Scientific Journal Impact Factor (SJIF): 1.711
International Journal of Modern Trends in Engineering and Research
IJMTER
ISSN (ONLINE) : 2349 - 9745
ISSN (PRINT) : 2393 - 8161

Efficient Algorithms for Mining Compact Representation High Utility Itemsets

Muhasina C M[1], Shahina K K,[2]
[1,2]Department of Computer Science, IIET, Nellikuzhi

Abstract—High utility itemsets mining from databases is an important data mining task, which refers to the discovery of itemset with high utilities (e.g. high profits). Many studies have been proposed for mining HUIs, including Two-Phase , IHUP , TWU-Mining , IIDS and HUP-Tree UP-Growth . The former three algorithms use TWDC property to find HUIs. Mainly these algorithms consist of two phases. In first Phase , they find all HTWUIs from the database. In second Phase, HUIs are identified from the set of HTWUIs by calculating the exact utilities of HTWUIs. Although these methods capture the complete set of HUIs but they may generate too many candidates in Phase I, i.e., HTWUIs, which degrades the performance of Phase II and the overall performance. To reduce the number of candidates in Phase I, various methods have been proposed Recently, Tseng et al. proposed UPGrowth Though the above methods perform well in some cases, their performance degrades quickly when there are many HUIs in the databases. A large number of HUIs and candidates cause these methods to suffer from long execution time and huge memory consumption. When the system are limited (e.g. the memory space and processing power), it is often impractical to generate the entire set of HUIs. Besides, a large amount of HUIs is hard to be comprehended or analyzed by users. In this propose a novel framework in this paper for mining closed high utility itemsets (CHUIs), which serves as a compact and lossless representation..

Keywords—DataMining, frequent itemset, high utility itemset, closed itemset

I. INTRODUCTION

Mining frequent itemset (FIM) is a fundamental research topic in data mining. Market basket analysis is the main application of FIM .Relative importance of each item is not considered in frequent pattern mining. However, in this application ,may discover a large amount of frequent but low revenue itemsets and lose the information on valuable itemsets.The existing model of FIM may discover a large amount of frequent but low revenue itemsets and lose the information on valuable itemsets which provide huge profit. These problems are caused by the facts that FIM treats all items as having the same importance/ profit/weight and also does not indicate its purchase quantity in the transaction. Hence, FIM cannot discover itemsets with high utilities such as high profits. To address these issues of FIM, utility mining, emerges as an important topic in data mining.. In utility mining, each item is associated with a weight and purchase quantity. The utility of an itemset represents the importance of an itemset, which can be measured in many terms like weight, profit, cost, quantity or other information depending on the user. An itemset can be called a high utility itemset (HUI) if its utility is not less than a user-specified threshold; otherwise, it is called as low utility itemset. Utility mining is an important task and has a wide range of applications such as biomedical applications, cross marketing in retail stores, mobile commerce environment and website click stream analysis However,high utility itemset mining is not an easy task since the downward closure property does not hold in utility mining and the search space for mining high utility itemsets could not be directly reduced because a superset of a low utility itemset can be a high utility itemset. Many studies were proposed for HUIs, but they present a large number of high utility itemsets. A very large number makes it difficult for the users to comprehend the results. It may also make the algorithms inefficient in terms of time and memory requirement. It is recognized that the more high utility itemsets the algorithms generate, the more processing the algorithm consume. The performance decreases greatly for low minimum utility thresholds or when dealing with dense databases.

In frequent itemset mining , to reduce the computational cost of the mining task many studies focused on developing concise representations, such as free sets, closed itemsets [11], maximal itemsets These representations successfully reduce the number of itemsets found, but they are developed for FIM instead of HUI mining. Here proposed an algorithm CHUD (Closed High Utility itemset Discovery) to find closed sets. The CHUD algorithm includes three novel strategies . Result show that CHUD is much faster than the state-of-the-art algorithms for mining all HUIs.

The remainder of this paper is organized as follows. In Section II, we introduce the background for compact representations and utility mining. Section III defines the representation of closed+ HUIs and presents our methods. FHM are shown in Section IV and conclusions are given in Section V.

