AN EFFICIENT ALGORITHM FOR MINING HIGH UTILITY RARE ITEMSETS OVER UNCERTAIN DATABASES

S. Zanzote Ninoria
Research Scholar, Dept. of Mathematics and Computer Science, R.D. University, Jabalpur, Madhya Pradesh State.

S. S. Thakur
Department of Applied Mathematics, Jabalpur Engineering College, Jabalpur, Madhya Pradesh State

ABSTRACT

In modern era, due to the broad applications of uncertain data, mining itemsets over uncertain databases has paying much more attention. Association Rule Mining (ARM) is a well known and most popular technique of Data Mining. It identifies itemsets from the dataset which appears frequently and generates association rules. This is the procedure which is followed by the traditional ARM it does not consider the utility of an itemsets. In real-world applications such as retail marketing, medical diagnosis, client segmentation etc., utility of itemsets is varied on various constraints such as based on cost, profit or revenue. Utility Mining intend to discover itemsets with their utilities by considering profit, quantity, cost or other user preferences.[22]High-utility itemset mining (HUIM) has thus emerged as an important research topic in data mining. But most HUIM algorithms only handle precise data, even though big data collected in real-life applications using experimental measurements or noisy sensors is often uncertain. High-Utility Rare Itemset (HURI) mining finds itemsets from a database which have their utility no less than a given minimum utility threshold and have their support less than a given frequency threshold. Identifying high-utility rare itemsets from a database can help in better business decision making by highlighting the rare itemsets which give high profits so that they can be marketed more to earn good profit. Koh and Rountree (2005) proposed a modified apriori inverse algorithm to generate rare itemsets of user interest. In this paper we propose an efficient algorithm named Mining High Utility Rare Itemsets over Uncertain Database (HURIU). This novel approach uses the concept of apriori inverse over uncertain databases. This paper will also give the new version or extension of the algorithm HURI proposed by Jyothi et al. The implementation of an algorithm for the analysis is done on JDK 6.1 and referred the sample dataset presented by Lan Y.et al,2015[15] for uncertain database.
1. INTRODUCTION

Data Mining entails a multistage process, which involves the input data in the form of pre-processed data and as an output gives patterns extraction as per the requirement of user. In recent times, with the increasing popularity of Internet of Things (IoT) and persistent computing, a large amount of uncertain data has been collected from different kinds of devices, such as RFID, sensors, and real-time monitoring systems[36], [37]. Data mining of uncertain data has become an vigorous region of research recently. A detailed survey of uncertain data mining techniques may be found in [3]. Association analysis is one of the most important data-mining models. An example of market-basket analysis a dataset has tuples which contains items that a customer has purchased in a transaction. The dataset is analyzed to discover associations among different items. An essential action in the process of mining is the extraction of frequent itemsets, or sets of items that co-occur in a major portion of the transactions. Moreover market-basket analysis, frequent itemsets mining is also a core section in other variations of association analysis, such as association-rule mining and sequential-pattern mining. Agarwal and Srikant in 1994 have proposed an algorithm named Apriori Algorithm [4]. It is a bottom up approach. It is most well-known algorithm to find the frequent patterns to give association between the itemsets. Apriori is designed to work on databases those containing transactions. Each transaction is considered as a set of items (a pattern). For a given user specified threshold, the Apriori algorithm identifies the item sets which occur frequently in the database. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time, and groups of candidates are checked against the database[43]. Many of the early frequent pattern mining algorithms were Apriori-based [4], which depend on a generate-and-test concept to mine frequent patterns from transaction databases of precise data by first generating candidates i.e. generated and then examining their actual support (i.e., occurrences) in the database. To improve algorithmic efficiency, some sampling [35] and progressive sampling techniques [34] have been develop. One of the major development was a tree-based frequent pattern mining algorithms [13] construct an extended prefix-tree structure (FP-tree) to capture the contents of the database and perform the mining process using a restricted test-only approach. These tree-based algorithms do not generate candidates, only test for support. As the popularity of wireless sensor networks is in demand, many real-life organic, medical or life science applications, huge volumes of data which struggling with uncertainty were collected. The existence or nonexistence of items in a dataset in these applications is uncertain partially due to natural measurement inaccuracies and sampling errors (e.g., in sensors or laboratory equipment), human reaction time, and intentional blurring of data to preserve secrecy (e.g., preserve patient’s privacy). Hence, mining uncertain data is more in demand[43]. In case of handling probabilistic databases of uncertain data, the existence or nonexistence of items in the databases is uncertain. The level of uncertainty can be expressed in terms of existential probability, which is coupled with each item in transactions in the probabilistic databases of uncertain data[43]. As an example of uncertain datasets the pattern recognition applications can be seen such as for a given satellite picture, image processing techniques can be applied to extract features that specify the existence or nonexistence of certain target objects (such as bunkers). Due to noises and limited resolution, the presence of a feature in a spatial area is often uncertain and expressed as a probability [9]. The Utility Mining is the
An Efficient Algorithm for Mining High Utility Rare Item sets over Uncertain Databases

