Indian Journal of Science and Technology, *Vol 10(13), DOI: 10.17485/ijst/2017/v10i13/95258, April 2017*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

An Enhanced Algorithm for Retrieving High Utility-Frequent Item Sets with Negative Utility Values

C. Sivamathi * and S. Vijayarani

Department of Computer Science, Bharathiar University, Coimbatore - 641046, Tamil Nadu, India;
c.sivamathi@gmail.com, vijimohan_2000@yahoo.com

Abstract

Objectives: Utility mining gains more attraction in recent years. Utility mining can be defined as mining item's utility and revealing high utility items. In this work an algorithm EHUFIM (i.e Enhanced High Utility Frequent Itemset Mining), was proposed to reveal high utility – frequent itemsets even with negative profits. **Methods/Statistical Analysis**: The proposed algorithm uses utility mining methods to retrieve high profitable items. Then it uses support measure to reveal high occurrence items. The algorithm implements filter procedure of HUINIV algorithm to handle negative profit items. **Findings**: This algorithm discovers itemset that have more frequency and high utility with negative profit. Discovering such items helps in decision making in super markets, cross product marketing etc. The proposed algorithm was executed and performance of the algorithms was calculated. **Application/Improvement:** Existing utility frequents mining algorithms does not consider negative profit values. But, this proposed algorithm, takes negative utility values into account.

Keywords: High Utility Itemset, Negative Utility Itemset, Transaction Weighted Utility Property, Utility Frequent Itemset Mining, Utility Mining

1. Introduction

Today researchers started to contribute and develop new algorithms, techniques and concepts in the area of utility mining and this is very essential for making correct decisions[1,2,3,4]. The term utility in utility mining exemplify any measurement defined by the researcher. For example in web mining, utility signifies the time duration that was spent by the user in visiting a particular web page, in transaction database, utility represents the number of purchased items or profit of item etc. Thus utility of a thing varies based on its application. In utility mining, high utility itemsets are identified based on their utility values. The basic condition applied for selecting the high utility item or itemset is, to verify whether the item/itemsets utility value is higher than the minimum threshold. Closed utility itemset mining, negative utility itemset mining, utility-frequent itemset mining and on-shelf utility mining are considered as the significant utility mining forms[5,6]. In this research, utility frequent itemset mining is selected. Utility – frequent itemset mining reveals itemsets with high utility and high support i.e it considers both utility of item and frequent occurrence of an item or itemset.

There are numerous algorithms available in utility mining. Some of them are[7,8,9]: Two phase algorithm, UP growth algorithm, Fast utility mining algorithm, HUI Miner algorithm, Efficient High Utility-Itemset Mining (EHUIM) Algorithm etc. However these algorithms can handle positive utility values only. It cannot handle if itemset has negative values. But, in practical, some retail store may have items that may be sold in lost or some items may be sold as free offer for other items. Such items are considered as items with negative profits. Faster High-Utility itemset miner with Negative unit profits (FHN) and High Utility Itemsets with Negative Item Values (HUINIV) algorithms can calculate profitable items even if the database contains some of the negative values. Faster On-Shelf High Utility itemset miner (FOSHU) and Three-Scan Mining Algorithm for High On-Shelf Utility

**Author for correspondence*

Itemsets (TP-OHUI) algorithms retrieve significant itemsets which takes the duration of item placed on shelf of super markets into account[5,6,7,8].

The concept of utility mining comes from frequent itemset mining (FIM)[10,11]. The basic difference between FIM and Utility mining is, FIM consider only instances of items in a dataset while utility mining considers the significance of an item. In real life there may be some items which may be infrequent in occurrence, but may have great significance. For example, in retail store {pen, notebook} may be a frequent item. But it cannot provide a profit more than a 25kg of rice bag. Hence the importance (utility) of item should also be considered for better decision. FIM concentrates on number of frequency of items and utility mining concentrates on importance of itemset. In some cases there may be a need for discovering utility – frequent itemsets. BU-UFM (Bottom-up two-phase algorithm) and FUFM (Fast Utility-Frequent Mining) algorithm disclose the itemsets whose occurrences and utility value is high. Both algorithms can handle positive utility values only[7,8]. It does not hold negative utility values[9,10,11]. In this work, a new algorithm is proposed to discover high utility frequent items

2. High Utility- Frequent Itemset Mining

In retail stores or in super markets there may be some items which may not be sold for several days. To clear the stock, the management either decides to sell the item as free with other items. So that they may earn earnings from other items which are cross-promoted with these free items. For example, consider item X was not sold for more days. So the management announces an offer. If a customer bought two of item Y, then, they would get one item X as free. Let us assume that management earns three rupees as a profit for every purchase of single item Y and loses two rupees for every item X. From the offer, if a customer buys two items of Y, they offer item X. Here two rupees loss of item X can be compensated with 9 rupees profit of 3 items Y. This is known as cross product marketing. Hence there is need to discover high profitable and high frequent products, though some of the items may have negative utility. So far, no such algorithm is implemented with this strategy. This research work focused to propose an algorithm EHUFIM which has the ability to handle negative utility items.

