International Journal of Engineering & Technology, 7 (4) (2018) 5188-5195

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i4.11722
Research paper

Mining correlated high utility-frequent association rules under various discount notations

Kanakamedala Vineela [1] *, D. S. Bhupal Naik [1]

[1] Department of Computer Science and Engineering, VFSTR University, Andhra Pradesh, India
**Corresponding author E-mail:*

Abstract

Association analysis is effective to explore relationships or similarities between items that are concealed in massive datasets. The uncovered associations can be characterized as association rules. i.e. discovering new-opportunities for cross-selling the product. Various algorithms elaborate high utility association rules as positive utility values. In real-life appliances, however, a high utility association rules may be associated with items having negative utility values when discounts are considered for certain products. This abundantly hinders their profits for various real-time appliances such as cross-selling or product recommendations so, finding high utility-frequent itemsets under various discount notations is significant for exploring interesting patterns. Also, a well-known constraint of association rules that are determined by using frequent patterns or utility patterns is that, they do not yield a measure of lift to find correlation between items. In this paper, we introduce a novel algorithm called HUFARM-N (High utility-frequent association rule mining with Negative utility values) which incorporates several expansions to mine high utility-frequent association rules that can meet the business profits ensuing to firms. Empirical analysis on real world datasets exhibits that, HUFARMN is highly capable and also enhances both execution time and memory usage.

Keywords: *Association Rules; Frequent Pattern Mining; Utility Mining.*

1. Introduction

1.1. Mining association rules

Association rule mining (ARM) [1] is mainly useful for analyzing and predicting customer's behavior. It is a prominent procedure for discovering co-occurrences, associations, correlations, frequently generated patterns [2-3] or utility generated patterns[4-6] through distinct items in transaction dataset. For example, frequent patterns could be discovered by scrutinizing retail data [7-8] and then all association rules could be developed by anticipating of buying another item by the confidence.

Association rules can be stated as P➔Q, where P and Q are the itemsets from that we can yield pattern as customers who purchase item P also think to buy Q item simultaneously.

Although most of the research has been interested to explore the patterns of high utility itemsets or frequent itemsets or mining association rules between frequent itemsets [9] or high utility itemsets [10] none of these patterns mining approaches doesn't meet the sales manager objective to mine association rules when he is interested in both the aspects of high utility and frequent itemsets.

The high utility-frequent association rule can be utilized for two possible schemes:

For example, assume

Rule 1 :{{ milk (10), sugar (10)} (utility: 50) ➔ {coffee-nuts (10)}} (rule-utility: 100)

Rule 2: {{milk (10), bread (10)} (utility: 50) ➔ {coffee-nuts (10)}} (rule-utility: 90)

Packaging high utility-frequent itemsets: In case of packaging high utility frequent itemsets, individual package is assumed to be collection of high utility itemsets that are frequently sold together. In the above example, {milk, sugar} is a frequent itemset that is sold with high utility: 50 with its each item's unit profit (e.g. milk is 1.5$ and sugar is 3.5$). {milk, bread} is a frequent itemset that is sold with high utility 50 with its each item's profit (e.g. milk is 1.5$ and bread is 2.5$).

Cross-selling: If cross-selling campaign is organized in the form of rules, items to be purchased would be unalike from those in the packaging of high utility frequent itemsets. The rules that are explored in association analysis are expected in choosing items to recommend to purchaser who bought frequent items previously. In the above example, the rule for cross-selling is taken as {milk, sugar } is assumed to be sold as packaging high utility frequent itemset purchased by the purchaser according to the rule {coffee-nuts} can be selected for the recommendation.

The revenue of cross-selling depends upon transaction weighted utilization of packaging high utility frequent sets rule 1:{milk , sugar}:50 is greater than transaction weighted utilization of rule {milk, sugar, coffee-nuts}:100 with each item's unit profit(e.g. milk is 1.5$,sugar is 3.5$,coffee-nuts is 5$s). rule2: {milk, bread}:50 is greater than transaction weighted utilization of rule {milk, bread, coffee-nuts}:90 with each item's unit profit (e.g. milk is 1.5$, bread is 2.5$, coffee-nuts is 5$s) of these two rules, rule 1 is taken as it gives high profits when {coffee-nuts} recommends to packaging of high utility frequent itemsets {{milk (10), sugar (10)}.

And also many algorithms for high utility itemset mining are not designed to hold items with profits that can differ under various discount notations[11] in real-life chain stores, the profits obtained from items on sale may depends on cost price, tag price, and discount notations .Various outlets may use various discount notations to sell the similar products, The conventional way of computing the utility of itemsets on various discount notations is to examine not only the positive utilities gained by the sale of items but negative utilities [12] also. For example, if a customer buy a mobile then he will receive one bluetooth at free of cost in promotional offer. In this case the mobile store loses the profits for each unit of bluetooth may yield negative utility for the marketers. Nevertheless, these items are frequently cross-promotes with another items having positive utility gain, all-together gets the positive profits. Determining high utility itemsets and high utility association rules under different discount notations is mostly beneficial since discounts happens reality in retail stores.

A novel framework is proposed that can mine all the association rules between high utility-frequent itemsets [13], [14] with user specified thresholds a new property is incorporated named upper-bound transaction weighted utilization that can extend the framework to mine high utility-frequent association rules under different discount notations. To address the efficiency some pruning strategies are used. The major contributions of the proposed architecture as in Fig.1 as outlined below.

The raw datasets transactional database, price table and discount notation table are transformed into Upper-bound transactional utility data when any of item meets negative utilities. These Upper-bound transactional utility data is represented in "matrix" data structure [15]. The columns of matrix represent "Items" and row of matrix represents "transactional id's". The matrix entries are represented in binary format. The presence of an item is represented as "1" and absence as "0". Generally retail data is represented in sparse format. To reduce the dimensionality, "Sparse matrix" data structure is used for efficient usage of memory.

