Volume 4, Issue 5, May 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Survey on Utility Mining Methods 2PUF, IHUP, FUFM

P.Dhana Lakshmi [1], K. Ramani [2]

Assistant Professor, Department of Computer Science And Systems Engineering, SVEC, A.Rangampet[1]

Professor, Department Of Information Technology, SVEC, A. Rangampet [2]

Abstract:Data mining is a process of retrieving useful information from large databases. It is not easy to get the required information from large data manually. Several algorithms were proposed to extract the information in which the user is interested in, from voluminous data using various algorithms including Association rule mining, Classification, Clustering ,Prediction techniques etc. The association rule mining derives some rules which describe the relationship between item sets. The prediction techniques helps to predict the future based on these association rules. This leads to better decision making about the future. The organizations are interested in finding the items which gives more profit and also customers who contribute more profit to them. The item sets which give more profit are high utility item sets. In this paper, the study includes 3 classical utility mining methods that are 2PUF , IHUP and FUFM and discusses some issues related with these algorithms.

Keywords: Frequent Pattern Mining, 2PUF IHUP, Quasi support, Extended support

1. INTRODUCTION

Now -a-days, Data mining is widely used in marketing to analyze the data and to predict the future. Data mining is defined as the process of retrieving useful information from large databases [1]. The retail organizations stores large amounts of data regarding their sales and customers in databases. Later they view those databases and extract the information which they are interested in. Some organizations may interest in the items frequently purchased by the customers, some may interested in the items which gives more profit to the organization. The organizations extract the information required to them based on their interest from the databases. This information helps the organization later, to take decisions to improve their market.

When we have fewer amounts of data it is easy to get particular information what we are interested. But, when there is a voluminous amount of data, it is very difficult to get the particular information manually. So, different data mining techniques and algorithms were developed to extract the data from such voluminous databases. These techniques include Association rule mining, Prediction techniques etc. The association rule [2,4] mining derives some rules which describe the relationship between item sets. The prediction techniques helps to predict the future based on these association rules. This leads to better decision making about the future.

The researchers were concentrated on the items which are purchasing more in number by the customers. Those items are called as frequent items. Different algorithms were proposed to find the frequent item sets like Apriori [1,2,3], Apriori TId etc. Later their interest moved to the high utility item sets. The high utility items are those which contribute more profit to the organization rather than their frequency. An item set is said to be a high utility item [3], if the utility of that item is greater than or equal to user specified utility. The utility is obtained by the product of internal utility and external utility. The internal utility is the quantity of each item and the external utility is the profit obtained on that item.

After finding the high utility item sets, the customers will be classified into two groups based on their profit contribution to the organization. The customers whose profit contribution is more those are called as premium customer, otherwise ordinary customers. The number of customers in future will be predicted using Linear Regression prediction technique based on the past data.

II. TECHNIQUES FOR UTILITY PATTERN MINING:

There are various techniques are proposed for generating utility frequent item sets so that high profitable items are mined efficiently. The approaches of generating utility item sets are divided into 3 basic techniques:

- 2PUF Algorithm
- IHUP Algorithm
- FUFM Algorithm

2PUF Algorithm:

The utility based data mining is a research area interested in all types of utility factors .Utility can be defined in terms of profit or usage of items based on user interest. The utility based mining in data mining contributes a lot to the decision making in marketing. The user can increase or decrease the sales based on the utility of the item sets. V.S Tseng

focuses on the item sets which are frequent and gives high utility. The utility item sets are those which give more profit on the items. The frequent items are those which are frequently purchased by the customers without considering their profits. If these two techniques are combined, that is the items which gives more profit and which are purchased frequently by the customer are called as utility frequent item sets.

To find these utility frequent item sets, 2P-UP [8], 2Phase Utility Pattern algorithm Ps.yu introduced but it not a much efficient one. So another algorithm by name FUFM , Fast Utility-Frequent Mining algorithm is also discussed to find utility-frequent item sets.

