ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 8, Issue 12, June 2019

An efficient algorithm for mining maximum utility patterns from incremental databases for product selling strategies

Mamata S. Kalas, Amruta G. Unne
Department of Computer Science & Engineering
KITs College of Engineering Kolhapur

Abstract— ***Maximum utility pattern drilling has been examined as one of the most significant topics in the data mining domain. The common pattern mining cannot enough consider the characteristics of real-world databases. Moreover, database sizes have been bigger continuously in various applications such as product sales data of retail markets and connection information of web services. The usual methods for static databases are not suitable for processing dynamic databases and extracting useful information from them. Although to maximum utility pattern drilling approaches in previous strategies uses two or more scans for maximizing utility pattern irrespective of structure. However, the approaches with multiple scans are actually not adequate. Thus an efficient algorithm for mining maximum utility patterns from continuously increasing databases with single database scan based on the structure with candidate generation is proposed. The proposed system uses the retail transactional database and to increase the efficiency of the system we add the constraints like length, attribute, etc., which helps for accurate prediction to for maximizing sale in a retail transactional database.***

Index Terms—**Temporal data, data mining, frequent item sets.**

I. INTRODUCTION

Data mining has a variety of applications in various areas like banking, trading, scientific analysis, drug and among government agencies. Data mining is widely used in Financial, Retail Industry, Data Analysis, Retail Industry, Spatial Data Analysis, Biological Data Analysis, Scientific Applications and Intrusion Detection. It is important for the forecast of forthcoming trends, judgment making, customer purchase attitudes, and market basket analysis and scam detection. It comprises various duties like association rule mining, frequent pattern mining, classification, utility mining, clustering. Data mining has been used in various domains. The main intent of data mining is to extract hidden information from a large database. Data mining tasks classified as Descriptive Mining and Predictive Mining [11]. The Descriptive Mining techniques used to find patterns that describe the data. The Predictive Mining techniques are like. To examine data and identify interesting knowledge [2],[5],[7],[10], and data mining has produced a significant contribution to data analysis. Pattern mining [2],[5] is also data mining techniques to finds meaningful information hidden in large databases as pattern forms. Although frequent pattern mining [12] has played an essential role in data mining, and the limitation is that cannot fully reflect characteristics of real-world databases to mining processes. To label this issue, utility pattern mining [11] has been active consideration of relative importance and non-binary occurrence of items. In reverse to frequent pattern mining, uses utility pattern mining to satisfy the anti-monotone property [11], the first criterion for efficient pattern mining, where no valid pattern is generated from an invalid one. For this improves mining performance by effectively reducing search space , the overestimation concept was proposed and employed in utility pattern mining, but it degrades mining performance by extracting a large number of candidate patterns.

Manuscript received: 23 May 2019
Manuscript received in revised form: 20 June 2019
Manuscript accepted: 05 July 2019
Manuscript Available online: 10 July 2019

The proposed system develops data structure to maintain increasing data efficiency including its construction and reconstruction techniques. And a mining algorithm is proposed based on constraints like minutil threshold, length, attribute etc. to mining maximum utility pattern.

II. LITERATURE SURVEY

P. Fournier-Viger , J.C.-W. Lin, R. Kiran , Y. Koh , R.Thomas [1]. In This work introduction to serial pattern mining, and a survey of recent advances and analysis opportunities stated. The paper is split into four main elements. First, the task of serial pattern mining is outlined and its applications area unit reviewed. Key ideas and language area unit introduced. Moreover, main approaches and techniques to unravel serial pattern mining issues area unit conferred. Limitations of ancient serial pattern mining approaches are highlighted, and widespread variations of the task of serial pattern mining area unit conferred. The paper conjointly presents analysis opportunities and therefore the relationship to different widespread pattern mining issues. Unil Yun, Heungmo Ryang, Gangin Lee, Hamido Fujita [2].

Author propose an algorithm for mining high utility patterns from incremental databases with one database scan

DOI:10.17605/OSF.IO/RXWUE

based on data structure without candidate generation. They sort the database according to each item utility to create the utility list for each item in order. Then mine the list by threshold value entered by user. Their experimental results with real and synthetic datasets show that the algorithm outperforms previous one phase construction methods with candidate generation [2].