TABLE I. AN EXAMPLE TRANSACTIONAL DATABASE

TID	Transaction	TU
T_1	A(1), B(1), E(1), W(1)	5
T_2	A(1), B(1), E(3)	8
T_3	A(1), B(1), F(2)	8
T_4	E(2), G(1)	5
T_5	A(1), B(1), F(3)	11

TABLE II. UNIT PROFITS FOR EVERY ITEM

Item	A	B	E	F	G	W
Unit Profit ($)	1	1	2	3	1	1

II. RELATED WORKS

Many studies are proposed for mining high utility itemset mining including Two-Phase , IHUP, TWU-Mining , IIDS and HUP-Tree, PB , UP-Growth . These algorithms consist of two phases. In Phase I, they find all HTWUIs from the database. In Phase II, HUIs are identified from the set of HTWUIs by calculating the exact utilities of HTWUIs. Although these algorithms find the complete set of HUIs, but they may generate too many candidates in Phase I.To reduce the number of candidates various methods have been proposed . Recently, UPGrowth with four strategies DGU, DGN, DLU and DLN, for mining HUIs. Though above methods perform well in some cases,their performance degrades quickly when there are many high utility itemsets in the databases. A large number of HUIs and candidates itemsets cause these algorithms proposed suffer from long execution time and huge memory consumption .Besides, a large amount of HUIs is hard to analyzed by users. To provide not only compact but also complete information about high utility itemsets to users closed itemset mining was proposed.

High Utility ltemset Mining

Let I = $\{a_i\ a_2,\ldots, a_M\}$ be a finite set of distinct items. A transactional database D = $\{T_1,T_2,\ldots T_N\}$ is a set of transactions, where each transaction $T_R \in D(1 \leq R \leq N)$ is a subset of I and has an unique identifier R, called Tid. Each item $a_i \in I$ is associated with a positive real number $p(a_i, D)$, called its external utility. Every item a_i in the transaction TR has a real number $q(a_i, a_2,\ldots,a_k\}$ called its internal utility. An item- set $X = \{a_1, a_2\ldots a_K\}$ is a set of K distinct items, where $a_i \in I$, $I\leq i\leq K$, and K is called the length of X. A K-itemset is an itemset is an of length K. An itemset X is said to be contained in a transaction T_R if $X \subseteq T_R$.

Definition 1 (Support of an itemset). The support count of an itemset X is defined as the number of transactions Containing X in D and denoted as SC(X). The support of X is defined as the ratio of SC(X) to |D| . The complete set of all the itemsets in D is defined as L = {X |X $\subseteq$ I. SC(X) > 0}.

Definition 2 (Absolute utility of an item in a transaction). The absolute utility of an item a_i in a transaction T_R is denoted as au(a_i,T_R) and defined as p(a_i, D) x q(a_i,T_R).

Definition 3 (Absolute utility of an itemset in a transaction). The absolute utility of an itemset X in a transaction T_R is defined as au(X,T_R) $\sum_{ai\in X} au$(a_i,T_R).

Definition 4 (Transaction utility and total utility). The transaction utility (TU) of a transaction T_R is defined as TU(T_R) = au(T_R,T_R). The total utility of a database D is denoted as Total U and defined as $\Sigma_{TR\in DTU}$(T_R)

Definition 5 (Absolute utility of an itemset in a database). The absolute utility of an itemset X in D is defined as au(X) = $\sum_{X\subseteq TR\wedge TR\in D} U(X,T_R)$. The (relative) utility of X is defined as u(X) = au(X)/Total U.

Definition 6 (High utility itemset). *An itemset X is called high utility itemset iff u(X) is no less than a user-specified mini- Minn utility threshold min_utility (0% < ininvtii <100%). Otherwise, X is a low utility itemset. An equivalent definition is that X is high utility iff au(X) abs_min_util, where abs_min_util is defined as min...util x Total U*

Definition 7 (Complete set of HUIs in the database). Let S be a set of itemsets and a function f_H(S)= {X|N $\in$ S, u(X) $\geq$ min_utility}. The complete set of HUIs in D is denoted as H(H$\subseteq$L) and defined as f_H(L). The problem of mining HUIs is to find the set H in D

.

Example 1 (High Utility Itemsets). Let Table I. be a database containing five transactions. Each row in Table 1 represents a transaction, in which each letter represents an item and has a purchase quantity (internal utility).

