HIGH UTILITY RARE ITEMSET MINING (HURI): AN APPROACH FOR EXTRACTING HIGH-UTILITY RARE ITEM SETS

By

JYOTHI PILLAI * O.P. VYAS **

* Associate Professor, Bhilai Institute of Technology, Durg, Chhattisgarh, India.
** Professor, Indian Institute of Information Technology Allahabad, Uttar Pradesh, India.

ABSTRACT

Association Rule Mining (ARM) is a well-studied technique that identifies frequent itemsets from datasets and generates association rules by assuming that all items have the same significance and frequency of occurrence without considering their utility. But in a number of real-world applications such as retail marketing, medical diagnosis, client segmentation etc., utility of itemsets is based on cost, profit or revenue. Utility Mining aims to identify itemsets with highest utilities by considering profit, quantity, cost or other user preferences.

Rare items are items that occur less frequently in a transaction dataset. High Utility Itemsets may either be frequent or rare. Similarly rare itemset may be of high or low utility. In many real-life applications, high-utility itemsets consist of rare items. Rare itemsets provide useful information in different decision-making domains, customers purchase microwave ovens or plasma televisions rarely as compared to bread, washing powder, soap etc. The former may yield more profit for the supermarket than the latter. Koh and Rountree (2005) proposed a modified apriori inverse algorithm to generate rare itemsets of user interest. In this paper, the authors propose a High Utility Rare Itemset Mining [HURI] algorithm that uses the concept of apriori inverse, for generating high utility rare itemsets of users' interest[Koh and Rountree (2005)]. We demonstrate the approach with a synthetic dataset. Apriori inverse is used to find only the rare itemsets. HURI is used to find those rare itemsets, which are of high utility according to users' preferences, i.e., algorithm for generation of rare itemsets is extended to find high-utility rare itemsets.

Keywords: Association Rule Mining, Utility Mining, Rare itemset, High Utility Rare itemset Mining.

INTRODUCTION

Data Mining involves an algorithmic process, which takes preprocessed input data and extracts patterns. Various techniques exist such as association rule mining, classification, clustering, etc. An important and widely used data mining technique is the discovery of association rules. Association rule mining aims at discovering frequent itemsets from market basket data and generating association rules.

Most association rule mining algorithms implicitly consider the utilities of the itemsets to be equal [Yao H. et al (2004)]. A utility is a value attached to an item depending on its evaluation e.g. if coke has support 20 and profit of 2%, cookies may have support 10 but with a profit of 20%. This means the utility of cookies is higher than coke. Agrawal R. et al (1993) proposed the apriori algorithm that uses simple support–confidence model i.e. first find all frequent itemsets and then generate all association rules satisfying minimum thresholds. Yao et al defined Utility as a measure of how useful or profitable an itemset is [Yao H. et al (2004)]. Liu, Y. et al (2005) stated that frequent itemsets may only contribute a small portion of the overall profit whereas infrequent itemsets may contribute a large portion of the profit [Liu, Y. et al (2005)]. Frequency is not sufficient to answer questions such as whether an itemset is highly profitable or whether an itemset has a strong impact.

Rare itemset mining is a challenging topic in the field of association rule mining as stated by Adda M. et al (2007) and Szathmary L. et.al (2007). In many practical situations, the rare combinations of items in the itemset with high utilities provide very useful insights to the user. Rare itemsets are the itemsets that occur infrequently in the transaction data set. In most business applications, frequent itemsets may not generate much profit while rare itemsets may generate a very high profit.

For example, a sales manager may not be interested in frequent itemsets that do not generate significant profit. Rare itemsets are very important and can be further promoted together because they possess high associations and can bring some acceptable profits [Lan et al]. A modified apriori inverse algorithm was proposed by Koh and Rountree (2005) to generate rare itemsets of user interest.

In this paper, the authors proposed how apriori inverse algorithm can be used in High Utility Rare Itemset Mining [HURI] algorithm. HURI finds high profitable rare itemsets according to user's perspective. They have demonstrated this approach with synthetic dataset in section 2.