Association rules are generated with user defined minimum support and minimum confidence thresholds. A property named "anti-monotonicity" is used to prune the candidates for generating frequent itemsets[16]. A " upper-bound transaction weighted downward closure" property is used for computing the transaction weighted utilization of frequent itemsets indicates those items which do not satisfy the minimum utility threshold [17 - 20], supersets of these items are considered as low transaction weighted utilization and pruned. To compute upper-bound transaction weighted utilization, a strategy named "apply" function is used to avoid explicit usage of loop constructs and time complexity is reduced. An interesting measure named "Lift" is used to identify correlation [21], [22] between itemsets. A strategy named "maximal" is used to find the most significant rules. Let assume, an association rule A➔ B has two parts. The part that is on left hand side of rule is antecedent itemset (named as packaging of high utility-frequent itemset) and the part that is on right hand side is consequent item (predictable as high utility item) for cross-selling campaign thus making an interesting association rule .In the above example, Rule1 :{milk, sugar} is antecedent itemset and a consequent item {coffee-nuts} is a high utility predictable item for cross-selling campaign thus making an interesting high utility-frequent association rule.

The proposed method has two levels filter level and refine level. In the filter level, all potential high utility-frequent association rules are generated where consequent of rule is dominated by antecedent of high utility frequent itemsets. All the rules are finally substantiated in the refine level.

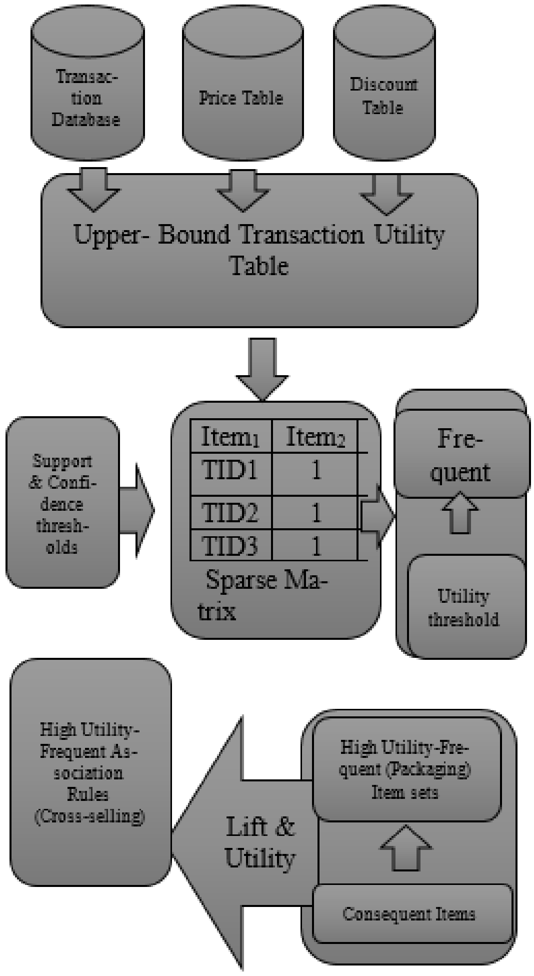

Fig. 1:

2. Related work

Let A= $\{a_1, a2, a_3 \ldots a_m\}$ be a set of finite discrete items. D= $\{T_1, T_2, T_3 \ldots T_n\}$ is a set of transactions in transaction database where each transaction $T_n \epsilon D$ is a subset of A and has unique transaction identifier n called TID. Each item $a_m \epsilon T_n$ has a positive quantity value called Q (a_m, T_n) is internal utility in T_n. Each item is $a_m \epsilon$ A is associated with a positive unit profit value P ($a_{m, D}$) called external utility. An itemset C= $\{a_1, a_2 \ldots a_N\}$ is a set of N distinct items where $a_N \epsilon A$. An N-itemset is an itemset of size N. A rule C ➔E is an association rule where $C \subset A$, $E \subset A$, $C \cap E = \emptyset$.A frequent itemset B= {E, F} holds in transaction database D with minimum support S, if Spercentage of transactions in database D that contain $S \cup R$. The rule E➔ F confined in transaction database D with minimum confidence C, if Cpercentage of transactions in database D contain E also contain F.

Table.1: Transaction Database

Transactions	Items
TID1	a(6),c(4),e(8)
TID2	d(2),f(4)
TID3	a(2),b(6),c(2),d(6),f(2)
TID4	b(2),d(2),f(6)
TID5	b(2),c(8)
TID6	a(4),b(12),c(6),d(8),f(2)
TID7	c(2),d(4),e(10)
TID8	b(4),e(2)
TID9	b(4),d(2),f(2)
TID10	a(8),b(2),d(2),e(6)

Definition 1: *(Price of Item): The cost and tag prices of an item I_j are respectively marked as $cp(I_j)$ and $tp(I_j)$ .In general, the tag price of an item is always more than its cost price.*

For example, consider the Price table of distinct items in Table.2 cp(f) is 10 and its tag price is 18.

Table.2: Price Table

Item	Cost Price	Tag Price
A	15	35
B	4	10
C	70	128
D	10	20
E	19	25
F	10	18

We have various discount strategies in real-life situations they are specified as
Discount notation 1:
An item is sold out with different discount levels with ranges from 0% to 100%.
Discount notation 2:
If the purchaser purchases P units of an item, purchaser will gain R units free of this item.
Discount notation 3:
If the purchaser purchases P units of an item, purchaser will gain r % discount on every supplementary unit bought of this item.
Definition2 (Discount notation table): A

Table.3: That Indicates Various Discount Notation Holds for Each Item

Item	SG_I	$Value_{(V1)}$	$Value_{(V2)}$
a	1	0.50	-
b	1	0.75	-
c	2	3	1
d	1	0	-
e	3	2	0.6
f	1	0.75	-

Let quantity of an item I_j in transaction T_q is represented as Q (I_j, T_q) .Let tp(I_j) and cp(I_j) be the tag and cost prices of an item I_j from Table. Let u(I_j, T_q) be the utility of an item I_j in transaction T_q defined with various discount strategies as below

SG_1: u (I_j,T_q) = Q(I_j,T_q) * (tp(I_j) * $Value_1$ − cp(i_j)).

SG_2 : u(I_j,T_q) =($\frac{Q(Ij,Tq)}{V1+V2}$ * V_1 + Q(I_j,T_q) mod (V_1 +V_2)) * tp(I_j) – Q(I_j ,T_q) * cp(I_j) .