The utility can be measured by the product of **internal utility and external utility. Internal utility**[3,8] refers to the quantity of items and **external utility[5]** refers to the profit of each item. The item sets are high utility item sets if their utility is greater than or equal to a user specified extended support value. The item sets are frequent if the percentage of the support count in a database is greater than or equal to user specified support threshold.

Example:

Table1:Transaction Table

TID	A	B	C	D	F
T1	1	0	10	1	0
T2	2	0	6	0	2
T3	2	2	0	6	2
T4	0	4	13	3	1
T5	0	2	4	0	1
T6	1	1	1	1	0

Utility of Database and Transactions:

U[DB]= U[T1]+U[T2]+U[T3]+U[T4]+U[T5]+U[T6]
U[T1]= 1*5+0+10+1*2=5+10+2=17
U[T2]= 10+0+6+0+6+5+0=27
U[T3]= 10+4+0+12+6+5+0=47
U[T4]= 8+13+6+3=30
U[T5]= 4+4+9+2=19
U[T6]= 5+2+1+2+2=12
U[DB]= 17+27+47+30+11+12=144

Support calculation:

Total items=7

Take 4 itemsets
{ABCD}
{ABCE}
{BCDE}
{ABDE}
{ACDE}

Support threshold=0.63

Quasi:

{ABCD}
T1= 1*5+1*10+1*2= 5+10+2 = 17
T2= 2*5+6*1=10+6= 16
T3= 2*5+2*2+6*2=10+4+12= 26
T4= 4*2+13*1+3*2=8+13+6= 27
T5= 2*2+4*1=4+4= 8
T6=1*5+1*2+1*1+1*2= 5+2+1+2= 10

Quasi support {ABCD}=17+16+26+27+10+8/144

= 0.72

FOR ITEMSET {ABCE}
T1= 1*5+10*2= 25
T2= 2*5+6*1+2*3=22
T3= 2*5+2*2+2*3=20
T4= 4*2+13*1+1*3=24
T5= 2*2+4*1+1*3=11
{ABCE}= 22+20+24+11+8/144
=0.590

FOR ITEMSET {BCDE}

T1= 10*1+2*1= 12
T2= 6*1+2*3= 12
T3= 2*2+6*2+2*3= 22
T4= 4*2+13*1+2*3+1*3= 30
T5= 2*2+4*1+1*3=11

{BCDE}=12+12+22+30+11+5/144
=0.63

FOR ITEMSET {ABDE}

T1= 1*5+1*2= 7
T2= 2*5+2*3= 16
T3= 2*5+2*2+6*2+2*3= 32
T4= 4*2+3*2+1*3= 17
T5=2*2+1*3= 7

{ABDE}=(7+16+32+17+7+9)/144
=0.611

FOR ITEMSET {ACDE}

T1= 1*5+10*1+1*2=17
T2= 2*5+6*1+2*3=22
T3= 2*5+6*2+2*3= 28
T4= 13*1+3*2+1*3= 21
T5= 4*1+1*3= 7
{ACDE}=(17+22+28+21)/144
=0.680

PERFORM INTERSECTION OPERATION ON SETS {ABCD}{BCDE}{ACDE}
{ABCD} ∩ {ACDE}={ACD}
{ABCD} ∩ {BCDE}={BCD}
{BCDE } ∩ {ACDE}={CDE}