The rest of paper is organized as follows. In section 1, we discusses some related works: section 2 presents the HURI algorithm and section 3 presents conclusion and future work.

1. Related Work

In conventional pattern mining, the main target is to find frequent patterns and associations between the items. But in many applications, some items appear more frequently in the data, while others rarely appear. If frequencies of items vary, two problems may be encountered – (1) If minsup is set too high, then rules of rare items will not be found (2) To find rules that involve both frequent and rare items, minsup has to be set very low, where minsup is the minimum support of an item. This may cause combinatorial explosion in the number of itemsets. The basic bottleneck in association rule mining is the rare itemset problem.

Utility mining is now an important association rule-mining paradigm. Adda M. et.al (2007) introduced a good foundational and theoretical model of utility itemset mining, where a utility table UT<I,U> is defined by items I and their utilities U computed for each transaction and termed local utility of a transaction. Utility mining approach was improved by Yao and Hamilton (2006). Some utility approaches have considered performance enhancements to enable handling of large candidate sets, for example, in Kiran and Reddy (2009), which is adopted theoretically from Yao H. et al (2004).

Rare itemsets provide very useful information in real-life applications such as security, business strategies, biology, medicine and super market shelf management. Adda M. et al (2007) shows that normal behavior is very frequent whereas abnormal or suspicious behavior is less frequent. Considering a database where the behavior of people in sensitive places such as airports is recorded, if those behaviors are modeled, it is likely that normal behaviors can be represented by frequent patterns and suspicious behaviors by rare patterns. Rare itemsets contain items of high utility and may appear rarely in transactions or datasets. High utility frequent itemsets contribute the most to a predefined utility, objective function or performance metric [Erwin A. et.al (2007)]. For example, from a marketing strategy perspective, it is important to identify product combinations that have a significant impact on company's bottom line i.e. having the highest revenue generating power [Erwin A. et al (2007)].

There are several different approaches to discover rare association rules. The simplest way is to directly apply the apriori algorithm by simply setting the minimum threshold (minsup) to a low value. This leads to a combinatorial explosion in the number of patterns, most of them frequent with only a small number of them actually rare.

Shankar S. et.al (2009) presented a novel algorithm Fast Utility Mining (FUM), which finds all high utility itemsets within the given utility constraint threshold. The authors also suggest a novel method of generating different types of itemsets such as High Utility and High Frequency itemsets (HUHF), High Utility and Low Frequency itemsets (HULF), Low Utility and High Frequency itemsets (LUHF) and Low Utility and Low Frequency itemsets (LULF) using a combination of FUM and Fast Utility Frequent mining (FUFM) algorithms.

A different approach known as, Apriori Inverse was proposed by Koh and Rountree (2005), involves the modification of the apriori algorithm to use only the infrequent itemsets during rule generation. This simple change makes use of the maximum support measure, instead of the usual minimum support, to generate candidate itemsets, i.e., only items with a lower support than a given threshold are considered. Then rules are generated by an apriori approach.

Szathmary L. et al (2007) presented a novel algorithm for computing all rare itemsets by splitting the rare itemset mining task into two steps. The first step is the identification of the minimal rare itemsets. These itemsets jointly act as a minimal generation seed for the entire rare itemset family. In the second step, the minimal rare itemsets are processed in order to restore all rare itemsets.

Two algorithms were proposed for the first step: (i) a naive one that relies on an apriori-style enumeration, apriori-rare and (ii) an optimized method that limits the exploration to frequent generators only. The second task is solved by a straightforward procedure. Apriori-rare is a modification of the apriori algorithm used to mine frequent itemsets.

Apriori-rare generates a set of all minimal rare generators, also called MRM, that correspond to the itemsets usually pruned by the apriori algorithm when seeking for frequent itemsets. To retrieve all rare itemsets from minimal rare itemset (mRIs), a prototype algorithm called "A Rare Itemset Miner Algorithm (ARIMA)" was proposed. ARIMA generates the set of all rare itemsets, splits into two sets: the set of rare itemsets having a zero support and the set of rare itemsets with non-zero support. If an itemset is rare then any extension of that itemset will result a rare itemset.