Example 1 (High Utility Itemsets). Let Table 1 be a database containing five transactions. Each term in row in Table 1 represents a transaction, in which each letter represents an item and has a purchase quantity (internal utility).The unit profit of each item is shown in Table 2 (external utility).In Table 1, the absolute utility of the item au{F} in the transaction T3 is au{F},T3)=p({f},D)*q({f},T3)=3*2=6.The absolute utility of {BF} in T3 is au({BF},T3)=au({B},T3)+au({F},T3)=1+6=7. The absolute utility of {BF} is u(BF},T3)+u{BF},T5)=17. If abs_min_ utility = 10, the set of HUIs in Table 1 is H ={E 12},{F 15},{AE:10},{AF:17}.{BE:10},{BF:17}.{ABE12},{ABF:19} where the number beside each itemset is its absolute utility.

It is said that a superset of a low utility itemset can be high utility itemset and a subset of a HUI can be low utility itemset. Hence, we cannot directly use downward closure property to prune the search space. To facilitate the mining task introduced the concept of transaction-weighted downward closure (TWDC) [17], which is based on the following definitions.

Definition 8 (TWU of an itemset). The transaction-weighted (TWU) of an iteinset X is the sum of the transaction utilities of all the transactions containing X, which is denoted as TWU(X) and defined as TWX = $\sum_{X \subseteq TR \wedge T_R \in D} TU(T_R)$

Definition 9 (HTWUI). An item-set X is a high transaction weighted utilization item-set (HTWUI) iff TWU(X)≥ abs_min_utility. An HTWUI of length K is abbreviated as K- HTWUI.

Property 1 (TWDC Property). The transaction-weighted downward closure property, slates that for au' item-set X that is not a HTWUI, all its supersets are low utility item-sets 12], 117],[19], [24].

Example 2 (TWDC Property). The transaction utilities of 'and T_1 are T_3 are $TU(T_i)$ = =au({ABE},T_1) = 5 and TU(T_3) = 8. If abs_min_utility=10, {AB} is a HTWUI since TWU({AB}) = TU(T_1) +TU(T_3) =13 is not a HTWUI, and therefore all the supersets of (W} are low utility (Property 1).

Closed Itemset Mining

In this section introduce closed itemset mining.

Definition 10 (Tidset of an itemset). The Tidset of an itemset X is denoted as g(X) and defined as the set of Tids of transactions containing X. The Support count of X is expressed in terms of g(X) as SC(X) =\g(X)|.

Property 3. For item-sets X, Y ∊ L. SC(X ∪ Y) = \g(X)∩ g(Y)|.

Definition 11 (Closure of an itemset). The closure of an itemset X∊ L, denoted as C(X), is the largest set Y ∊L such that X ⊆ Y and SC(X)= SC(Y). Alternatively, it is defined as C(X) = $\cap_{R \in g(X)} T_R$

Property 3. ∀X ∊ L,SC(X) = SC(C(X))⇔ g(X) = g(C(X)).

Definition 12 (Closed itemset). An itemset X ∊ L' is a closed itemset iff there exists n itemset Y ∊ L such that X ⊆ Y and SC(X) = SC(Y). Otherwise, X is non-closed itemset. An equivalent definition is that X is closed iff C(X) = X.

For example, in the database of Table I, {B} is non-closed because C({ B}) = $T_1 \cap T_2 \cap T_3 \cap T_5$ = {AB}

Definition 13 (Complete set of closed itemset in the data base). Let S be a set of item-sets and a function fc(S) ={X|X∊ , ¬ ∃ Y∊ **S** such that X ⊂ Y and SC(X) = SC(Y)}.The complete set of closed item-sets in D is denoted as C(C ⊆ L) and defined as fc(L).

For example the set of closed itemsets in table 1 is ={E,3},{EG,1}.{AB,4},{ABE,2},{ABF,2},{ABEW,1} in which the number indicates the support count.The supersets of {B} are {AB,4},{ABE,2},{ABF,2},{ABEW,1}..

