ISSN: 2321-7782 (Online)
Volume 3, Issue 12, December 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

An Innovative Approach for Mining High Utility Itemset

Saloni Lunia[1]
CSE Dept M.E 4th Sem
Jawahar Lal Institute of Techonology Borawan
Khargone M.P. India

Devendra Singh Kaushal[2]
A.P. CSE Dept
Jawahar Lal Institute of Techonology Borawan
Khargone M.P. India

Abstract: ***Utility item set mining is an important data mining techniques and an extension of frequent pattern mining. The utility is a measure of how useful or profitable an itemset X is. The utility of an itemset X, i.e., u(X), which is the sum of the all utilities of itemset X in all the transactions containing X. An itemset X is called a high utility itemset if and only if u(X) greater than or equal to minimum utility, where minimum utility is a user defined minimum utility threshold. The main objective of high-utility itemset mining is to find all those itemsets having utility greater or equal to user- defined minimum utility threshold. Several algorithms have been proposed in past year. In This paper we proposed an efficient approach which reduces search space and number of candidate's generations.***

Keywords: Utility, Candidates, Transactions, Thresholds, Itemset

I. INTRODUCTION

Mining high utility itemsets thus upgrades the standard frequent itemset mining framework as it employs subjectively defined utility instead of statistics-based support measure. User-defined utility is based on information not available in the transaction dataset. It often reflects user preference and can be represented by an external utility table or utility function. Utility table (or function) defines utilities of all items in a given database (we can also treat them as weights). Besides subjective external utility we also need transaction dependent internal utilities (e.g. quantities of items in transactions). Utility function we use to compute utility of an itemset takes into account both internal and external utility of all items in a itemset. The most usual form that is also used in this paper is defined as a sum of products of internal and external utilities of present items[1,2,3].

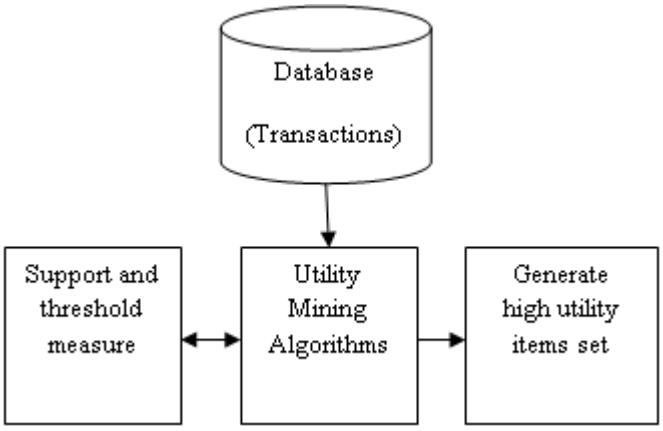

Figure 1 Architecture of utility Item set mining

II. RELATED TERMINOLOGY

Consider a simple example of transactional database

Table 1 Transactional data set

TID & ITEM	A	B	C	D	E
T1	0	0	18	0	1
T2	0	6	0	1	1
T3	2	0	1	0	1
T4	1	0	0	1	1
T5	0	0	4	0	2
T6	1	1	0	0	0
T7	0	10	0	1	1
T8	3	0	25	3	1
T9	1	1	0	0	0
T10	0	6	2	0	2

The utility table, the right column displays the profit of each item per unit in dollars

Table 2 Profit table

ITEM	PROFIT($)(Per Unit)
A	3
B	10
C	1
D	6
E	5

A. External Utility: - The external utility of an item i_p is a numerical value y_p defined by the user. It is transaction independent and reflects importance (usually profit) of the item. External utilities are stored in a utility table. For example, external utility of item B in Table 2) is 10.

B. Internal Utility:-The internal utility of an item i_p is a numerical value x_p which is transaction dependent. In most cases it is defined as the quantity of an item in transaction. For example, internal utility of item E in transaction T5 is 2 in table 1.

C.The utility of item: - The utility of item i_p in transaction T is the quantitative measure computed with utility function from above definition $u(i_p, T) = f(x_p, y_p)$, $i_p \in T$. For example: utility of item E in transaction T5 is $2 * 5 = 10$.

The utility of itemset S in transaction T: - The utility of itemset S in transaction T is defined as

$$u(S,T) = \sum_{i_p \in S} u(i_p, T),\ S \subseteq T$$

For example utility of itemset {B, E} in transaction T2 is

u({B,E} , T2) = u({B} , T2) + u({E} , T2) = $6 * 10 + 1 * 5 = 65$.