A totally different approach to all these algorithms presented demands developing new algorithms to tackle these new challenges. Firstly consider apriori-inverse [Koh and Rountree (2005)], which can be seen as a more intricate variation of the traditional apriori algorithm. The main idea is that given a user-specified maximum support threshold, MaxSup and a derived MinAbsSup value, a rule X is rare if Sup(X)<MaxSup and Sup(X)>MinAbsSup.

Adda M. et.al (2007) proposed a framework to represent different categories of interesting patterns and then instantiate it to the specific case of rare patterns. A generic framework called *AfRIM* for *Apriori Rare itemset,* was presented to mine patterns based on the apriori approach. The generalized apriori framework was instantiated to mine rare itemsets. The resulting approach is apriori-like where the itemset lattice representing the itemset space in classical apriori approaches is traversed on a bottom-up manner, equivalent properties to the apriori exploration of frequent itemsets are provided to mine rare itemsets.

Jyothi and Vyas (2010) presented a new foundational approach to temporal weighted itemset utility mining where item utility values are allowed to be dynamic within a specified period of time, unlike traditional approaches where these values are static within those times. A Conceptual model was presented by Jyothi and Vyas (2011), which allows development of an efficient and applicable algorithm to real world data and captures real-life situations in fuzzy temporal weighted utility association rule mining.

The proposed algorithm is developed to derive out high-utility rare-itemsets, which may be useful in many real-life applications such as yielding high-profit in business. HURI algorithm considers the utility of itemsets other than the frequency of items in the transaction set. The utility of items is decided by considering factors such as profit, sale, temporal aspects, etc. of items. By using HURI, high-utility rare itemsets can be generated based on minimum threshold values and user preferences.

2. Proposed Algorithm

This section presents the proposed algorithm. They have first presented theoretical underpinnings of the proposed algorithm.

2.1 Definition (Utility Mining)

Let D (Table 1) be a given transaction database with a set of transactions $\{T_1, T_2, \ldots, T_n\}$ and a set of quantities of items $I=\{i_1, i_2, i_3, \ldots, i_m\}$ where each item i ε I has a set of utilities defined as $U=\{u_1, u_2, u_3, \ldots, u_k\}$ (Table 2). For example in transaction T_{29}, the quantities of items A001, B002, C003, D004, E005… are 1,3, 0,1,1… respectively.

Utility is a measure of how useful or profitable an itemset *X* is. The utility can be measured in terms of cost, profit or other expressions of user preferences. The utility of an itemset *X*, i.e., *u*(*X*), is the sum of the utilities of itemset X in all the transactions containing *X*. An itemset *X* is called a *high utility itemset* if and only if *u*(*X*) >= *min_utility*, where *min_utility* is a user-defined minimum utility threshold as defined by Yao H. et al (2006). Identification of the itemsets with high utilities is called as Utility Mining [Adda M. et al (2007)].

2.2 Definition (Utility Table)

A utility table UT (Table 2) is a table containing items and their corresponding utility values where each item i has some utility value u_j in U={u_1, u_2, u_3,... , u_k}for some k > 0. For example utility of item A001 is u(A001) = 4 in (Table 2).

2.3 Definition *(Internal Utility)*

The internal utility value of item i_p in a transaction T_q, denoted o(i_p, T_q) is the value of an item i_p in a transaction T_q (Table 2). The internal utility reflects the occurrence of the item in a transaction database. In Table 1, internal utility of item A0001 in transaction T1 is o(A001, T1) = 1, while internal utility of item A0001 in D is o(A001, D) = 21.

2.4 Definition (*External Utility*)

The external utility value of an item is a numerical value s(i_p) associated with an item i_p such that s(i_p)=u(i_p), where u is a utility function, a function relating specific values in a domain according to user preferences (Table 2). From Table 3, external utility of item A0001 is s(A0001) = u(A0001) = 4.