Table 1. Terms in utility mining

S.No	Utility Terms	Equivalent term in Transaction database
1	Internal Utility (QU)	Quantity of item A (Q)
2	External Utility (PU)	Profit of item A (P)
3	Utility Function (UF)	Q * P
4	Utility of item A in a Transaction T_i (UT)	$\sum UF(A, T_i)$
5	Utility of itemset AB in Transaction T_i	$UF(A, T_i) + UF(B, T_i)$
6	Utility of itemset AB in transaction T_i	$UF(AB, T_i)$
7	Utility of itemset AB in database.	UF(AB) for all transactions
8	Utility of Transaction T_i(TU)	$\sum UF(i)$ for all $I \in T_i$
9	Transaction Weighted Utility TWUtility	$TU(T_i) \forall T$

3. Proposed Algorithm -EHUFIM

The proposed algorithm EHUFIM i.e Enhanced High Utility Frequent Itemset Mining, finds frequency of itemset along with high utility values. The pseudocode is shown in Figure 1. Moreover this algorithm can handle negative utility values. The proposed algorithm uses filter procedure of HUINIV algorithm[5], for generating less number of high transaction weighted utility itemsets. Consider the following sample database:

TS1 {A(1), B(2),D(2), E(1)}
TS2 {A(2), B(1), C(2), D(6)}
TS3 {A(1), C(1), E(6)}
TS4 {A(1), C(1), E(1)}
TS2 {B(1), C(2), D(6),E(1)}

Here TS1, TS2, TS3, TS4 and TS5 are transactions. A, B, C, D and E are items. The number given within the parenthesis is the quantity of that item sold in a respective transaction. i.e, TS1 {A(1), B(2),D(2), E(1)} means that TS1 is transaction id, which enclose the products{A,B,D,E}. The number of item A sold in TS1 is1, B is 2, D is 2 and E is 1.

Consider the Profit of A , B, C ,D , E are -4, -3,2,5 and 5 respectively. Here products A and E are assumed to have negative value. High utility itemset are calculated using

preliminary definitions like: Utility of itemset, Transaction utility, Transaction weighted utility (TWUtility)[12,13,14,15,16]. It is given in Table 1.

The proposed algorithm first calculates transaction weighted utility value for all items. Then high transaction-weighted utilization itemsets i.e whose utility > TWUtility are stored in HTWUI set. Then the itemset with all items as negative items are removed from it. Finally high utility itemsets[5] are discovered.

Transaction Utility (TU) for all items are calculated without considering negative values[5]. TWUtility and Support arecalculated for each item. For example, TWUtility(A) = TU(TS1) + TU(TS3) + TU(TS4). Similarly TWUtility of B, C, D and E are calculated. All these values are given in Table 2.

Table 2. TU, TWU and Support of Items

Item	TU	TWUtility	Support
A	15	63	80
B	34	118	60
C	7	87	80
D	7	118	60
E	39	108	80

```
Input : A database DB.
for i=1 to n
        calculateTWU(i); // Calculate Transaction weighted utilization of each item.
        calculateSupport(i);
        if(TWU(i)  >= Min_Util)
                HTWUItems  = HTWUItems U I; //Add this item to High TWU items.
end for;
for = i=2 ; HTWUItems ? NULL ; i++
i.TWU = i.GenerateHTWU(i); //Generated high transaction weighted utilization I candidate itemsets.
i.TWU = i.GenerateSupport(i);//Generate Support for each itemsets.
If(i.TWU ? NULL)
        For j=1 to n
                If(i.TWU >= Min_Util && i.support >=Min_Support)
                        HTWUItems  = HTWUItems U i; //Add this item to High TWU items.
        end for;
end if;
for each itemset i ? HTWUItems  && i.value > 0 // itemset with negative utility
for k= 1 to n
        if(i.twu >=  Min_Util && i.support >= Min_Support)
        HUItems = HUItems U I; // Add item to high utility items.
        end if;
end for;
return HUItems;
```

Figure 1. Pseudo Code for the Proposed Algorithm.