D. The utility of item i_p in itemset S: - The utility of item i_p in itemset S is defined as

$$u(i_p, S) = \sum_{T \in DB,\ S \subseteq T} u(i_p, T).$$

For example, utility of item E in itemset {B, E} is u(E, {B,E}) = u(E, T2) + u(E, T7) + u(E, T10) = 20.

E. The utility of itemset S in database DB: - The utility of itemset S in database DB is defined as

$$u(S) = \sum_{T \in DB, S \subseteq T} u(S,T) = \sum_{T \in DB, S \subseteq T} \sum_{i_p \in S} f(x_p, y_p).$$

For example, utility of itemset {A,E} in database from Table 1 is

u({A,E}) = u({A,E} , T3) + u({A,E} , T4) + u({A,E} , T8) = 33.

***F. The utility of transaction T*:-** The utility of transaction T is defined as

$$u(T) = \sum_{i_p \in T} u(i_p, T).$$

For example: utility of transaction T10 is

u(T10) = u({B} , T10)+u({C} , T10)+ u({E} , T10) = 72.

G. The utility of database DB: - The utility of database DB is defined as

$$u(DB) = \sum_{T \in DB} u(T).$$

For example, utility of database DB from table 1 and table 2 is

u(DB) = u(T1) + . . . +u(T10) = 23 + . . . + 72 = 400.

***H. The utility share of itemset S in database*:-** The utility share of itemset S in database DB is

For example, utility share of itemset {A,D,E} in database fromTable 1 is U({A,D,E}) = 46/400 = 0.115 = 11.5%.

III. Brief Review of Work Already Done

Agarwal (1994) studied the mining of association rules for finding the relationships between data items in large databases. Chan (2003) observes that the candidate set pruning strategy exploring the ant monotone property used in apriori algorithm does not hold for utility mining. Yao (2004)[15,16] defines the problem of utility mining formally. The work defines the terms transaction utility and external utility of an itemset. The mathematical model of utility mining was then defined based on the two properties of utility bound and support bound. Yao (2006, 2007) defines the utility mining problem as one of the cases of constraint mining. This work shows that the downward closure property used in the standard Apriori algorithm and the convertible constraint property are not directly applicable to the utility mining problem[4,5,6]. Li (2008) proposed two efficient one pass algorithms MHUI-BIT and MHUI-TID for mining high utility itemsets from data streams within a transaction sensitive sliding window. Liu et al in proposes a Two-phase algorithm for finding high utility itemsets[7,8]. Shankar (2009) presents a novel algorithm Fast Utility Mining (FUM) which finds all high utility itemsets within the given utility constraint threshold[9,10]. In 2010 Vincent S. Tseng, Cheng-Wei Wu, Bai-En Shie, and Philip S. Yu proposed a data structure,named UP-Tree, and then describe a new algorithm, calledUP-Growth, The framework of the UP-Growth. In 2012 Mengchi Liu and JunfengQu proposed "Mining High Utility Itemsets without Candidate Generation"[11,12]. In 2013 Arumugam P and Jose Proposed "Advance Mining Of High Utility Itemsets In Transactional Data". In 2014 More Rani N. and AnbhuleReshma V "Mining High Utility Item sets From Transaction Database"[13,14].

IV. Problem Statements

A. Unnecessary candidate generation: Proposed method use large search space to generate unnecessary candidate at second level and higher levels.

B. Accuracy: - The function used for calculatingUtility is also inefficient.

C. Arithmetic complexity: -proposed methods used complex calculation and formula. Thus, since calculation takes times, the overall computation time consuming.

V. PROPOSED ALGORITHM

Step1. Scan each Transaction ∈DB

Step2. Compute the utility value ∀single item set and transaction utility of each transaction.

Step3. Calculate transaction weight utility of each item.