SG_3:

$$\begin{cases} V1 + (Q(Ij,Tq) - V1) * V2) * tp(Ij) - Q(Ij,Tq) * cp(Ij), if\ Q(Ij,Tq) \geq Value1 \\ Q(Ij,Tq) * (tp(Ij) - cp(Ij)), otherwise \end{cases}$$

For example,
SG_1: Utility of an item a in TID1 is computed as u (a, TID6) = 4*(35*0.50-15) =10, where item a is sold at 50% discount.
SG_2: Utility of an item c in TID7 is computed as

U(c, TID7) = ($\frac{2}{3+1}$ * three + [2] mod (3+1)) * 128 – 2*70 is 117.5.

SG_3: Utility of an item e in TID8 is computed as:

U (e, TID8) = (2+ (2-2)*0.6) * 25 -2*19 = -36.

Here 2 unit of an item {e} is purchased in TID8, which is smaller than the parameter of
SG_3= (-36<2).
By, applying these strategies on Table.1, Table.2 and Table.3 an external utility where each item with corresponding unit profits are generated Table contains transactions with items and their profits.

Table.4: Transaction Utility Table with Each Item Profits

Transactions	Items with Profits
TID1	a(15),c(-277),e(-12)
TID2	d(-20),f(14)
TID3	a(5),b(21),c(117.5),d(-60),f(7)
TID4	b(7),d(-20),f(21)
TID5	b(7),c(-554)
TID6	a(10),b(42),c(-159.5),d(-80),f(7)
TID7	c(117.5),d(-40),e(-20)
TID8	b(14),e(-36)
TID9	b(14),d(-20),f(7)
TID10	a(20),b(7),d(-20),e(-4)

Definition 3: *(Utility of item): The quantity of individual item in i_p in transaction database multiplied with their external utility of unit profit table. For example, u (a, TID1) = 15.*

Definition 4: *(Utility of item set in database): Utility of item-set Z in transaction database T_q, $\sum_{ip\ \epsilon Z}$ U (i_p , T_q) where, Z= {i_1 ,i_2 ,$i_{3...}i_L$} is a L-item set ,Z⊆T_q and 1≤L≤m .For example, u(e) =-12-20-36-4 = -72.*

Definition 5: *(Transaction Utility): The TU of transaction T_q, is represented as total sum of the utilities of all the items in transaction database T_q: TU (T_w) =$\sum_{ip\ \epsilon Z}$ u (i_p, T_q) is the transaction utility in transaction database Table.4*

Table.5: Transaction Utility Table

Transactions	Transaction utilities
TID1	-274
TID2	-6
TID3	90.5
TID4	8
TID5	-547
TID6	-180.5
TID7	57.5
TID8	-22
TID9	1
TID10	3

For example, TU (TID2) = -20+14=-6

Definition 6: *(utility of an Itemset with/without negative unit profits): It was observed that in Table.4 an item may have negative utilities also. The question of high utility itemset mining with negative profits is to explore all high utility itemsets in transaction database where external utility profits can be negative or positive.*

Property1 (Utility of an items having negative utility units) :
It can be observed that from Table.4 utility may consists of items having negative profits also.
For example, an itemset {d} from Table having negative profits.
Property2 (Utility may hold at least an item having positive utility unit):
However, an utility may or may not consists of an item having negative profits, an utility may hold at least an item having a positive utility unit else its utility unit prefer to be negative utility unit and it is not considered as a HUI.

Definition 7: *(Reformulate transaction utility): The TU of transaction T_q, is represented as total sum of the utilities of all the items in transaction database T_q consists of positive utility unit : TU(T_q) =$\sum_{ip\ \epsilon Z \wedge p(ip) > 0}$ u (i_p, T_q).The reformulate transaction utilities of transactions TID2, TID3, TID4, TID6, TID7, TID9, TID10 in below Table.6.*

Table.6: Upper-Bound Transaction Utility Table

Transactions	Upper bound Transaction utilities
TID1	15
TID2	14
TID3	150.5
TID4	28
TID5	7
TID6	59
TID7	117.5
TID8	14
TID9	21
TID10	27

Definition 8: *(Transaction weighted utilization): The TWU of itemset Z, represented as TWU (Z), is the total sum of reformulate transaction utility of all transactions consisting Z: TWU (Z) =* $\sum_{Z \subseteq Tq \in D}$ *TU* (T_q)*. For example, TWU (a) = 251.5.*

Definition 9: *(Support): Relative frequency of all transactions* $T_{p,}$ *in transaction database that contains both a and b items.*

Support (S) = Relative frequency/Total transactions

For example, how frequently {d, b} occurs together in total transactions Table 1 in transaction database D^*.

Support= 5 / 10 = 0.5.

Definition 10: *(Confidence): It measures how frequently each item in B occurs in transaction that contains A.*

Confidence (C) = S (a∪b) / S (a).

For example, how frequently {a} occurs in transaction $T_{q \text{ that}}$ contains {b}

Confidence (a →b) = S (a∪b)/ S (a) = 0.7.

Let assume,
Buys (Y,” Computer games”) →Buys (Y,” Videos”) [Support 50%, Confidence 70%]
This rule is significant but it is confusing. The probability of purchasing Videos is 75% which satisfies the min_conf but in reality Computer games and Videos are negatively correlated to each other because buying of one of these products actually declines the likelihood of buying the other then the confidence of the rule Computer games → Videos can be misleading .It only estimates the conditional probability of itemset{Videos} given itemset{Computer games}.It does not compute the significance of the correlation measure between these two products. In this context “Lift” measure is used to find Correlation between two items.

Definition 11: *(Lift): The lift of rule A→B is the ratio of confidence of rule with the expected confidence; imagine if both item-sets A and B are independent.*

Lift = Confidence / Expected confidence.

