pattern is also uninteresting. Such a property can be employed in pruning search space, which is also the foundation of all frequent pattern mining algorithms [3]. Unfortunately, the anti-monotonicity property does not apply to utility mining with the itemset share

framework. Therefore, utility mining with the itemset share framework is more challenging than the other categories of utility mining as well as frequent pattern mining. Most of the prior utility mining algorithms with the itemset share framework [4], [15] adopt a two-phase, candidate generation approach, that is, first find candidates of high utility patterns in the first phase, and then scan the raw data one more time to identify high utility patterns from the candidates in the second phase. The challenge is that the number of candidates can be huge, which is the scalability and efficiency bottleneck. Although a lot of effort has been made [4], [15] to reduce the number of candidates generated in the first phase, the challenge still persists when the raw data

contains many long transactions or the minimum utility threshold is small. Such a huge number of candidates causes scalability issue not only in the first phase but also in the second phase, and consequently degrades the efficiency. One exception is the HUIMiner algorithm [20], which is however even less efficient than two-phase algorithms when mining large databases due to inefficient join operations, lack of strong pruning, and scalability issue with its vertical data structure[16].

II. LITERATURE SURVEY

A. Depth first generation of long patterns [1]

R. Agarwal, C. Aggarwal, and V. Prasad present an algorithm for mining long patterns in databases. The algorithm finds large itemsets by using depth first search on a lexicographic tree of itemsets. The focus of this paper is to develop CPU-efficient algorithms for finding frequent itemsets in the cases when the database contains patterns which are very wide. Author refer to this algorithm as Depth Project, and it achieves more than one order of magnitude speedup over the recently proposed MaxMiner algorithm for finding long patterns. These techniques may be quite useful for applications in areas such as computational biology in which the number of records is relatively small, but the itemsets are very long. This necessitates the discovery of patterns using algorithms which are especially tailored to the nature of such domains.

B. Efficient tree structures for high utility pattern mining in incremental databases [4]

Recently, high utility pattern (HUP) mining is one of the most important research issues in data mining due to its ability to consider the nonbinary frequency values of items in transactions and different profit values for every item. On the other hand, incremental and interactive data mining provide the ability to use previous data structures and mining results in order to reduce unnecessary calculations when a database is updated, or when the minimum threshold is changed. In this paper, we propose three novel tree structures to efficiently perform incremental and interactive HUP mining. The first tree structure, Incremental HUP Lexicographic Tree ({\rm IHUP}_{{\rm {L}}}-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 ({\rm IHUP}_{{\rm {TF}}}-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 ({\rm IHUP}_{{\rm {TWU}}}-Tree) is designed based on the TWU value of items in descending order. Extensive performance analyses show that our tree structures are very efficient and scalable for incremental and interactive HUP mining.

C. ExAnte A preprocessing method for frequent pattern mining [6]

F. Bonchi, F. Giannotti, A. Mazzanti, and D. Pedreschi presents approach named ExAnte is a simple yet effective for preprocessing input data for mining frequent patterns. The approach questions established research in that it requires no trade-off between antimonotonicity and monotonicity. Indeed, ExAnte relies on a strong synergy between these two opposite components and exploits it to dramatically reduce the data being analyzed to that containing interesting patterns. This data reduction, in turn, induces a strong reduction of the

candidate patterns' search space. The result is significant performance improvements in subsequent mining. It can also make feasible some otherwise intractable mining tasks. The authors describe their technology and experiments that proved its effectiveness using different constraints on various data sets.

C. Extending the state-of-the-art of constraint-based pattern discovery [8]

In the last years, in the context of the constraint-based pattern discovery paradigm, properties of constraints have been studied comprehensively and on the basis of this properties, efficient constraint-pushing techniques have been defined. In this paper we review and extend the state-of-the-art of the constraints that can be pushed in a frequent pattern computation[9]. This paper introduce novel data reduction techniques which are able to exploit convertible anti-monotone constraints (e.g., constraints on *average* or *median*) as well as tougher constraints (e.g., constraints on *variance* or *standard deviation*)[9]. A thorough experimental study is performed and it confirms that our framework outperforms previous algorithms for convertible constraints, and exploit the tougher ones with the same effectiveness[9]. Finally, author highlight that the main advantage of this approach, i.e., pushing constraints by means of data reduction in a level-wise framework, is that different properties of different constraints can be exploited all together, and the total benefit is always greater than the sum of the individual benefits. This consideration leads to the definition of a general Apriori-like algorithm which is able to exploit all possible kinds of constraints studied so far.

E. UP-Hist tree: An efficient data structure for mining high utility patterns from transaction databases [12]

High-utility itemset mining is an emerging research area in the field of Data Mining. Several algorithms were proposed to find high-utility itemsets from transaction databases and use a data structure called UP-tree for their working. However, algorithms based on UP-tree generate a lot of candidates due to limited information availability in UP-tree for computing utility value estimates of itemsets. In this paper, author present a data structure named UP-Hist tree which maintains a histogram of item quantities with each node of the tree. The histogram allows computation of better utility estimates for effective pruning of the search space[16]. Extensive experiments on real as well as synthetic datasets show that their algorithm based on UP-Hist tree outperforms the state of the art pattern-growth based algorithms in terms of the total number of candidate high utility itemsets generated that needs to be verified.