2.5 Definition (Item Utility)

The utility of an item i_p in a transaction T_q, denoted U(i_p, T_q) is product of o(i_p, T_q) and s(i_p), where o(i_p, T_q) is the internal utility value of i_p, s(i_p) is the external utility value of i_p(Table 3). For example, total utility of item A0001 is U(A001) = s(A001) * o(A001) = 4 * 21 = 84 (Table 2).

2.6 Definition (Transaction Utility)

The transaction utility value of a transaction, denoted as U(T_q) is the sum of utility values of all items in a transaction T_q (Table 1, Table 2). The transaction utility reflects the utility in a transaction database. From Table 1 and Table 3, the transaction utility of the transaction T1, U(T1) = U(A001)+U(B002)+U(C003)+U(D004)+ ... + U(T020)=39.

2.7 Definition (Frequent Itemset Mining)

Agrawal R. et al (1993) introduced the concept of frequent itemset mining. Frequent itemsets are the itemsets that occur frequently in the transaction data set. The goal of frequent itemset mining is to identify all the frequent itemsets in a transaction dataset. An itemset *X* = (*i*1, *i*2, ..., *ik*) with *k* items is referred to as *k*-itemset. The frequency of an itemset X is the probability of X occurring in a transaction T. A frequent itemset is the itemset having frequency support greater a minimum user specified threshold.

2.8 Definition (Rare Itemset Mining)

In many practical situations, the rare combinations of items in the itemset with high utilities provide very useful insights to the user. Rare itemsets are the itemsets that occur infrequently in the transaction data set. In most business applications, frequent itemsets may not generate much profit while rare itemsets may generate a very high profit.

Rare itemsets are very important and can be further promoted together because they possess high associations and can bring some acceptable profits [Lan et al 2011].

Given a user-specified maximum support threshold maxsup, and a generated minabssup value, we are interested in a rule X if sup(X) < maxsup and sup(X) > minabssup. Rules above maximum support are considered frequent rules, which are of no interest to us, whereas we consider rules appearing below the maximum support value. Rare rules are generated in the same manner as in apriori rule generation. Apriori-Inverse produces rare rules that do not consider any itemsets above maxsup.

By applying Apriori- Inverse algorithm [Koh and Rountree (2005)] on Transaction dataset described in Table 1 and by setting the value of maximum support threshold to 40%, the rare itemsets generated are listed in Table 3.

Apriori-Inverse concept is extended in High Utility Rare Itemset Mining (HURI) algorithm presented and generates high utility rare itemsets. Rare itemsets of users' interest or high utility rare itemsets fall below a maximum support