FOR ITEMSET {ACD}

T1=1*5+10*2+1*2=27
T2=2*5+6*1=16
T3=2*5+6*2=22
T4=13*1+3*2=1
T5=4*1=4
T6=1*5+1*1+1*2=8
{ACD}=(27+16+22+1+4+8)/144
=0.66

FOR ITEMSET {BCD}

T1=10*1+1*2=12
T2=6*1=6
T3=2*2+*2=16
T4=4*2+13*1+3*2=27
T5=2*2+1*1+1*2=5
T6=2*1+1*1=1*2=5

{BCD}=(12+6+16+27+8+5)/144
=0.506

FOR ITEMSET {CDE}
T1=10*1+1*2=12
T2=6*1+2*3=12
T3=6*2+2*3=18
T4=13*1+3*2+1*3=21
T5=4*1+1*3=7
T6=1*1+1*2=3

{CDE}=(12+12+18+21+7+3)/144
=0.506

PERFORM INTERSECTION OPERATIONS:
{ACD} ∩ {BCD}={CD}
{BCD} ∩ {CDE}={CD}
{ACD} ∩{CDE}={CD}

FOR IITEMSET {CD}

T1=10*1+1*2=12
T2=6*1=6
T3=6*2=12
T4=13*1+3*2=19
T5=4*1=4
T6=1*1+1*2=3
{CD}=(12+6+12+19+4+3)/144
=0.361

Therefore the high utility itemsets are:
{ABCD}
{BCDE}
{ACDE}
{ACD}.

FU-UP algorithm:

The FP-UP algorithm uses a special measure called quasi support where as our focused algorithm FUFM uses extended support.The quasi support is defined as,quasiSupport(I,u)= $\left|\frac{T'_{I,u}}{D}\right|$ where T' is a set of transactions and $T'_{I,u} = \{T \mid u(I,T) \geq u \wedge T \in D\}$.It is not necessary for an item set I to be a true subset of transaction T when computing its quasisupport [3]. In the first phase of 2P-UP algorithm, all quasi frequent item sets are collected and in the second phase all utility infrequent items are discarded. The algorithm starts with item sets of length n-1, where n is the number of items in database. It computes intersection of each item set with all other item sets. Candidates of length n-2 which do not have quasi utility-infrequent supersets and satisfy the given utility threshold are appended to the set of quasi utility-frequent item sets. These item sets are used in next iteration to find all quasi utility-frequent item sets of length n=3.The process repeats till candidates of length 1 are generated.

The **disadvantages** of this algorithm:

- The reversed way of candidate generation.

- It wastes time checking long item sets that are unusual to be utility-frequent.
- Candidate generation is also very slow and inefficient as it computes intersection of every pair of candidates in each iteration.
- Moreover the computation of quasi support is also inefficient.

FUFM: FAST UTILITY FREQUENT MINING ALGORITHM:

Fast Utility Frequent Mining Algorithm overcomes the disadvantages of the 2P-UP algorithm.

The algorithm starts with one item set generation and it finds the set of candidates with support greater than or equal to user specified minsup. Then compute extended support for all candidates. The extended support measure can be calculated as, support(I,u)= $\left|\frac{T_{I,u}}{D}\right|$,where $T_{I,u} = \{T \mid I \subset T' \wedge u(I,T) \geq \mu \wedge T \in D\}$.

The item sets which satisfy user specified support utility value are the utility frequent item sets[1]. Now generate next candidate item sets using frequent item set mining algorithm, Apriori[1,4,6] by using the old set of frequent candidates. It the item set is not null then calculate extended support and find the utility frequent item sets. The process repeats till the item set becomes null.

The **Disadvantages** of this algorithm:

- It generates more candidate itemsets.
- It consumes more time.

Example:

Consider the following database:

minsupp=2,utility threshold =1,support threshold=0.1

TID	A	B	C	D	E
1	0	0	18	0	1
2	0	6	0	1	1
3	2	0	1	0	1
4	1	0	0	1	1
5	0	0	4	0	2
6	1	1	0	0	0
7	0	10	0	1	1
8	3	0	25	3	1
9	1	1	0	0	0
10	0	6	2	0	2

Table2: Transaction Table

PROFIT TABLE:

ITEM	A	B	C	D	E
PROFIT ON EACH ITEM	3	10	1	6	5

ALGORITHM:

GENERATE 1-ITEM SET:

ITEM	COUNT
A	8
B	24
C	50
D	6
E	10

GENERATE ITEM SETS THAT SATISFIES MINSUP=2

ITEM	COUNT
{A}	8
{B}	24
{C}	50
{D}	6
{E}	10

CALCULATE EXTEND SUPPORTS OF EACH ITEM SET:

ITEM SET	EXTENDED SUPPORT
{A}	0.057
{B}	0.57
{C}	0.12
{D}	0.08
{E}	0.12

IHUP:

IHUP-Tree algorithm:

IHUP-tree is to maintain the information about utilities and their itemsets. Every node in IHUP-tree consists of an item name, TWU value and a support count.