III. PROPOSED SYSTEM

To provide the efficient solution to mine the large transactional datasets, recently improved methods presented propose two novel algorithms as well as a compact data structure for efficiently discovering high utility itemsets from transactional databases. Experimental results show that d2HUP and CAUL outperform other algorithms substantially in terms of execution time. But these algorithms further needs to be extend so that system with less memory will also able to handle large datasets efficiently. The below Fig. are shows Proposed System.

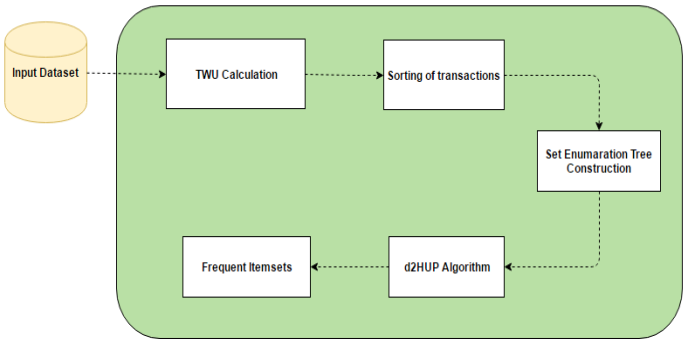

Fig. 1 Proposed System

The algorithms presented in paper are practically implemented with memory 3.5 GB, but if memory size is 2 GB or below, the performance will again degrade in case of time. In this project we are presenting new approach which is extending these algorithms to overcome the limitations using the MapReduce framework on Hadoop.

IV. CONCLUSION

Proposed a novel data structure, **utility-list**, and developed an efficient algorithm, **HUI-Miner**, for high utility itemset mining. Utility-lists provide not only utility information about itemsets but also important pruning information for HUI-Miner[16]. HUI-Miner can mine high utility itemsets without candidate generation, which avoids the costly generation and utility computation of candidates.

REFERENCES

[1] R. Agarwal, C. Aggarwal, and V. Prasad, "Depth first generation of long patterns," in Proc. ACM SIGKDD Int. Conf. Knowl. DiscoveryData Mining, 2000, pp. 108–118.

[2] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 1993, pp. 207–216.

[3] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in Proc. 20th Int. Conf. Very Large Databases, 1994, pp. 487–499.

[4] C. F. Ahmed, S. K. Tanbeer, B.-S. Jeong, and Y.-K. 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.

[5] R. Bayardo and R. Agrawal, "Mining the most interesting rules," in Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 1999, pp. 145–154.

[6] F. Bonchi, F. Giannotti, A. Mazzanti, and D. Pedreschi, "ExAnte: A preprocessing method for frequent-pattern mining," IEEE Intell. Syst., vol. 20, no. 3, pp. 25–31, May/Jun. 2005.

[7] F. Bonchi and B. Goethals, "FP-Bonsai: The art of growing and pruning small FP-trees," in Proc. 8th Pacific-Asia Conf. Adv. Knowl. Discovery Data Mining, 2004, pp. 155–160.

[8] F. Bonchi and C. Lucchese, "Extending the state-of-the-art of constraint-based pattern discovery," Data Knowl. Eng., vol. 60, no. 2, pp. 377–399, 2007.

[9] C. Bucila, J. Gehrke, D. Kifer, and W. M. White, "Dualminer: A dual-pruning algorithm for itemsets with constraints," Data Mining Knowl. Discovery, vol. 7, no. 3, pp. 241–272, 2003.

[10] C. H. Cai, A. W. C. Fu, C. H. Cheng, and W. W. Kwong, "Mining association rules with weighted items," in Proc. Int. Database Eng. Appl. Symp., 1998, pp. 68–77.

[11] R. Chan, Q. Yang, and Y. Shen, "Mining high utility itemsets," in Proc. Int. Conf. Data Mining, 2003, pp. 19–26.

[12] S. Dawar and V. Goyal, "UP-Hist tree: An efficient data structure for mining high utility patterns from transaction databases," in Proc. 19th Int. Database Eng. Appl. Symp., 2015, pp. 56–61.

[13] T. De Bie, "Maximum entropy models and subjective interestingness: An application to tiles in binary databases," Data Mining Knowl. Discovery, vol. 23, no. 3, pp. 407–446, 2011.

[14] L. De Raedt, T. Guns, and S. Nijssen, "Constraint programming for itemset mining," in Proc. ACM SIGKDD, 2008, pp. 204–212.

[15] A. Erwin, R. P. Gopalan, and N. R. Achuthan, "Efficient mining of high utility itemsets from large datasets," in Proc. 12th Pacific-Asia Conf. Adv. Knowl. Discovery Data Mining, 2008, pp. 554–561.

[16] P. Fournier-Viger, C.-W. Wu, S. Zida, and V. S. Tseng, "FHM: Faster high-utility itemset mining using estimated utility cooccurrence pruning," in Proc. 21st Int. Symp. Found. Intell. Syst., 2014, pp. 83–92.

[17] L. Geng and H. J. Hamilton, "Interestingness measures for data mining: A survey," ACM Comput. Surveys, vol. 38, no. 3, p. 9, 2006.

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

[19] R. J. Hilderman, C. L. Carter, H. J. Hamilton, and N. Cercone, "Mining market basket data using share measures and characterized itemsets," in Proc. PAKDD, 1998, pp. 72–86.

[20] R. J. Hilderman and H. J. Hamilton, "Measuring the interestingness of discovered knowledge: A principled approach," Intell. Data Anal., vol. 7, no. 4, pp. 347–382, 2003.