III. PROPOSED WORK

3.1 CLOSED HIGH UTILITY ITEMSET MINING

The first point is to discuss is how to incorporate the closed constraint into high utility itemset mining. There are several possibilities. First, define the closure on the utility of itemsets.However, this definition is unlikely to achieve a high reduction of the number of extracted itemsets.A second possibility is to define the closure on the supports of itemsets. In this case, there are two definitions.

— Mine all the high utility itemsets first and then apply the closed constraint.

— Mine all the closed itemsets first and then apply the utility constraint.

Definition 14 (Closed high utility itemset). We define the set of closed high utility itemsets as HC = {X | X ∊ L. X = C(X), u(X) ≥ min...utility}, HC = H' = C. An itemset X is called non-closed high utility itemset iff X ∊ H and X ∉ C.

For example, if atm...min....utility = 10, the complete set of closed HU1s in Table 1 is HC = {{E}, {ABE}, {ABF}}.

Definition 17 (Promising item).
An item i_p is a promising item iff TWU(i_p)≥ abs_min_utility. Otherwise, it is an unpromising item.

3.1.1 Discovery of closed high utility itemsets

The main procedure of CHUD is that it takes as parameter a database D and the min_utility threshold. CHUD first scans D at the same time, CHUD computes the transaction utility for each transaction and calculates TWU of items. When a transaction is retrieved, its transaction id and transaction utility are loaded into a global TU Table named GTU. An item is called a promising item if its TWU is no less than min_utility. Only promising items are kept since supersets of unpromising items are low utility itemsets. The utilities of unpromising items can be removed from the GTU table. Then, CHUD algorithm generates candidates in a recursive manner, starting from candidates containing a single promising item and recursively joining single items to them to form larger candidates itemsets. In phase II on these candidates to obtain all closed+ HUIs.
In this framework first find high utility itemset and apply closed constraint. The main disadvantage of the system is that it does not give result for huge datasets because the computing clousure operation is a slow process. In order to outperform the proposed framework combine both CHUD an another algorithm names FHM: Faster High-Utility Itemset Mining using Estimated Utility Co-occurrence Pruning.

Strategy 1.The first strategy that have incorporated in CHUD is to only consider promising items for generating candidates and remove all the utilities of unpromising items from the global transaction utility table.

Strategy 2. Discarding itemsets having an transaction weighted utility lower than min_utility.

Strategy 3. Removing the Exact utilities of items from
the Global TU-Table

3.2 FASTER HIGH UTILITY ITEMSET MINING

The proposed system is called FHM (Fast High-Utility Miner) used to mine high utility itemsets. This algorithm integrates a novel structure named EUCP (Estimated Utility Cooccurrence Pruning) to reduce the number of operations when mining high-utility itemsets using. The help of EUCP its able to prune some non dependant itemsets from the structure by that closed high utility itemsets can be easily recovered.

Tid	Transactions
T_1	(a,1)(c,1)(d,1)
T_2	(a,2)(c,6)(e,2)(g,5)
T_3	(a,1)(b,2)(c,1)(d,6),(e,1),(f,5)
T_4	(b,4)(c,3)(d,3)(e,1)
T_5	(b,2)(c,2)(e,1)(g,2)

Item	a	b	c	d	e	f	g
Profit	5	2	1	2	3	1	1

Fig 2: Transaction database and utility values

TID	TU
T_1	8
T_2	27
T_3	30
T_4	20
T_5	11

Item	TWU
a	65
b	61
c	96
d	58
e	88
f	30
g	38

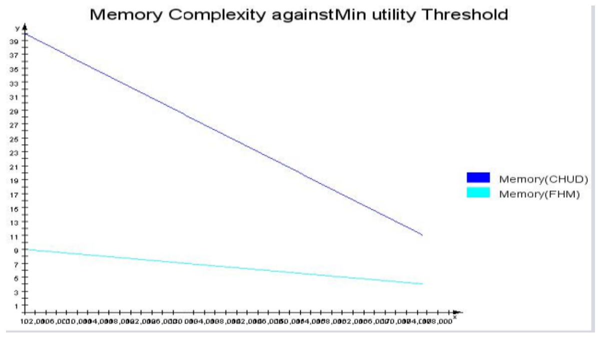
Item	a	b	c	d	e	f
b	30					
c	65	61				
d	38	50	58			
e	57	61	77	50		
f	30	30	30	30	30	
g	27	38	38	0	38	0