IHUP algorithm has three steps:

1) IHUP tree construction.
2) For the generation of HTWUIs.
3) High utility itemsets are identified in last step.

In first step we have to rearrange in a fixed order such as lexicographic order, support descending order or TWU [7] descending order in each item in transaction. After the rearranged transaction are inserted into an IHUP-tree.

Table 1 shows the database for the global IHUP-tree in fig 1 in which items are in the descending order of TWU. For every node in fig 1, the first numbers besides item name is its TWU and the second number is its support count.

By applying FP-growth[6] the HUP-tree, the HTWUIs are generated. The set of HTWUIs are identified by the high utility itemsets and their utilities by just scanning the original database only once.

Disadvantages:

- It produces too many HTWUIs in phase 1.
- A large number of HTWUIs will degrade the mining performance in terms of execution time and memory consumption.

Example:

IHUP-Tree algorithm:

TID	Transaction	TU
T_1	(A,1),(C,1),(D,1)	8
T_2	(A,2),(C,6),(E,2),(G,5)	27
T_3	(A,1),(B,2),(C,1),(D,6),(E,1),(F,5)	30
T_4	(B,4),(C,3),(D,3),(E,1)	20
T_5	(B,2),(C,2),(E,1),(G,2)	11

Table3: **Example database**

Item	A	B	C	D	E	F	G
Profit	5	2	1	2	3	1	1

Table4: **Profit table**

min_util=40

$Tu(T_1)$=1x5+1x1+1x2=5+1+2=8

$Tu(T_2)$=2x5+6x2+2x2+1x5=10+12+4+5=27

$Tu(T_3)$=1x5+2x2+1x1+6x2+1x3+5x1=5+4+1+12+3+5=30

$Tu(T_4)$=4x2+3x1+3x2+1x3=8+3+6+3=20

$Tu(T_5)$=2X2+1X2+3X1+1X2=4+2+3+2=11

Calculate transaction weighted utilities:

$Twu(A) = Tu(T_1)+Tu(T_2)+Tu(T_3)$

$=u(\{ACD\},T_1)+u(\{ACFG\},T_2)+u(\{ABCDEF\},T_3)$

$=(5x1+1x1+2x1)+(2x5+6x1+3x2+1x5)+(1x5+2x2+1x1+6x2+3x1+5x1)$

=(5+1+2)+(10+6+6+5)+(5+4+1+12+3+5) =8+27+30=65

$Twu(B)=Tu(T_3)+Tu(T_4)+Tu(T_5)$

$=u(\{ABCDEF\},T_3\}+u(\{BCDE\},T_4)+u(\{BCEG\},T_5)$

= (1x5+2x2+1x1+6x2+3x1+5x1)+(4x2+3x1+3x2+3x1)+(2x2+2x1+1x3+2x1)

=30+20+11=61

$Twu(C)=Tu(T_1)+Tu(T_2)+Tu(T_3)+Tu(T_4)+Tu(T_5)$

$=u(\{ACD\},T_1)+u(\{ACEG\},T_2)+u(\{ABCDEF\},T_3)+u(\{BCDE\},T_4)+u(\{BCEG\},T_5$

)