T_ ID	A 001	B 002	C 003	D 004	E 005	F 006	G 007	H 008	I 009	J 010	K 011	L 012	M 013	N 014	O 015	P 016	Q 017	R 018	S 019	T 020
T1	1	2	2	0	0	1	1	0	2	0	1	0	5	0	0	1	4	0	1	0
T2	1	0	1	1	1	0	0	0	0	3	1	1	0	4	0	1	0	3	0	1
T3	1	3	2	0	0	0	0	0	2	0	1	0	3	2	1	0	0	2	0	1
T4	0	0	0	0	1	3	0	0	0	4	1	1	0	1	0	1	0	1	1	0
T5	0	1	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	1	0	0
T6	0	2	0	0	0	0	0	1	0	0	0	0	0	1	0	1	3	0	1	0
T7	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1	1	5	1	1
T8	1	0	1	1	1	0	0	0	3	0	1	0	4	4	0	1	0	0	0	1
T9	0	0	1	0	2	4	0	2	0	0	0	1	0	0	1	1	0	4	1	1
T10	2	3	1	1	1	0	0	0	5	0	0	1	6	2	1	1	6	0	0	0
T11	1	1	0	0	0	1	0	0	0	3	1	0	0	0	1	0	0	3	0	1
T12	1	0	1	0	1	0	1	1	1	1	0	0	1	5	1	0	0	0	0	1
T13	0	2	0	1	0	0	0	1	3	1	1	1	0	0	1	1	1	1	1	0
T14	0	0	1	0	2	3	1	0	1	5	0	0	3	2	0	0	5	0	0	1
T15	1	1	1	0	1	0	0	0	1	1	1	1	0	0	1	1	1	2	0	1
T16	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
T17	0	0	0	0	0	0	0	0	2	1	1	1	0	4	0	1	0	2	0	1
T18	1	3	0	0	1	4	0	0	0	0	0	0	5	0	0	1	0	0	1	0
T19	0	0	0	1	2	0	0	1	0	0	0	1	0	2	0	1	1	1	0	1
T20	0	0	2	0	0	0	0	1	2	0	0	0	1	0	0	1	5	0	1	0
T21	2	0	1	0	0	3	0	1	0	2	0	1	1	1	0	0	0	3	0	1
T22	0	0	2	0	0	0	0	1	0	0	0	0	0	1	1	0	0	1	0	0
T23	0	0	0	0	2	1	1	0	1	0	0	1	1	0	0	0	1	2	1	1
T24	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	4	1	0	0
T25	2	2	1	1	1	0	1	0	1	1	2	1	4	1	0	0	1	1	0	0
T26	0	0	0	2	1	0	0	0	0	0	0	0	1	1	1	0	1	0	1	0
T27	1	0	0	0	0	0	1	0	1	0	0	0	1	5	0	0	2	5	0	1
T28	0	0	0	0	0	4	0	1	2	0	0	0	2	0	1	0	1	0	1	1
T29	1	3	0	1	1	2	1	0	1	2	1	0	0	1	0	0	2	2	0	0
T30	0	0	2	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	1
T31	2	1	0	1	1	0	1	0	1	0	0	0	2	1	0	1	0	2	0	1
T32	0	1	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0
T33	1	1	0	1	0	1	1	0	1	1	1	1	5	2	0	0	1	1	0	0
T34	0	1	0	0	0	0	0	1	1	1	0	0	1	0	0	1	0	0	0	0
T35	0	1	1	1	2	0	0	0	0	1	1	1	1	0	0	1	1	1	0	0

Table 1. Transaction Dataset D

value but above a user provided high utility threshold. Hence for example by setting high utility threshold as 45, the high utility rare itemsets generated are listed in Table 4.

In Apriori inverse algorithm, rare itemsets are generated by considering those itemsets which fall below maxsup value. But in HURI Algorithm, high utitlity rare itemsets are generated in two phases:-

- In first phase, rare itemsets are generated by considering those itemsets which have support value less than the maximum support threshold(using apriori-inverse concept).
- In second phase, by inputting the utility threshold value according to users' interest, rare itemsets having utility value greater than the minimum utility threshold are generated.

Both HURI and apriori inverse algorithm considers utility values of all items in transaction set in addition to frequency. But apriori inverse produces only rare itemsets whereas HURI produces high utility rare itemsets according to users' interest.

Algorithm HURI

Description: Finding High Utility Rare Itemsets of users interest

C_k: Candidate itemset of size k

Items	External Utility	Internal Utility	Total Utility
A 001	4	21	84
B 002	1	28	28
C 003	3	21	63
D 004	2	12	24
E 005	7	23	161
F 006	5	27	135
G 007	6	10	60
H 008	1	13	13
I 009	1	34	34
J 010	4	27	108
K 011	3	15	45
L 012	1	14	14
M 013	1	50	50
N 014	2	40	80
O 015	3	14	42
P 016	1	18	18
Q 017	1	42	42
R 018	1	44	44
S 019	1	11	11
T 020	0	17	0