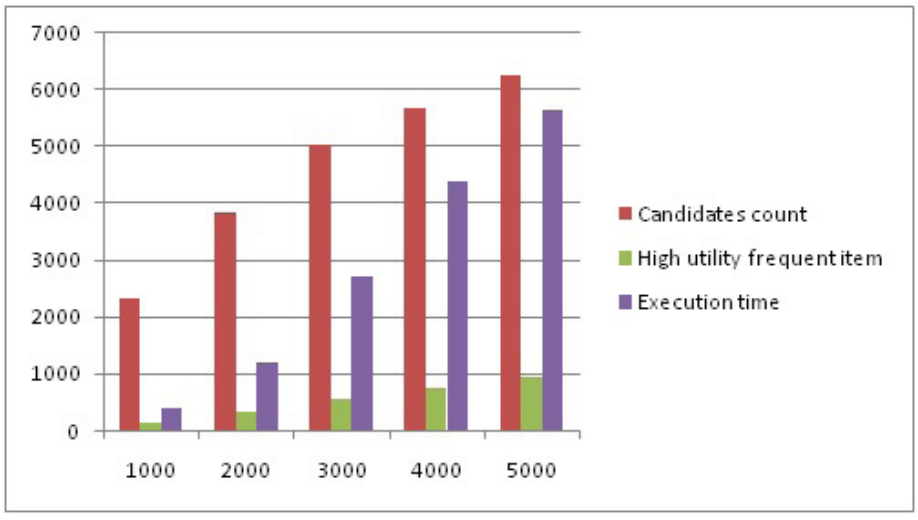

Figure 2. Comparison of No.of Candidates and High Utility Frequent Itemset at Various No. of Transactions.

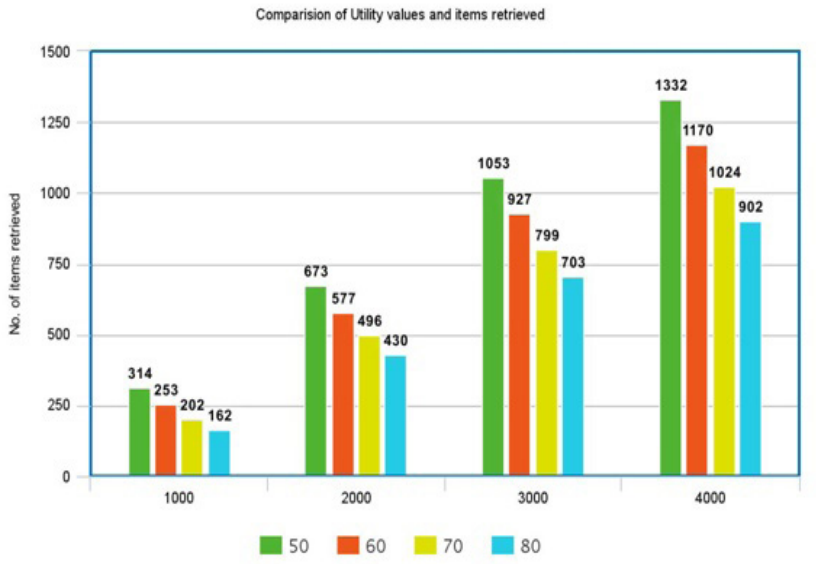

Figure 3. Comparison of No. of High Utility Frequent Itemset at Various No. of Transactions with Varying Utility Value.

If min_utility=100 and min_support =60 then,
candidate 1-itemsets -> {(B),(D),(E)}.
candidate 2-itemsets ->{(B,C), (B,E)}
candidate 3-itemsets -> {(B,C,D)}

After candidate generation, High utility Frequent itemsets are revealed.

Table 3. Retail dataset

RETAIL DATASET			
No. of Transactions	**Candidates count**	**No. of Items retrieved**	**Execution time in milli seconds**
1000	2323	124	375
2000	3844	322	1203
3000	5047	560	2703
4000	5689	738	4375
5000	6244	928	5648

4. Experimental Evaluation

The proposed algorithm was implemented using RETAIL DATASET. It is a benchmark dataset. It contains retail market basket data from an anonymous Belgian retail store. It contains 541909 instances with 8 attributes. Algorithm was implemented in Java and executed in a machine with 3.20GHz CPU. In this experiment the algorithm was executed with 1000, 2000, 3000, 4000 and 5000 transactions. Candidates count, execution time and number of Itemsets retrieved for each dataset transaction are compared. Also the numbers of items with various support values are compared. The results are shown in Table 3. Figure 2 shows the comparison of number of candidates and high utility frequent itemset at various number of transactions. Figure 3 displays the comparison of number of high utility frequent itemset at various number of transactions with varying utility value.