C.-W. Lin , G.-C. Lan , T.-P. Hong [3].In This Association rule mining is used to mine the relationships among the occurrences item sets in a transactional database. An item is treated as a binary variable whose value is one if it appears in a transaction and zero otherwise. In real-world applications, several products may be purchased at the same time, with each product having an associated profit, quantity, and price Association-rule mining from a binary database is thus not sufficient in some applications. Utility mining was thus proposed as an extension of frequent-item set mining for considering various factors from the user. Most utility mining approaches can only process static databases and use batch processing. In real-world applications, transactions are dynamically inserted into or deleted from databases.

G. Lee, U. Yun [4]. In this paper an exact, efficient algorithm for mining uncertain frequent patterns based on novel data structures and mining techniques, which can also guarantee the correctness of the mining results without any false positives. The newly proposed list-based data structures and pruning techniques allow a complete set of uncertain frequent patterns to be mined more efficiently without pattern losses.

U. Yun , D. Kim [5]. The present study proposes an improved upper-bound approach that uses the prefix concept to create tighter upper bounds of average utility values for Item sets, thus reducing the number of unpromising Item sets for mining. Results from experiments on two real databases conclude the proposed algorithm outperforms other mining algorithms under various parameter settings.

J.Lin, Shifeng Ren, Philippe Fournier-Viger, Tzung-Pei Hong [7]. High-utility item set mining (HUIM) has become a popular data mining task, as it can reveal patterns having a high-utility, contrarily to frequent pattern mining (FIM), which focuses on discovering frequent patterns. High average utility itemset mining (HAUIM) is a variation of HUIM that provides an alternative measure, called the average utility, to select patterns by considering both their utilities and lengths. [8] [9] Gangin Lee, Unil Yun, Heungmo Ryang, Donggyu Kim[10] author propose a new tree-based erasable itemset mining algorithm for dynamic databases, which finds erasable itemsets considering the weight conditions from incremental databases. The proposed algorithm uses new tree and list data structures for performing its mining operations more efficiently. Algorithm is capable of reducing the number of mined erasable itemsets by considering the different weight information of items with in product databases. [11] [12] J. Liu, K. Wang, B.C.M. Fung [13],author proposes a algorithm that finds high utility patterns in a single phase without generating candidates. Concretely, their pattern growth approach is to search a reverse set enumeration tree and to prune search space by utility upper bounding.

III. PROBLEM STATEMENT

To design a method for constraint based maximum utility pattern mining with databases, given threshold minutil by user and addition of constraints i.e. length, attributes etc. for analyzing the patterns in decision making process in selling strategies.

IV. GENERAL PROCESS OF THE APRIORI ALGORITHM

The entire algorithm can be divided into two steps:
1) Apply minimum support to find all the frequent sets with k items in a database.
2) Use the self-join rule to find the frequent sets with k+1 items with the help of frequent k-itemsets. Repeat this process from k = 1 to the point when we are unable to apply the self-join rule.
This approach of extending a frequent itemset one at a time is called the “bottom up” approach

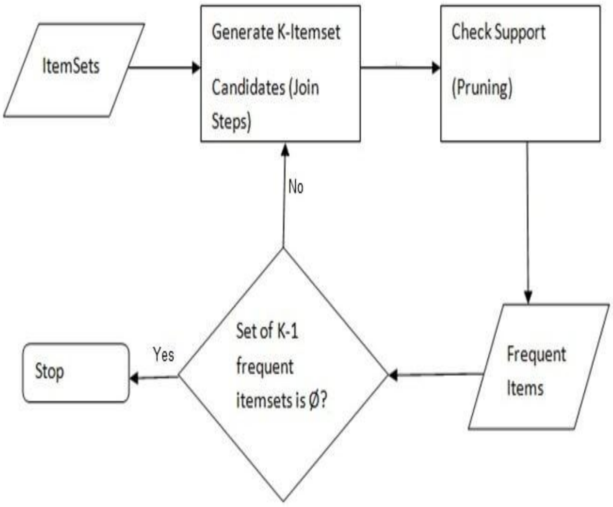
ItemSets
Generate K-Itemset Candidates (Join Steps)
Check Support (Pruning)
No
Set of K-1 frequent itemsets is Ø?
Yes
Stop
Frequent Items