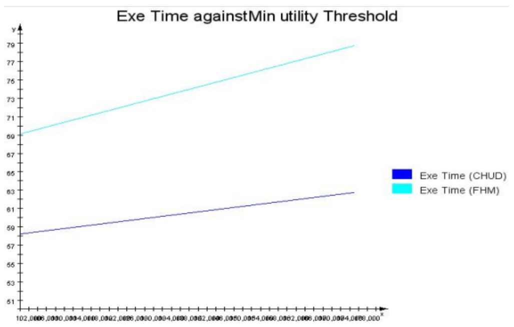
Fig: 3 Transaction utility(left) TWU(center) EUCP(right)

The rest of the steps are same as that of closed high utility itemset mining algorithm.

IV. RESULTS

The graph below plotted gives the comparison of CHUD and FHM , It shows that the memory requirement is small and execution time is faster inproposed algorithm.It increases efficiency effectiveness of the system





V. CONCLUSION

. In this paper addressed the problem of redundancy in high utility itemset mining by compact representation by identify set of closed high utility itemsets .Besides CHUD is faster than UP Growth one of the method to mine high utility itemsets from databases but it is slower than FHM .The one of the disadvantage of the system is that it does not perform well when the number of datasets are large.The proposed algorithm FHM overcome this by incorporating a novel strategy named Estimated utility co-occurrence structure.Hence by the execution speed ,memory requirement and profit can be high

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, pp487-499,1994.

[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. InIEEE Transactions on Knowledge and Data Engineering, Vol. 21, Issue 12, pp. 1708-1721, 2009.

[3] J. -F. Boulicaut, A. Bykowski, and C. Rigotti. Free-sets: a condensed representation of boolean data for the approximation of frequency queries. In Data Mining and Knowledge Discovery , Vol. 7, Issue 1, pp. 5–22

[4] C.-J. Chu, V. S. Tseng, T and Liang. An efficient algorithm for mining temporal high utility itemsets from data streams. In Journal of Systems and Software Vol. 81, Issure 7, pp. 1105- 1117, 2008.

[5] C.-J. Chu, V. S. Tseng, and T. Liang. An Efficient Algorithm for MiningHigh utility Itemsets with Negative Values in Large Databases. In Applied Mathematics and Computation, Vol. 215, Issue. 2, pp. 767-778, 2009.

[6] R. Chan, Q. Yang, and Y. Shen. Mining high utility itemsets. In Proc. Of IEEE Int'l Conf. on Data Mining, pp. 19-26, 2003.

[7] A. Erwin, R. P. Gopalan, and N. R. Achuthan. Efficient Mining of High utility Itemsets from Large Datasets. In Int'l Conf. on PAKDD, pp. 554- 561, 2008.

[8] K. Gouda and M. J. Zaki. Efficiently mining maximal frequent itemsets. In Proc. of IEEE Int'l Conf. on Data Mining, pp. 163 170, 2001.

[9] 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.

[10] H.-F. Li, H.-Y. Huang, Y.-C. Chen, Y.-J. Liu and S.-Y. Lee. Fast and Memory Efficient Mining of High Utility Itemsets in Data Streams. In Proc. of IEEE Int'l Conf. on Data Mining, pp. 881- 886, 2008.

[11] B. Le, H. Nguyen, T. A. Cao, and B. Vo. A Novel Algorithm for Mining High utility Itemsets. In Proc. of First Asian Conference on Intelligent Information and Database Systems, pp.13-17, 2009.

[12] Y. Liu, W. Liao, and A. Choudhary. A fast high utility itemsets mining algorithm. In Proc. of the Utility-Based Data Mining Workshop, pp. 90-99, 2005

[13] C. Lucchese, S. Orlando and R. Perego, "Fast and Memory Efficient Mining of Frequent Closed Itemsets," In IEEE Transactions on Knowledge and Data Engineering, Vol. 18, Issue 1, pp. 21-36, 2006.

[14] V. S. Tseng, C.-W. Wu, B.-E. Shie, and P. S. Yu. UP-Growth an efficient algorithm for high utility itemset mining. In Proc. of Int'l Conf. on ACM SIGKDD, pp. 253–262, 2010.