Table 2. Item Utility Table

Rare itemsets	List of rare itemsets	Itemset Utility
1-itemset	{D004}	24
	{G007}	60
	{H008}	13
	{S019}	11
2-itemset	{D004,G007}	84
	{D0004,H008}	37
	{D004,S019}	35
	{G007,H008}	73
	{G007,S 019}	71
	{H008,S019}	24
3-itemset	{D004,G007,H008}	97
	{D004,G007,S019}	95
	{G0007,H0008,S0019}	84
	{D004,H008,S019}	48
4-itemset	{D004,G007, H0008,S019}	108

Table 3. Rare Itemset Table

High Utility Rare Itemsets	List of high utility rare itemsets	Utility
1-itemset	{G007}	60
2-itemset	{D004,G007}	84
	{G007,H008}	73
	{G007,S 019}	71
3-itemset	{D004,G007,H008}	97
	{D004,G007,S019}	95
	{G007,H008,S019}	84
4-itemset	{D004,G007, H008,S019}	108

Table 4. High Utility Rare Itemset Table

L_k: Rare itemset of size k

For each transaction t in database

begin

increment support for each item i present in *t*

End

L_1= {Rare 1-itemset with support less than user provided max_sup}

for(k= 1; L_k!=Ø; k++)

begin

C_{k+1}= candidates generated from L_k;

//loop to calculate total utility of each item

For each transaction t in database

begin

Calculate total quantity of each item i in t

Find total utility for item i using following formula:- u(i,t) = quantity[i] * user_provided_utility for I

End

//loop to find rare itemsets and their utility

For each transaction t in database

begin

increment the count of all candidates in C_{k+1} that are contained in *t*

L_{k+1} = candidates in C_{k+1} *less than* min_support

Add L_{k+1} to the Itemset_Utility table in database by calculating rare itemset utility using following formula:

Utility(R,t) = $\sum_{\text{for each individual item i in R}}$ (u(i,t));

End

//loop to find high utility rare itemset

For each itemset iset in rare itemset table R

begin

If (Utility(iset) > user_provided_threshold_for_high_utility_rare_itemset)

then iset is a rare_itemset that is of user interest i.e.high_utility_rare_itemset

else iset is a rare itemset but is not of user interest

End

Return high_utility_rare_itemsets

END

3. Comparative Results

The proposed High Utility Rare Itemset Mining (HURI) algorithm uses the concept of apriori inverse, for generating high utility rare itemsets of users' interest [Koh and Rountree (2005)]. Apriori Inverse and HURI algorithms were compared using a transactional database D (Table 1). Java as front end and MS Access as backend tool were used to evaluate HURI algorithm.

The execution results of both algorithms are:

- Number of rare itemsets generated.
- Total execution time taken for generation of rare itemsets.
- Comparison using different support threshold and data sizes.

In item utility table (Table 2), each item is assigned an external utility and internal utility is calculated from database D. The total time to generate rare itemsets using both Apriori Inverse and HURI is shown in Table 5. Figure 2 shows the execution time of both algorithms.

Number of rare itemsets generated using Apriori Inverse and HURI is shown in Table 6 and comparative graph is shown in Figure 3 respectively. A range of support thresholds was taken for comparative study. In both Apriori Inverse and HURI, number of rare itemsets increases as maximum support increases.

Both HURI and apriori inverse algorithm considers utility values of all items in transaction set in addition to frequency. But apriori inverse produces only rare itemsets

max_supp	Time Taken (secs.)	
	HURI	INV
10	0.004	0.006
20	0.009	0.011
30	0.014	0.015
40	0.031	0.031
50	13.325	25.484
60	26.941	29.054

Table 5. Execution time on database D

max_supp	No. of rare itemsets	
	HURI	INV
10	1	1
20	3	1
30	4	1
40	16	5
50	18	13
60	23	19

Table 6. Number of rare itemsets generated from database D

whereas HURI produces high utility rare itemsets according to users' interest. Figure 2 shows the effect on execution time caused by varying the support threshold. The results show that the proposed HURI algorithm yields more rare itemsets with less execution time as compared to apriori inverse.