Fig 1. Bottom up approach for mining association rules

Mining Association Rules

Till now, we have looked at the Apriori algorithm with respect to frequent itemset generation. There is another task for which we can use this algorithm, i.e., finding association rules efficiently. For finding association rules, we need to find all rules having support greater than the threshold support and confidence greater than the threshold confidence. But, how do we find these? One possible way is brute force, i.e., to list all the possible association rules and calculates the support and confidence for each rule. Then eliminate the rules that fail the threshold support and confidence. But it is computationally very heavy and prohibitive as the number of all the possible association rules increase exponentially with the number of items. Given there are n items in the set I, the total number of possible

DOI:10.17605/OSF.IO/RXWUE

association rules is 3n-2n + 1 + 1. We can also use another way, which is called the two-step Approach, to find the efficient association rules. The two-step approach is:

1. **Frequent itemset generation:** Find all itemsets for which the support is greater than the threshold support following the process we have already seen earlier in this article.
2. **Rule generation:** Create rules from each frequent itemset using the binary partition of frequent itemsets and look for the ones with high confidence. These rules are called candidate rules.

Let us look at our previous example to get an efficient association rule. We found that OPB was the frequent itemset. So for this problem, step 1 is already done. So, let see step 2.

All the possible rules using OPB are:

OP ->B;OB -> P; PB -> O;B -> OP; P -> OB;O ->PB

If X is a frequent item set with k elements, then there are 2k2 candidate association rules.

We will not go deeper into the theory of the Apriori algorithm for rule generation.

1. Pros of the Apriori algorithm It is an easy-to-implement and easy-to-understand algorithm. It can be used on large itemsets.
2. Cons of the Apriori Algorithm Sometimes, it may need to find a large number of candidate rules which can be computationally expensive.

Calculating support is also expensive because it has to go through the entire database.

V. MAX-UTILITY PATTERN MINING ALGORITHM

Algorithm 1 Max-Utility Pattern Mining Algorithm

```
Input: Transactional database, minutil, constraints (item, Length, date)
Output:Max Utility Patterns
1: Initialization
2: Select the database and pre-process
3: Construct the data structure
4: For each transaction in the dataset
5: Create utility list of each item
6: Sort utility list in by order
7: For increasing data repeat step2 to step6
8: Take threshold from user if required
9: Take constraints such as item, length or date
10: Mine all the patterns
11: Return high utility patterns
```

VI. THEORETICAL ANALYSIS OF ALGORITHM

The proposed method uses the transactional database with utility information as the input along with a minimum utility threshold minutil and constraint such as item, length or date from user.

TABLE I. TRANSACTIONAL NON-BINARY DATABASE

TID	Transaction	Total Utility
T1	(A,1)(D,2)(G,1)	9
T2	(A,2)(B,2)(C,1)(D,2)	22
T3	(B,3)(C,1)(D,2)(E,2)	28
T4	(A,2)(D,1)(F,1)	10

TABLE II. EXTERNAL UTILITY TABLE

Item	A	B	C	D	E	F	G
Profit	3	6	2	1	3	3	4

A. *Construction of novel list-based data structure*

The novel list-based data structure consists of list called as utility list. In utility list, information for candidate pattern is stored and maintained. Each utility list is created for a candidate itemset with the length one.

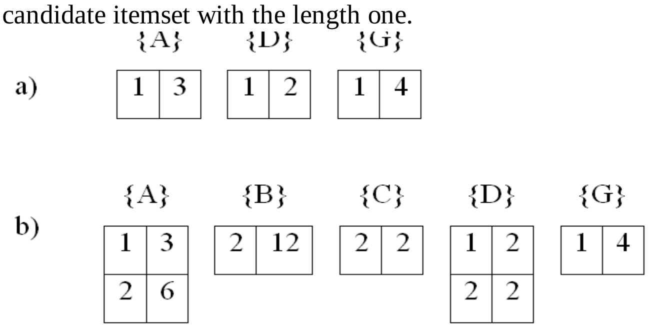

Fig 2. Construction of list-based data structure

Figure.1 a) shows the utility lists of items A,D,G after T1transaction and Figure.2 b) shows the utility lists of items A,B,C,D,G after T2 transaction. Utility list of each item contains transaction id and the cost (i.e. external utility x quantity) of that item. As transaction goes on increasing utility list gets updated.