Elucidation of lift:
The value of lift with greater than 1 indicates that A and B appears more frequently together than expected. That means occurrences of both item-sets A and B has a positively correlated to each other.
The value of lift with smaller than 1 indicates that A and B appears less frequently together than expected. That means occurrences of both item-sets A and B has a negatively correlated to each other.
For example, assume that {d} and {b} are independent item-sets then lift of these item-sets:
Lift (a →b) = Confidence (a→b) / Support (b) = 0.7 / 0.7= 1
A and b has no affect to each other.
Lift (b →c) = Confidence (b→c) / Support (c) = 0.4/0.5=0.8 < 1
B and c are negatively correlated to each other.
Lift (a →f) = Confidence (a→b) / Support (b) = 1.25
A and f are positively correlated to each other.
Definition12 (Association Ruleoccurrence):
Association rules are explored by scrutinizing the data for frequent patterns (if/then) by incorporate the interest measures support, confidence and lift to mine the association relationships between them. It can be expressed as A→ B. Here, A is an antecedent and B is a consequent.
An antecedent is an itemset frequently revealed in the transaction data. Consequent is an itemset that depends in association with the antecedent.For example, assume that {a} and {f} are independent two itemsets. The rule between them can be generated as: Lift (a→f) = Confidence (a→f) / Support (f) = 1.2. a and f are positively correlated to each other. That means, there is an interesting relationship between these items i.e. a→f, the customers who purchase item {f} are also interested to purchase item {a}.

3. Proposed method

A novel approach named Mining All high utility-frequent association rules to predict high utility product from basket of high utility-frequent itemsets thus, making interesting rule.

3.1. Mining correlated high utility-frequent association rules

The proposed approach is designed in two levels.
Filter level:
Property 1(Anti-monotonicity):
Frequent item-sets are explored by user specified minimum support value for each frequent item-set all its subsets are also frequent. For example, specify minimum support is 0.2.
Form Table.1 transaction database calculate then frequency of itemset {a, d} is 0.3 which is frequent itemset all its subsets are also frequent itemsets and consider {a, d, f} = 0.2 is also frequent itemset.
Property 2 (Upper-Bound downward closure property):
Upper bound downward closure property illustrates that any superset of transactions which are having under the specified minimum utility threshold consider it as low weighted utilization only composites of high transaction weighted itemsets are listed.
For example, calculate the Transaction weighted utilization of singleton itemsets Table.6 Transaction database.

Table.7: UBTWU of Singleton Items

Items	UBTWU
A	251.5
B	306
C	348.5
D	416.5
E	159.5
f	272

Specify minimum-utility is 200.0 only a, b, c, d , f item-sets are high TWU item-sets and item-set e is eliminated as it is low TWU and super sets of {e} composites are also low transaction weighted utilizations and removed.
Strategy3 (Sparse matrix representation):
The proposed method read customer transaction data where each transaction may contain presence (1) and absence (0) of items. Items which are not present in each transaction may wastes memory to minimize memory usage dynamically the sparse data will coerce to sparse matrix representation only non-zero rows are represented. For example,

Table.8: Sparse Matrix Representation

	a	b	c	d	e	f
TID1	1	0	1	0	1	0
TID2	0	0	0	1	0	1
TID3	1	1	1	1	0	1
TID4	0	1	0	1	0	1
TID5	0	1	1	0	0	0
TID6	1	1	1	1	0	1
TID7	0	0	1	1	1	0
TID8	0	1	0	0	1	0
TID9	0	1	0	1	0	1
TID10	1	1	0	1	1	0

The above Table.8 is transaction data where each transaction contains presence and absence of items. Sparse matrix representation coerces the data only non-zero rows are identified. A vector with row-wise indices of non-zero items is 1,3,5,4,6,1,2,3,4,6,2,4,6,2,3,1,2,3,4,3,4,5,2,5,2,4,6,1,2,4,5 and the pointers where each row of index vector start

2,4,6,7,8,9,11,17,19,21,23,25,28,29,30,35,37,38,42,43,45,46,48,49 ,51,53,57,60. . It can also name as item-Matrix representation

Strategy4 (maximal Rule):

Maximal item-set states if all the subset and superset contained in the item-set must be at the same frequency. Rules that are generated by the maximal item-set are considered as significant rules.

For example, assume {a} item set its frequency is 4 and its subsets {a, e}, {a, b, e} frequency is also 4 and consider it as maximal itemsets. Rules that are defined from these item-sets are maximal rules.

Strategy5 (apply function):

Apply function in R programming is another looping method which runs faster than ordinary looping. Using apply in programming is very powerful to reduce time complexity. Apply function applies to margins of matrix either row (1) or column (2) or both [1:2]. Generally, it is used to simplify aggregate functions and returns result as list or vector.

For example, Table.7 calculate the sum of UBTWU of all items Apply (TWU, 2, sum) gives 1754.Here 2 indicates column wise.

Definition8 (High utility-frequent itemsets):

An itemset may be high utility frequent itemset if it satisfies utility threshold (min_util), minimum support thresholds and all its supersets are also high utility-frequent itemsets.

For example, assume {a, b} itemset the frequency of {a, b} is 0.3 ≥ min-sup and utility of {a, b} itemset is 236 ≥ min-util.

Definition9 (Extendibility):

A High utility-frequent item can extend rule by antecedent (l.h.s) expansion in database D^* by adding item one by one in the itemset corresponds to lexicographical order.

For example, assume the rule {c, d} →{e}. This rule generates by considering the l.h.s expansion.

An l.h.s expansion of {c} →{e} with the item {d} can outcome in the rule {c, d} →{e}.

Definition10 (Upper-Bound Transaction Weighted Utilization of antecedent itemset):

The antTWU of itemset G, represented as antUBTWU (G), is the total sum of transaction utility of all high utility-frequent itemset G:

antUBTWU (G) = $\sum_{G \subseteq T_q \in D}$ TU (T_q).

For example, assume the rule {a, c} →{f} where {a, c} is the antecedent high utility frequent itemset and its antUBTWU is 224.

Definition11 (Upper-Bound Transaction-weighted utilization of consequent rule):

The conUBTWU of rule R, represented as conUBTWU(R), is the union of sum of transaction utility of all consequent of rule R : conUBTWU(R) =$\sum_{R \subseteq T_q \in D}$ TU(T_q).