Conclusion and Future Work

This paper proposes a new approach, HURI, for making business data mining more realistic and usable to business analyst. Marketers are interested in knowing how various marketing programs affect the discovery of subtle relationships. HURI provides user with the information of rare itemsets with high utilities. The high utility rare itemsets are generated according to the users' preference [Hu and Mojsilovic (2007). The outcome of HURI would enable the top management or business analyst in crucial decision-making such as providing credit facility, finalizing discount policy, analyzing consumers' buying behaviour, organizing shelf space, quality improvement in

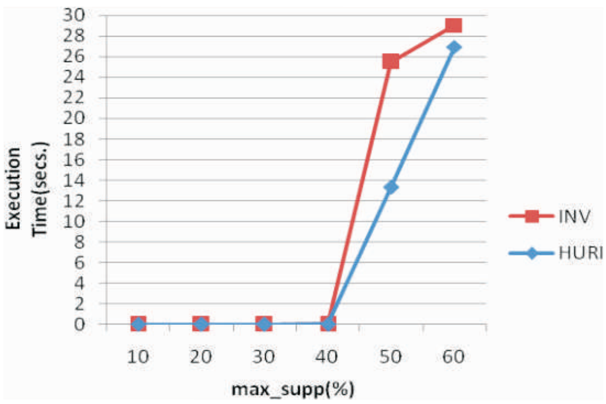

Figure 2. Execution time for Apriori-Inverse and HURI on Dataset D

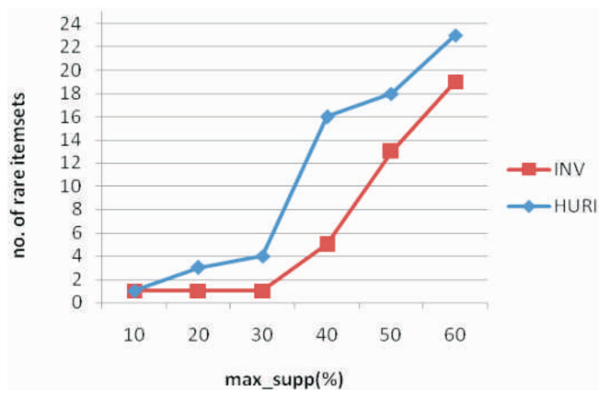

Figure 3. Number of rate itemsets generated from Apriori-Inverse and HURI

supermarket scenario.

HURI uses the concept of apriori inverse which produces only rare itemsets having support less than maximum support value where as HURI can produce high utility rare itemsets based on support threshold, utility threshold and users' interest. Hence HURI is said to be more beneficial on application to synthetic data set.

The future work includes the incorporation of temporal and fuzzy concept in HURI and using it for finding those rare items, which provide maximum profit to a transaction. HURI can also be used as a base for customer utility mining for classifying customers according to some criteria; for example, a retail business may need to identify valuable customers who are major contributors to a company's overall profit. The future work also includes finding the share of profitable transactions in the whole business, which may help in effective planning for retail marketing in Supermarket and online stream mining.

References

[1]. **Adda M., Wu L., and Feng Y. (2007).** "Rare Itemset Mining", *Sixth International conference on Machine Learning and Applications,* pp. 73- 80.

[2]. **Agrawal R., Imielinski T., and Swami A. (1993).** "Mining association rules between sets of items in large databases", *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data,* Washington, pp. 207–216.

[3]. **Erwin A., Gopalan R.P.and. and Achuthan N. R (2007).** "A Bottom-up Projection based Algorithm for mining high utility itemsets", in *Proceedings of 2nd Workshop on integrating AI and Data Mining(AIDM 2007)", Australia,* Conferences in Research and Practice in Information Technology (CRPIT),Vol. 84.

[4]. **Hu J., and Mojsilovic A. (2007).** "High-utility pattern mining: A method for discovery of high-utility item sets", *Pattern Recognition* 40 3317 – 3324.