Sorting of the utility lists is based on the two (transaction weighted utility) values of the transaction. Algorithm sort the first transaction according to initial order and result is T1’=A,D,G and create candidate patterns for A, D, and G.The next transaction is sorted as T2’=A,B,C,D,G and create candidate patterns for B and C since utility list for A, D, and G are build in first transaction. Figure 2 shows the result of transaction T1, T2, T3 and T4. Hence, their t_{wu} ascending order is G > F > E > A > B > C > D.

B. *Mining maximum utility patterns*

After creation of data structure and sorting, when user made the mining request he entered the minutil value and constraints such as length, date or item. Use of constraints reduces the search space and helps in improving efficiency of algorithm. Algorithm conducts a series of mining process recursively.

Let us assume user entered values are minutil=30 and selected constraint length=2. Our algorithm first consider utility list for G, UL (G). Utility value for G is 4 which is smaller than minutil so that pattern is not selected. Algorithm checks for all utility list with length one in sorting order. If utility sum is greater than or equal to minutil then pattern is selected as MUP. After completion of items with length one, algorithm again call mine function now with length two. This process repeats until constraint length is satisfied. the following are the output patterns for maximum util. ”#MUTIL:” appears and is followed by the utility of the itemset. For example, we show below the output file for this example.

B #MUTIL: 30
BC #MUTIL: 34
BD #MUTIL: 34

In our algorithm, addition of constraints reduces the search space which helps in finding max utility patterns faster [21].

VII. RESULT ANALYSIS

TABLE III. EXECUTION TIME ANALYSIS

Transaction (k)	Time in Ms			
	Without Constraint	Item Constraint	Date Constraint	Length Constraint
100	2800	520	860	670
200	5000	1100	1690	1525
300	6500	1490	2580	1380
500	8700	2000	3440	1500
600	10500	2523	4300	2000

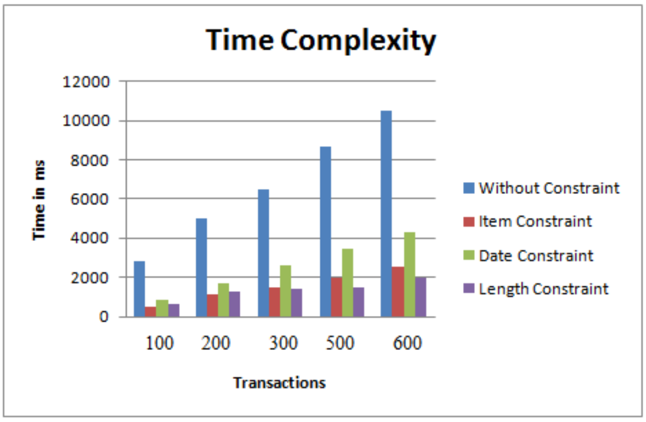

Fig 3. Execution Time analysis

VIII. CONCLUSION

Novel List-Based Maximum Utility Pattern Mining algorithm discovers constraint based max utility patterns. There are utility mining algorithms such as HUI-Miner, HUI-list- INS which discovers high utility patterns, but it requires an additional database scan, which consumes more data processing time. The novel List-based Maximum Utility Pattern Mining (LIMUP) algorithm is used which discovers high utility patterns with use of threshold and some user interested constraints like length, attribute etc. are also used which helps to find out maximum utility patterns that will be used for more fine prediction and analysis. The system is more useful in handling dynamic databases. The use of various constraints reduces execution time as well as memory requirements to several orders of magnitude. The system is more scalable and efficient than previous systems. Further work can be carried by using various dataset and it should be concentrate to minimize time factor in mining.

ACKNOWLEDGMENT

This review paper work was guided and supported by Mrs. Mamata S. Kalas. I would like to thank the guide, anonymous reviewers for their valuable and constructive comments on improving the paper.

REFERENCES

[1] P. Fournier-Viger , J.C.-W. Lin , R. Kiran , Y. Koh , R. Thomas , “A survey of sequential pattern mining”, Data Sci. Pattern Recognit. 1 (1) (2017) 54–77.

[2] Unil Yun, Heungmo Ryang, Gangin Lee, Hamido Fujita , “An efficient algorithm for mining high utility patterns from incremental databases with one database scan”. Knowledge-Based Systems, Volume 124, 15May 2017, Pages 188-206.