Step4. Now compare the given threshold with transaction weight utility and item utility

Step5. If TU ≥ threshold then add into high utility item set else delete

Step6. Merge those transaction which contain same items

Step 7.Update transaction weight utility value and go to step 4

Step8.Reapeat until no more candidates' generation is possible

Step9.Return high utility item set (H);

Step10.Ends

VI. CONCLUSION AND BENEFITS

Utility item set mining is an enhancement of frequent pattern mining is considered to be a type of un-supervised learning technique, from the viewpoint of a developer there is a need to develop a strategy to mine large sized categorical data set efficiently. Mining Expected Utility Two Phase and several other algorithms have mine high utility item set very efficiently. But there is need to enhance this algorithm so that it can be applied to large sized dataset. The complexity factor for frequent pattern mining algorithm includes several factors like Execution time and I/O costArithmetic complexity and Number of Candidates generation, Memory used, Accuracy of the algorithm, Scalability in term of number of recordsWe test MEU,TP and TRS algorithm using different parameter like threshold value , candidate generation ,number of record ,memory used and accuracy of the algorithm .Form the experiment it clear that proposed algorithm perform well and compared to the MEU and TP.From Graph it is clear that proposed method outperforms others by reducing arithmetic complexity, number of candidate generation and memory used to execute algorithm. This investigation has presented a new approach, for utility item sets mining

References

1. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proc. of the 20th Int'l Conf. on Very Large Data Bases, pp. 487-499, 1994.
2. C. F. Ahmed, S. K. Tanbeer, B.-S. Jeong and Y.-K. Lee. Efficient Tree Structures for High-utility Pattern Mining in Incremental Databases. In IEEE Transactions on Knowledge and Data Engineering, Vol. 21, Issue 12, pp. 1708-1721, 2009.
3. R. Chan, Q. Yang and Y. Shen. Mining high-utility itemsets. In Proc. of Third IEEE Int'l Conf. on Data Mining, pp. 19-26, Nov., 2003.
4. Y. L. Cheung, A. W. Fu, Mining frequent itemsets without support threshold: with and without item constraints. IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 6, pp. 1052-1069, 2004.
5. K. Chuang, J. Huang, M. Chen, Mining Top-K Frequent Patterns in the Presence of the Memory Constraint, The VLDB Journal, Vol. 17, pp. 1321-1344, 2008.
6. A. Erwin, R. P. Gopalan and N. R. Achuthan. Efficient Mining of High-utility Itemsets from Large Datasets. In PAKDD 2008, LNAI 5012, pp. 554-561, 2008.
7. A. W. Fu, R. W. Kwong and J. Tang, Mining N-Most Interesting Itemsets, In Proc. of ISMIS'00, 2000.
8. J. Han, J. Pei and Y. Yin.Mining frequent patterns without candidate generation. In Proc. of the ACM-SIGMOD Int'l Conf. on Management of Data, pp. 1-12, 2000.
9. J. Han, J. Wang, Y. Lu and P. Tzvetkov, "Mining Top-k Frequent Closed Patterns without Minimum Support," In Proc. of ICDM, 2002.
10. Y. Hirate, E. Iwahashi and H. Yamana, TF2P-Growth: An Efficient Algorithm for Mining Frequent patterns without any Thresholds, In Proc. of ICDM 2004.
11. H.-F. Li, H.-Y.Huang, Y.-C. Chen, Y.-J. Liu, S.-Y. Lee. Fast and Memory Efficient Mining of High Utility Itemsets in Data Streams. In Proc. of the 8th IEEE Int'l Conf. on Data Mining, pp. 881-886, 2008.

12. Y. Liu, W. Liao, and A. Choudhary. A fast high-utility itemsets mining algorithm. In Proc. of the Utility-Based Data Mining Workshop, 2005.

13. Y.-C. Li, J.-S.Yeh and C.-C. Chang. Isolated Items Discarding Strategy for Discovering High-utility Itemsets. In Data & Knowledge Engineering, Vol. 64, Issue 1, pp. 198-217, 2008.

14. S. Ngan, T. Lam, R. C. Wong and A. W. Fu, Mining N-most Interesting Itemsets without Support Threshold by the COFI-Tree, Int. J. Business Intelligence & Data Mining, Vol. 1, No. 1, pp. 88-106, 2005.

15. J. Pisharath, Y. Liu, B. Ozisikyilmaz, R. Narayanan, W. K. Liao, A. Choudhary and G. Memik, NU-MineBench version 2.0 dataset and technical report, http://cucis.ece.northwestern.edu/projects/DMS/MineBench.html

16. T. M. Quang, S. Oyanagi, and K. Yamazaki, ExMiner: An Efficient Algorithm for Mining Top-K Frequent Patterns, ADMA 2006, LNAI 4093, pp. 436 – 447, 2006.