[5]. **Koh Y.S., and Rountree N. (2005).** "Finding Sporadic Rules Using Apriori-Inverse", Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science, Volume 3518/2005, 153-168.

[6]. **Jyothi Pillai, and Vyas O.P. (2011).** "User centric approach to itemset utility mining in Market Basket Analysis", International Journal on Computer Science and Engineering (IJCSE) ISSN : 0975-3397, Vol. 3, No. 1 Jan, pp. 393-400.

[7]. **Jyothi Pillai, and Vyas O.P., (2010).** "Overview of Itemset Utility Mining and its Applications", *International Journal of Computer Applications (0975 – 8887)* Volume 5– No.11, August, Pp. 9-13.

[8]. **Jyothi Pillai, Vyas O.P., Sunita Soni, and Maybin Muyeba (2010).** "A Conceptual Approach to Temporal Weighted Itemset Utility Mining", *International Journal of Computer Applications (0975 - 8887)* Volume 1 – No. 28.

[9]. **Lan G.C., Hong T.P., and Tseng V.S. (2011).** "A Novel Algorithm for Mining Rare-Utility Itemsets in a Multi-Database Environment", *The 26th Workshop on Combinatorial Mathematics and Computation Theory,* pp. 293- 302.

[10]. **Liu, Y., Liao, W., and Choudhary, A. (2005).** "A Fast High Utility Itemsets Mining Algorithm", In *Proceedings of the Utility- Based Data Mining Workshop,* August.

[11]. **R.U. Kiran, and P.K. Reddy (2009).** "An improved multiple minimum support based approach to mine rare association rules", *IEEE symposium on Computational Intelligence and Data Mining.*

[12]. **Shankar S., Purusothoman T.P., Jayanthi S., and Babu N., (2009).** "A Fast Algorithm for Mining High Utility Itemsets", *Proceedings of IEEE International Advance Computing Conference (IACC),* Patiala, India, pp. 1459 – 1464.

[13]. **Szathmary L., Napoli A., and Valtchev P. (2007).** "Towards Rare Itemset Mining", *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence*, Volume 1, Pages: 305-312, ISSN:1082-3409, 0-7695-3015-X.

[14]. **Yao, H., and Hamilton, H,J. (2006).** "Mining Itemset Utilities from Transaction Databases", *Data and Knowledge Engineering*, 59(3): 603-626.

[15]. **Yao H., Hamilton H., and Geng L. (2006).** "A Unified Framework for Utility-Based Measures for Mining Itemsets", *In Proc. of the ACM Intel. Conf. on Utility-Based Data Mining Workshop (UBDM),* pp. 28-37.

[16]. Yao H., Hong, Hamilton H., and Butz, C. J. (2004). "A Foundational Approach to Mining Itemset Utilities from Databases", *Proceedings of the Third SIAM International Conference on Data Mining,* Orlando, Florida, pp. 482-486.

ABOUT THE AUTHORS

Mrs. Jyothi Pillai is currently working as an Associate Professor in Department of Computer Applications at Bhilai Institute of Technology, Durg (C.G.), India. She is a post-graduate from Barkatullah University, India. She is a Life member of Indian Society for Technical Education. She has a total teaching experience of 16½ years. She has a total of 15 Research papers published in National / International Journals / Conferences into her credit. Presently, she is pursuing Ph.D. from Pt. Ravi Shankar Shukla University, Raipur under the guidance of Dr. O.P.Vyas, IIIT, Allahabad.

Dr. O.P. Vyas is currently working as a Professor and Incharge Officer (Doctoral Research Section) in Indian Institute of Information Technology-Allahabad (Govt. of India's Center of Excellence in I.T.). He has done M.Tech (Computer Science) from IIT Kharagpur and has done Ph.D. work in joint collaboration with Technical University of Kaiserslautern (Germany) and I.I.T. Kharagpur. With more than 25 years of academic experience he has guided Four Scholars for the successful award of Ph.D. degree and has more than 80 research publications with two books to his credit. His current research interests are Linked Data Mining and Service