For example, assume the rule {a, c} →{f} where lift is greater than 1 means they are positively correlated and identifies for every transaction of item {f}, itemset {a, c} also occurs. The transaction weighted utilization of the rule R is, conUBTWU ({a, c, f}) is sum total of transaction utilities of transactions containing these itemset{a, c, f} which occurs in transactions 3 and 6 is 209.

Definition12 (High-Utility-frequent Association Rule):

A rule R can be high utility-frequent Association rule if consequent of rule is greater than minimum utility (conUBTWU(R)> min_util).

For example, conTWU ({a, c, f}) ≥ min_util and it is High Utility-Frequent Association Rule.

The main procedure of Mining All High Utility-frequent Association rules in Filter level:

The algorithm scans the transaction database, price table and discount notation table (Method 3) and calculate the upper bound transaction weighted utilization of singleton itemsets from Table.6 if they meets negative utility of itemsets and the minimum utility threshold is set by the user those itemsets which does not meet the specified utility threshold is discarded from the transaction database in Table.7. Itemset{e} does not meet the utility threshold and item {e} is removed from the transaction database. Then, algorithm scans for generating Frequent-itemsets (Method 1) those itemsets which does not meet the specified thresholds (min_sup=0.2 and min_conf=0.6) are pruned and the upper bound transaction weighted utilization of frequently generated itemsets are computed those itemsets which doesn't meet the utility threshold (min_util=200.0) are pruned and considered as high–utility frequent itemsets (18) High-Utility frequent itemsets (HUFIM) are generated in Table.9). The high utility-frequent itemsets are extended by left expansion with items in Table.1 in lexicographical order with (Method 2) and are ready for rule expansion (antecedent itemsets) with consequent items (each distinct items) in transaction database Table.1 and check whether they are positively correlated to each other or not by using the Lift measure.

Table.9: High Utility-Frequent Itemsets between Correlated Items

S.No	HUFIM	Support	Confidence	Lift
1	{a}→{b}	0.3	0.7	1
2	{a}→{c}	0.3	0.7	1.75
3	{a}→{d}	0.3	0.7	1
4	{a}→{f}	0.2	0.5	1.25
5	{a,b}→{c}	0.2	0.6	1.5
6	{a,b}→{f}	0.2	0.6	1.2
7	{a,c}→{f}	0.2	0.6	1.2
8	{a,d}→{f}	0.2	0.6	1.2
9	{a,b,c}→{d}	0.2	1	1.4
10	{a,b,c}→{f}	0.2	1	2
11	{a,b,d}→{f}	0.2	1	2
12	{b}→{d}	0.3	0.7	1
13	{b}→{f}	0.4	0.5	1
14	{b,c}→{f}	0.2	0.6	1.2
15	{b,c,d}→{f}	0.2	1	2
16	{c}→{d}	0.3	0.7	1
17	{c}→{f}	0.2	0.5	1
18	{c,d}→{f}	0.2	0.5	1

From Table.9 the high utility-frequent itemsets that are extended with consequent items thus making a high utility-frequent association rules, that are generated are greater than 1 when computing the correlation between items by using "Lift". That means, these itemsets are positively correlated to each other. Then compute the upper-bound transaction weighted utilization (Rule Utility) those rules which are greater than minimum utility threshold are considered for cross-selling campaign.

Table.9: High Utility-Frequent Association Rules between Correlated Items

S. No	High-utility Frequent Rules	Rule utility
1	{a}→{b}	236.5
2	{a}→{c}	224.5
3	{a}→{d}	236.5
4	{a}→{f}	209.5
5	{a,b}→{c}	209.5
6	{a,b}→{f}	209.5
7	{a,c}→{f}	209.5
8	{a,d}→{f}	209.5
9	{a,b,c}→{d}	209.5
10	{a,b,c}→{f}	209.5
11	{a,b,d}→{f}	209.5
12	{b}→{d}	285
13	{b}→{f}	258
14	{b,c}→{f}	209.5
15	{b,c,d}→{f}	209.5
16	{c}→{d}	326.5
17	{c}→{f}	209.5
18	{c,d}→{f}	209.5

Refine phase: Another database scan is required to verify high utility-frequent association rules that are generated in the filter level.

Algorithm:

```
Procedure: HUARM-N
$I_{A}$ = {$a_1, a_2, a_3 .......a_m$} // 1-itemset with high utilities
$I_{C}$= {$a_1, a_2, a_3 .......a_m$} // 1-itemset with high utilities
$R_H$ = Ø // set of high utility association rules
$A_{FH}^{j}$ = Frequent-itemsets ($I_A$)
For each itemset $_{FH}^{j}$ in $A_{FH}^{j}$
$A_{FH}^{j+1}$ = ExtentAntecedent ($A_{FH}^{j}$ , $I_A$ )
do
//Generate high utility association rules, by extending rule of Consequent with Antecedent of frequent itemsets
For each itemset $_{FH}^{j+1}$ in $A_{FH}^{j+1}$
For each item $_C$ in $I_C$ // set of 1-itemsets that extends with Antecedents $A_{FH}^{j+1}$
If Conf $(_{FH}^{j+1} \rightarrow {}_I) \geq$ min_conf
Then
Lift $(_{FH}^{j+1} \rightarrow {}_I) \geq 1$ // Lift greater than equal to 1 is positively correlated items
Then
TWU $(_{FH}^{j+1} \rightarrow {}_I) \geq$ min_util
$R_H = R_H \cup \{ _{FH}^{j+1} \rightarrow {}_I \}$
End if
Next
Loop while $|A_{FH}^{j+1}| > 0$
Next
Loop while $|I_C| > 0$
```

HUFARM-N algorithm

Frequent –itemsets function for HUARM-N algorithm
Procedure: Frequent-itemsets (I_A, D)
Input: set of high utility items I_A, database D, set min_ sup, min_conf, min_util
Output: j-frequent-itemsets A_{FH}^j

```
$A_{FH}^j = \emptyset$ // set of frequent-itemsets
For each pair of items $a_i$ , $a_j$ in $I_A$
If Sup($a_i \rightarrow a_j$) $\geq$ min_sup
Then
Conf ($a_i \rightarrow a_j$) $\geq$ min_conf
Then
TWU ($a_i \rightarrow a_j$) $\geq$ min_util
$A_{FH}^j = A_{FH}^j \cup \{ a_i \rightarrow a_j \}$
End if
Next
$I_A$= ScanTransaction ($A_{FH}^j$, $A_I$, D)
Return $A_{FH}^j$
```