=(1x5+1x1+2x1)+(2x5+6x1+2x3+1x5)+(1x5+2x2+1x1+6x2+3x1+5x1)+(4x2+3x1+3x2+1x3)+(2x2+2x1+1x3+2x1)
=8+27+30+20+11
=96
Twu(D)=Tu(T_1)+Tu(T_3)+Tu(T_4)
=u({ACD},T_1)+u({ABCDEF},T_3)+u({BCDE},T_4)
=(1x5+1x1+1x2)+(1x5+2x2+1x1+6x2+1x3+5x1)+(4x2+3x1+3x2+6x3)
=8+30+20
=58
Twu(E)=Tu(T_2)+Tu(T_3)+Tu(T_4)+Tu(T_5)
=u({ACEG},T_2)+u({ABCDEF},T_3)+u({BCDE},T_4)+u({BCEG},T_5)
=(2x5+6x1+2x3+5x1)+(1x5+2x2+1x1+6x2+1x3+5x1)+(4x2+3x1+3x2+1x3)+(2x2+2x1+1x3+2x1)
=27+30+20+11
=88
Twu(F)=Tu(T_3)
=u({ABCDEF},T_3)
=(1x5+2x2+1x1+6x2+1x3+5x1)
=30
Twu(G)=Tu(T_2)+Tu(T_5)
=u({ACEG},T_2)+u({BCEG},T_5)
=(2x5+6x1+2x3+5x1)+(2x2+2x1+1x3+2x1)
=27+11
=38

Table5:Transaction weight Utility Table

Item	A	B	C	D	E	F	G
Twu	65	61	96	58	88	30	38

Construction of IHUP-Tree:

1. compute TWU, remove unpromising items, and rearrange the items.

Mininimum_utility=40
The items whose utilty is less than the minimum utility are unpromising items.
The unpromising items are F and G.

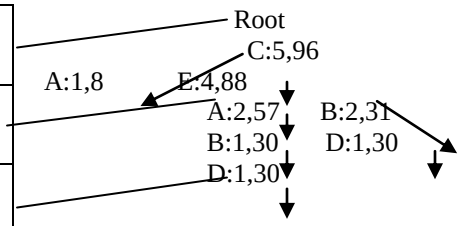
Tid	Transactions	Tu
T1	(C,1) (A,1) (D,1)	8
T2	(C,6) (E,2) (A,2)	27
T3	(C,1) (E,1)(A,1) (B,2) (D,6)	30
T4	(C,3) (E,1) (B,4) (D,3)	20
T5	(C,2) (E,1) (B,2)	11

TWU	96	88	65	61	58

2. Construct IHUP-Tree

Item	Twu

C	96
E	88
A	65
B	61
D	58



III. Conclusions and Future work:

The most popular data mining algorithm is association rule mining. Many number of efficient techniques available for association rule mining, which considers mining of frequent itemsets. But an promising technique in Data Mining is Utility Mining, which incorporates utility considerations during itemset mining. Utility Mining covers all aspects of economic utility in data mining and helps in detection of rare itemset having high utility. In this paper, a brief overview of various algorithms for mining of utility itemsets were presented. In the futurework, new algorithm will be proposed predicting and classifying the customers based by maintaining customer ids for improving the business .

References:

[1] Aakansha Sexena,Sohil Gandhiya," A Survey on Frequent Pattern Mining Methods Apriori,Eclat, Fpgrowth",2014 IJDER|Volume 2, issue|ISSN 232-9939

[2] Varsha Mashoria,Anju Singh,"Literature Survey On Various Frequent Pattern Mining Algorithm",vol 3,issue1(jan 2013),pp58-64.

[3] V.S. Tseng, C.-W. Wu, B.-E. Shie, and P.S. Yu, "UP-Growth: An Efficient Algorithm for High Utility Itemsets Mining," Proc. 16th ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD'10), pp. 253-262, 2010.

[4] Honglei Zhu, Zhigang Xu, An Effective Algorithm for Mining Positive and Negative Association Rules.2008 International Conference On computer Science and Software Engineering, pp. 978-0-7695-3336-0, 2008 IEEE.

[5] Y. Liu, W. Liao, and A. Choudhary, "A Fast High Utility Itemsets Mining Algorithm," Proc. Utility-Based Data Mining Workshop,2005.

[6] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," Proc. ACM-SIGMOD Int'l Conf. Management of Data, pp. 1-12, 2000

[7] C.H. Cai, A.W.C. Fu, C.H. Cheng, and W.W. Kwong, "Mining Association Rules with Weighted Items," Proc. Int'l Database Eng. and Applications Symp. (IDEAS'98), pp. 68-77, 1998.

[8] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," Proc. 20th Int'l Conf. Very Large Data Bases (VLDB), pp. 487-499, 1994.