[3] C.-W. Lin , G.-C. Lan , T.-P. Hong , “Mining high utility itemsets for transaction deletion in a dynamic database”, Intell. Data Anal. 19 (1) (2015) 43–55 .

[4] G. Lee , U. Yun , “A new efficient approach for mining uncertain frequent patterns using minimum data structure without false positives”, Future Generation Comp. Syst. 68 (2017) 89–110 .

[5] U. Yun , D. Kim , “Mining of high average-utility itemsets using novel list structure and pruning strategy”, Future Generation Comp. Syst. 68 (2017) 346–360 .

[6] D. Meana-Llori´an , C. Garc´ıa , V. Garc´ıa-D´ıaz , B. G-Bustelo , N.Garcia-Fernandez , Sense Q: replying questions of social networks users by using a wireless sensor network based on sensor relationships, Data Sci. Pattern Recognit. 1 (1) (2017) 1–12 .

[7] J.Lin, Shifeng Ren, Philippe Fournier-Viger, Tzung-Pei Hong ,“EHAUPM: Efficient High Average-Utility Pattern Mining with Tighter Upper-Bounds”.

[8] R. Agrawal , R. Srikant , “Fast algorithms for mining association rules”,in: Proceedings of the 20th International Conference on Very Large Data Bases (VLDB 1994), 1994, pp. 4 87–4 99.

[9] L. Chen , Q. Mei , “Mining frequent items in data stream using time fading model”, Inf. Sci. 257 (1) (2014) 54–69.

[10] G. Lee , U. Yun , H. Ryang , D. Kim , “Erasable itemset mining over incremental databases with weight conditions”, Eng. Appl. of AI 52(2016) 213–234 .

[11] “Overview of Itemset Utility Mining and its Applications”

Jyothi Pillai,O. P. Vyas International General of Computer Applications Volume 5-No. 11, August 2010.

[12] G. Lee , U. Yun , H. Ryang , D. Kim , “Approximate maximal frequent pattern mining with weight conditions and error tolerance”, IJPRAI 30(6) (2016) 1–42 .

[13] J. Liu , K. Wang , B.C.M. Fung , “Mining high utility patterns in one phase without generating candidates”, IEEE Trans. Knowl. Data Eng.28 (5) (2016) 1245–1257.

[14] U. Yun , D. Kim , H. Ryang , G. Lee , K.-M. Lee , “Mining recent high average utility patterns based on sliding window from stream data”, J.Intell. Fuzzy Syst. 30 (6) (2016) 3605–3617.

[15] C.F. Ahmed , S.K. Tanbeer , B.-S. Jeong , Y.-K. Lee , H.-J. Choi ,“Single-pass incremental and interactive mining for weighted frequent patterns”, Expert Syst. Appl. 39 (9) (2012) 7976–7994 .

[16] W. Song , C. Wang , J. Li , “Binary partition for itemsets expansion in mining high utility itemsets”, Intell. Data Anal. 20 (4) (2016) 915–931.

[17] V.S. Tseng , C.-W. Wu , P. Fournier-Viger , P.S. Yu , “Efficient algorithms for mining the concise and lossless representation of high utility itemsets”, IEEE Trans. Knowl. Data Eng. 27 (3) (2015) 726–739.

[18] M. Liu , J.-F. Qu , “Mining high utility itemsets without candidate generation”, in: International Conference on Information and Knowledge Management (CIKM 2012), 2012, pp. 55–64 .

[19] J. Liu , K. Wang , B.C.M. Fung , “Direct discovery of high utility itemsets without candidate generation”, in: Proceedings of the 2012 IEEE International Conference on Data Mining (ICDM 2012), 2012,pp. 984–989 .

[20] C.-W. Lin , G.-C. Lan , T.-P. Hong , “Mining high utility itemsets for transaction deletion in a dynamic database”, Intell. Data Anal. 19 (1)(2015) 43–55.

[21] A. Deshpande, R.Deshmukh “Improving efficiency of High Utility Pattern Mining Algorithm using Constraints”, Proceedings of the International Conference on Inventive Research in Computing Applications (ICIRCA 2018) IEEE Xplore Compliant Part Number: CFP18N67- ART; ISBN: 978-1-5386-2456-2.

DOI:10.17605/OSF.IO/RXWUE