Method 1

Antecedent extension function for HUARM algorithm
Procedure: ExtendAntecedent (A_{FH}^j, I_A, D)
Input: set of j-frequent- items A_{FH}^j, set of high utility items A_I, database D, set min_sup, min_util
Output: j+1-frequent-itemsets A_{FH}^{j+one}

```
$A_{FH}^{j+1} = \emptyset$ // set of j+1- frequent-itemsets
For each itemset $_{FH}^j$ in $A_{FH}^j$
For each item a in ($I_A$- $A_{FH}^j$)
If Sup($_{FH}^j \rightarrow a$) $\geq$ min_sup
Then
TWU ($_{FH}^j \rightarrow a$) $\geq$ min_util
$_{FH}^{j+1} = {}_{FH}^j \cup \{a\}$
$A_{FH}^{j+1} = A_{FH}^{j+1} \cup \{ _{FH}^{j+1} \}$
End if
Next
Next
$A_{FH}^j$= ScanTransaction ($A_{FH}^{j+1}$,$A_I$,$A_{FH}$,D)
Return $A_{FH}^{j+1}$
```

Method 2

Transactions scan method for HUARM-N
Procedure: ScanTransaction (A_I, D)
Input: set of items $_I$ in A_I, transaction database D, Cost price table, tag price table, discount strategy table,set min_util
Output: 1-high utility itemsets I_A with positive profits.

```
$I_A = \emptyset$
$A_I^{i+1} = \emptyset$
$A_I = \{a_1, a_2, a_3 \ldots\ldots a_z\}$
Do
For each item $_I$ in $A_I$
For each transaction TID in D
$X = \{a_I\}$
If UBTWU(X) $\leq$ min_util
Then
$A_I^{i+1} = A_I - \{a_I\}$
End if
Next
Next
Loop While $|A_I| > 0$
// reorganize the transaction database by ascending or descending order by removing the items that don't satisfy the minimum utility
If $_I^{i+1} \subset A_I^{i+1}$
$I_A = I_A \cup {}_I^{i+1}$
$I_C = I_A$
End if
Return $I_A$
Return $I_C$
```

Method 3

4. Experimental results

Spacious experiments were performed to analyze the proposed frame work using Intel Core-i7, 2.70 GHZ of clock speed with 12 GB RAM. The proposed method was implemented in R programming platform with version 3.3.2. The real world datasets like Amazon Bookstore, Online Retail data and Groceries data were taken from UCI and Kaggle.com respectively for performing various experiments.

Table.10: Database Characteristics

Dataset	Transactions	Items	Type
Amazon Bookstore	92108	220447	Sparse
Online Retail data	24426	4189	Sparse
Groceries	9835	169	Sparse

The dataset Amazon Bookstore generates 220447 high utility items with min_util = 25000.These high utility items generate 6156 high utility antecedent frequent-itemsets with min_sup= 0.02 and min_util = 45000.These high utility antecedent frequent itemsets expands the rule with consequent high utility items, generates 191 association rules with min_conf =0.75 later by using maximal rule function and min_util = 50000 generates 63 high utility association rules. The dataset Online Retail data generates 4189 high utility items with min_util=10000 these high utility items generates 1028 high utility-frequent itemsets with min_sup= 0.02 by specifying min_util = 15000 it generates 978 high utility antecedent frequent itemsets. These high utility antecedent frequent itemsets expands the rule with consequent high utility items generates 131 association rules with min_conf = 0.5later by using maximal rule function it generates 125 rules and by specifying min_util=25000 it generates 117 high utility association rules. The dataset Groceries generates 169 high utility items with min_util =5000 these high utility items generate 428 high utility-frequent itemsets with min_sup = 0.01 by specifying min_util =5000 it generates 301 high utility antecedent frequent itemsets. These high utility antecedent frequent itemsets expands the rule with consequent high utility items generates previously 125 association rules with min_conf = 0.5 later by using maximal rule function it generates 69 rules and by specifying min_util =25000 Fig.2. it generates 33 high utility-frequent rules.

In the Table.8, the rule generation ratio (RGENR) is the number of high utility-frequent association rules upon the number of scans that precisely generated. While several rules were explored up on the experimental datasets .The Amazon Bookstore gives 40% (of 191 rules only 63 high utility-frequent rules are explored) are used for cross-selling campaign in Fig.3, Online Retail data gives 89% (of 131 rules only 117 high utility-frequent rules are explored) are used for cross-selling campaign in Fig.4. and Groceries data gives 26% (of 125 rules only 32 high utility-frequent rules are explored) in Fig.5 .The high utility- frequent itemsets are considered as packaging of frequent itemsets. Finally, the rule utilities where consequent of rule utility is greater than minimum utility threshold are qualified for cross-selling campaign that can fulfill the marketer needs.

Table.11: Rgenr of Huarm-N

Dataset	Association Rules	HUFARM	RGENR%
Amazon Bookstore	191	63	40%
Online Retail data	131	117	89%
Groceries	125	32	26%

For mining high utility association rules, rule utilities are computed to avoid explicit usage of loops for complex data structures like matrix, apply function is used for computing an aggregate function like sum thus making less iteration. The performance results are given in Table .12. It can be observed that usage of apply function gives better performance and is efficient.