Table 4. Number of items retrieved with various transactions and minimum support and minimum utility value

RETAIL DATASET			
No. of Transactions	**Minimum Support value**	**Minimum Utility value**	**No. of Items retrieved**
1000	15	50	314
		60	253
		70	202
		80	162
2000	20	50	673
		60	577
		70	496
		80	430

3000	30	50	1053
		60	927
		70	799
		80	703
4000	40	50	1332
		60	1170
		70	1024
		80	902

5. Conclusions and Future Enhancement

Profitable items are retrieved in Utility Mining, whereas high occurrence items are discovered in frequent mining. Utility - frequent Itemsets is one of the forms of utility mining, which discovers both highly frequent and highly utility Itemsets. In this work an algorithm EHUFIM is proposed to retrieve utility frequent Itemsets with negative profits. The results at various numbers of transactions, with various support values are discussed. Retrieving such Itemsets with negative profit helps in better decision making for cross product. This algorithm discovers frequently occurring itemsets with negative profit. This helps to improve the sales of negative profit items. In future, the algorithm can be proposed to discover high utility frequent itemset in data streams with negative values.

6. Reference

1. Indumathi M, Vaithiyanathan V. Reduced overestimated utility and pruning candidates using incremental mining. Indian Journal of Science and Technology. 2016 Dec; 9(48):1–6. Crossref
2. Kannan KS. A Partial weighted utility measure for fuzzy association rule mining. Indian Journal of Science and Technology. 2016 Mar; 9(10): 1–6.
3. Grace LKJ, Maheswari V. Efficiency calculation of mined web navigational patterns. Indian Journal of Science and Technology. 2016 Sep; 7(9):1350–54.
4. Liu Y, Liao W, Choudhary AA. Fast High utility itemsets mining algorithm. Proceedings of the Utility-Based Data Mining Workshop; USA. 2005; 90–99. Crossref
5. Tseng VS, Chu CJ, Liang T. Efficient mining of temporal high utility itemsets from data streams. Proceedings of ACM KDD Workshop Utility-Based Data Mining (UBDM'06); Philadelphia, Pennsylvania. USA: 2006.p. 1–81.
6. Fournier-Viger P. FHN : Efficient mining of high utility itemsets with negative unit profits. 2014.p. 16–29.
7. Chu CJ, Vincent S, Tseng T, Liang T. An efficient algorithm for mininghigh utility itemsets with negative item values in large databases. Applied Mathematics and Computation. 2009; 215(2): 767–78. Crossref
8. Fournier-Viger P, Zida S. FOSHU: Faster on-shelf high utility itemset mining– with or without negative unit profit. In proceedings SAC; Salamanca, Spain. 2015.p. 1–8.
9. Lan GC, Hong TP, Vincent S, Tseng T . A three-scan mining algorithm for high on-shelf utility itemsets. Intelligent Information and Database Systems. 2009; 5991:351–58.
10. Lan GC, Hong TP, Vincent S, Tseng. Mining On-shelf High Utility Itemsets Taiwan. 2007.p. 1–8.
11. Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases. Proceedings of ACM SIGMOD Intl. Conf. on Management of Data; Washington, DC. 1993.p. 207–16. Crossref
12. LiuY, Liao W, Choudhary A. A two-phase algorithm for fast discovery of high utility itemsets. Proceeding. PAKDD Springer-Verlag Berlin Heidelberg; 2005. p. 689–95 . Crossref
13. Yao H, Hamilton HJ, Butz C J. A Foundational approach to mining itemset utilities from databases. Proceedings of the 2004 SIAM International Conference on Data Mining; 2004. p. 428–86 . Crossref
14. Tseng VS, Chu CJ, Liang T. An Efficient method for mining temporal emerging itemsets from data streams. International Computer Symposium (ICS), Workshop on Software Engineering, Databases and Knowledge Discovery, Taipei, Taiwan: 2006. 215(2). p. 767–78.
15. Shankar S, Purusothoman TP, Jayanthi S, Babu N. A Fast algorithm for mining high utility itemsets. Proceedings of IEEE International Advance Computing Conference; Patiala, India. 2009. p. 1459–64 . Crossref
16. Kanimuthu S. iFUM - Improved Fast Utility Mining. International Journal of Computer Applications. 2011; 27(11):32–36. Crossref