Table.12: Computational Performance between Huarmloop and Huarmap-ply

Approach	Parameters	Amazon Book Store	Online Retail data	Groceries
HUARM-$_{\text{loop}}$	Rule-Utilities: Total-Time	5306$_{\text{sec}}$	1954$_{\text{sec}}$	590$_{\text{sec}}$
HUARM$_{\text{ap-ply}}$	Rule-Utilities: Total-Time	4898$_{\text{sec}}$	1709$_{\text{sec}}$	295$_{\text{sec}}$

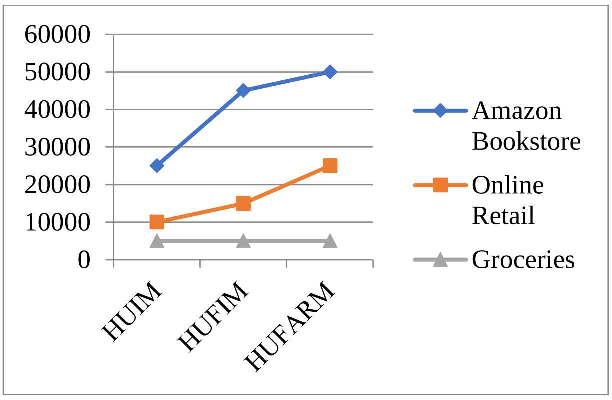

Fig.2: Min_ Util Thresholds Used in Data

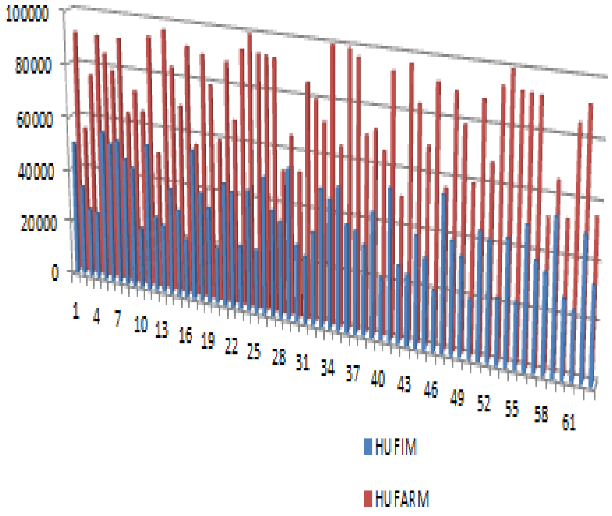

Fig. 3: Top High Utility-Frequent Association Rules for Amazon Bookstore.

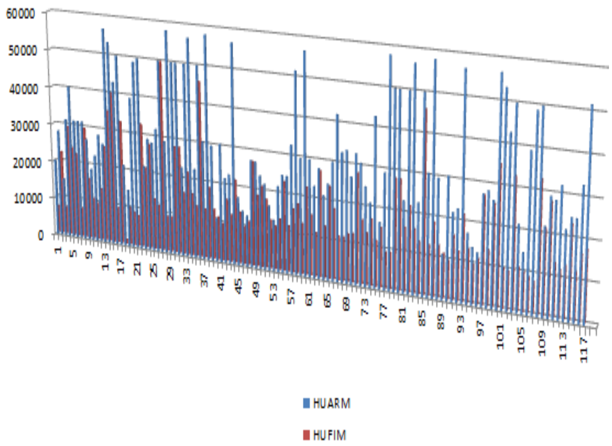

Fig. 4: Top High Utility-Frequent Association Rules for Online Retail Data.

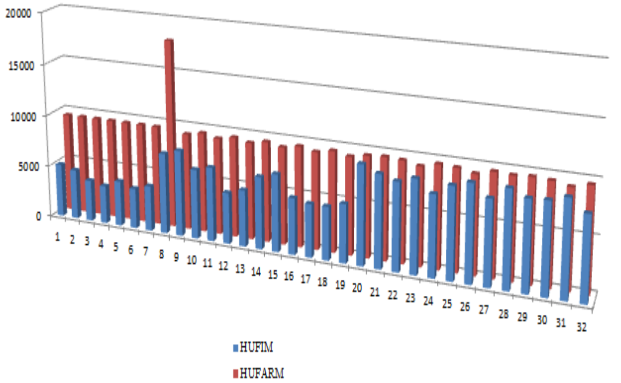

Fig. 5: Top High Utility-Frequent Association Rules for Groceries Data.

5. Conclusion

The proposed method "Mining Correlated High Utility-Frequent Association Rules with various Discount notations", generate high profits. Existing methodologies explores association rules only on frequency of items or utility of items but not focus on items having both the aspects of frequency and utility and also most of the methodologies that are developed under high utility itemset mining does not hold items with profits that can differ under various discount notations happens reality in retail stores. This abundantly impedes their profits for various real-time appliances such as cross-selling or product recommendations. Also a significant constraint is that they do not yield a measure of lift to find the correlation between items this may generate negative correlation between items that conflicts each other. By incorporating the "HUFARM-N" a framework user delineate the preference in both the features of utility and frequency and also identify the correlation between items to retrieve association rules that help them to attain the business goals. The high utility-frequent association rules that are determined can be utilized for schemes like packaging high utility-frequent itemsets, in which individual package is assumed to be collection of high utility itemsets that are sold together and cross-selling campaign is organized in the form of interesting rules where items to be purchased would be unalike from those in the packaging of high utility-frequent itemsets. A different data structure named "sparseMatrix" is adopted to minimize the memory usage and several pruning methods are used to efficiently mine the rules. Empirical analysis on various real world datasets shows how much a business vendor can gross for cross-selling application from the proposed approach.

References

[1] Zaki, Mohammed Javeed. "Scalable algorithms for association mining." *IEEE Transactions on Knowledge and Data Engineering* 12, no. 3 (2000): 372-390. https://doi.org/10.1109/69.846291.

[2] Pasquier, Nicolas, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. "Discovering frequent closed itemsets for association rules." In *International Conference on Database Theory*, pp. 398-416. Springer Berlin Heidelberg, 1999.

[3] Han, Jiawei, Jian Pei, and Yiwen Yin. "Mining frequent patterns without candidate generation." In *ACM Sigmod Record*, vol. 29, no. 2, pp. 1-12. ACM, 2000.

[4] Shankar, S., Nishanth Babu, T. Purusothaman, and S. Jayanthi. "A fast algorithm for mining high utilityitemsets". In *AdvanceComputingConference, 2009. IACC 2009. IEEE International*, pp. 1459-1464. IEEE, 2009.

[5] Zida, Souleymane, Philippe Fournier-Viger, Jerry Chun-Wei Lin, Cheng-Wei Wu, and Vincent S. Tseng. "EFIM: a fast and memory efficient algorithm for high-utility itemset mining." *Knowledge and Information Systems* (2016): 1-31.

[6] Fournier-Viger, Philippe, Cheng-Wei Wu, and Vincent S. Tseng. "Mining top-k association rules." In *Canadian Conference on Artificial Intelligence*, pp. 61-73. Springer Berlin Heidelberg, 2012.

[7] Chen, Yen-Liang, Jen-Ming Chen, and Ching-Wen Tung. "A data mining approach for retail knowledge discovery with consideration of the effect of shelf-space adjacency on sales." *Decision Support*

Systems 42, no. 3 (2006): 1503-1520. https://doi.org/10.1016/j.dss.2005.12.004.

[8] Song, Hee Seok, Jae kyeong Kim, and Soung Hie Kim."Mining the change of customer behavior in an internet shopping mall." *Expert Systems with Applications* 21, no. 3 (2001): 157-168. https://doi.org/10.1016/S0957-4174(01)00037-9.

[9] Liu, Bing, Wynne Hsu, and Yiming Ma. "Mining association rules with multiple minimum supports."In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 337-341. ACM, 1999. https://doi.org/10.1145/312129.312274.

[10] Lee, Dongwon, Sung-Hyuk Park, and Songchun Moon. "High-Utility Rule Mining for Cross-Selling." In System Sciences (HICSS), 2011 44th Hawaii International Conference on, pp. 1-10. IEEE, 2011.

[11] Li, Yao, Zhiheng Zhang, Wenbin Chen, and Fan Min. "Mining high utility itemsets with discount strategies." JOURNAL OF INFORMATION &COMPUTATIONAL SCIENCE 11, no. 17 (2014): 6297-6307. https://doi.org/10.12733/jics20104994.

[12] Fournier-Viger, Philippe. "FHN: efficient mining of high-utility itemsets with negative unit profits." In International Conference on Advanced Data Mining and Applications, pp. 16-29. Springer, Cham, 2014.

[13] Goyal, Vikram, Ashish Sureka, and Dhaval Patel."Efficient skyline itemsets mining." In *Proceedings of the Eighth International C* Conference on Computer Science & Software Engineering*, pp. 119-124. ACM, 2015.

[14] Lin, Jerry Chun-Wei, Lu Yang, Philippe Fournier-Viger, Siddharth Dawar, Vikram Goyal, Ashish Sureka, and Bay Vo. "A More Efficient Algorithm to Mine Skyline Frequent-Utility Patterns." In *International Conference on Genetic and Evolutionary Computing*, pp. 127-135. Springer International Publishing, 2016.

[15] Wu, Chieh-Ming, and Yin-Fu Huang. "Generalized association rule mining using an efficient data structure." *Expert Systems with Applications* 38, no. 6 (2011): 7277-7290. https://doi.org/10.1016/j.eswa.2010.12.023.

[16] Agrawal, Rakesh, and Ramakrishnan Srikant. "Fast algorithms for mining association rules." In *Proc. 20th int. conf. very large databases, VLDB*, vol. 1215, pp. 487-499. 1994.

[17] Yao, Hong, Howard J. Hamilton, and Cory J. Butz. "A foundational approach to mining itemset utilities from databases." In Proceedings of the 2004 SIAM International Conference on Data Mining, pp. 482-486. Society for Industrial and Applied Mathematics, 2004. https://doi.org/10.1137/1.9781611972740.51.

[18] Liu,Ying,Wei-keng Liao, and Alok Choudhary."A two-phase algorithm for fast discovery of high utility itemsets." In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 689-695. Springer Berlin Heidelberg, 2005. https://doi.org/10.1007/11430919_79.

[19] Fournier-Viger, Philippe, Cheng-Wei Wu, Souleymane Zida, and Vincent S. Tseng. "FHM: Faster high-utility itemset mining using estimated utility co-occurrence pruning."In International Symposium on Methodologies for Intelligent Systems, pp. 83-92. Springer International Publishing, 2014.

[20] Zida, Souleymane, Philippe Fournier-Viger, Jerry Chun-Wei Lin, Cheng-Wei Wu, and Vincent S. Tseng. "EFIM: a fast and memory efficient algorithm for high-utility itemset mining." Knowledge and Information Systems (2016): 1-31.

[21] Xiong, Hui, Mark Brodie, and Sheng Ma. "Top-cop: Mining top-k strongly correlated pairs in large databases."In Data Mining, 2006. ICDM'06. Sixth International Conference on, pp. 1162-1166. IEEE, 2006.

[22] Xiong, Hui, P-N. Tan and Vipin Kumar. "Mining strong affinity association patterns in data sets with skewed support distribution." In Data Mining, 2003. ICDM 2003. Third IEEE International Conference on, pp. 387-394. IEEE, 2003.

[23] Fournier-Viger, Philippe, Cheng-Wei Wu, and Vincent S. Tseng. "Mining top-k association rules." In *Canadian Conference on Artificial Intelligence*, pp. 61-73. Springer Berlin Heidelberg, 2012.

[24] Tseng, Vincent S., Cheng-Wei Wu, Philippe Fournier-Viger, and S. Yu Philip. "Efficient algorithms for mining top-k high utility itemsets." *IEEE Transactions on Knowledge and Data Engineering* 28, no. 1 (2016): 54-67. https://doi.org/10.1109/TKDE.2015.2458860.

[25] Zida, Souleymane, Philippe Fournier-Viger, Cheng-Wei Wu, Jerry Chun-Wei Lin, and Vincent S. Tseng. "Efficient mining of high-utility sequential rules." In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pp. 157-171. Springer International Publishing, 2015. https://doi.org/10.1007/978-3-319-21024-7_11.

[26] Yun, Hyunyoon, Danshim Ha, Buhyun Hwang, and Keun Ho Ryu. "Mining association rules on significant rare data using relative support." *Journal of Systems and Software* 67, no. 3 (2003): 181-191. https://doi.org/10.1016/S0164-1212(02)00128-0.