concept which is the extension to the traditional frequent itemset mining. The traditional ARM approaches consider the utility of the items by its occurrences in the transaction set. The frequency of itemset isn't adequate to mirror the particular utility of associate itemset. For example, the sales manager might not have an interest in frequent itemsets that don't generate important profit. Recently, one of the most challenging data mining tasks is the mining of high utility itemsets efficiently. Identification of the itemsets with high utilities is named as Utility Mining. The utility is measured in terms of price, profit or alternative expressions of user preferences. For example, a computer system may be more profitable than a telephone in terms of profit [20]. Utility Mining emerges as an important topic in data mining field. Mining high utility itemsets from databases is the process of finding the itemsets with high profits. Here, itemset utility that means it’s level of interest, significance, or profitability to the user. Utility of items in a transaction database comprises with two major characteristic: 1) External Utility which can be defined as the importance of distinct items 2) Internal Utility which can be defined as the importance of items in transactions. Utility of an itemset can be expressed as the product of its external utility and its internal utility. An itemset is called a high utility itemset if its utility is no less than a user-specified minimum utility threshold; otherwise, it is called a low-utility itemset [38].

Utility of Itemset (U) = internal utility (i) * external utility (e)

Example

Let Table 1 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). Table 2 represents the unit profits associated with each itemset.

Table 1: Transaction Database

| Trans_id | Transaction | Transaction Utility |
|----------|-------------|---------------------|
| T1       | A(1),B(1),E(1),W(1) | 5                  |
| T2       | A(1),B(1),E(3)    | 8                  |
| T3       | A(1),B(1),F(2)    | 8                  |
| T4       | E(2),G(1)        | 5                  |
| T5       | A(1),B(1),F(3)    | 11                 |

Table 2: Unit Profits associated with items

| Item Name | A | B | E | F | G | W |
|-----------|---|---|---|---|---|---|
| Unit Profit | 1 | 1 | 2 | 3 | 1 | 1 |

The most popular extension of weighted itemsets mining is the High Utility Itemset Mining (HUIM) in which not only the weights are considered but also the purchase quantities also considered in transactions [8, 42,7,41,40,17,5]. In HUIM, weights can be considered as the unit profit of items [29,18,6,31,19,22]. The objective of HUIM is to discover all itemsets that have a utility higher than a given threshold in a database. The journey of HUIM has embarked in 2003[20]. In HUIs mining, items can have the quantity and relative importance. HUIs have numerous applications such as market basket analysis [42], [5], [38], website click stream [17], [44], cross-marketing in retail stores [17], [16], [11], [28] biomedical applications [10] and
mobile commerce applications [30], [39]. HUIs mining does not agree with downward closure property1 because the utility of an itemset may be smaller, equal or greater to the utility of its supersets (or subsets)[32]. Rare items are items that appear fewer frequently in a database. High Utility Itemsets can be frequent or rare. Even rare itemset can also be of high or low utility[21]. Apriori Inverse algorithm is proposed by Koh et al [14] which can be used in High Utility Rare Itemset Mining [24] algorithm. HURI finds high profitable rare itemsets according to user's perspective [24]. Apriori Inverse algorithm by Koh and Rountree in 2005 has been proposed. In this work the algorithm is used to generate rules that may contain items over the maximum support threshold called as perfectly sporadic rules. This algorithm is much faster than Apriori in finding perfectly rare itemsets, that are a subclass of rare itemsets containing itemsets whose all subsets are rare[14]. Szathmary et al. in 2007 has proposed the algorithm Apriori-Rare which finds all minimal rare itemsets. This algorithm finds out two sets of items. One is Maximal Frequent Itemset (MFI) and another is minimal Rare Itemset (mRI). An itemset is referred as a MFI if it is frequent but not all its supersets, similarly an itemset is referred as a mRI if it is rare but all its proper subsets are not. It also finds the generator which generates the Frequent Itemsets (FIs) [33].

Adda et al. described ARANIM algorithm for Apriori Rare and Non-Present Itemset Mining to mine rare and non present itemsets in [1][2]. In the proposed approach the technique is same like Apriori and the mining idea is that if the item-set lattice representing the itemset space presented in classical Apriori approaches is traversed in a bottom-up manner, than the equivalent properties for the Apriori exploration is provided to discover rare item-sets [1]. Apriori-rare, it is a alteration of the Apriori algorithm which is used to mine frequent itemsets. To extract all rare itemsets from minimal rare itemset (mRIs), a prototype algorithm called - A Rare Itemset Miner Algorithm (Arima) was proposed in [33]. Arima generates the set of all rare itemsets and splits those into two sets: the set of rare itemsets with a zero support and other with non-zero support. If an itemset is rare then any extension to that itemset will also result to a rare itemset [33]. Adda et al. proposed a framework in which is used to represent different categories of interesting Patterns. A common framework was presented to mine patterns based on the Apriori approach. The generalized Apriori framework was represented to mine rare itemsets. The Apriori algorithm, called AfiRIM for Apriori Rare itemset to mine rare itemsets was proposed which performs a level-wise search. The backward traversal method is used with a property that leads to prune out potentially non-rare itemsets in the mining process. This includes an anti-monotone property and a level wise exploration of the itemset space[2]. A Rare Itemset Mining Algorithm(Arima) presented by Adda et.al. in 2007[21]. Table 3 and Table 4 shows the Transaction Database with Total Utility and Utility of an itemset.

Table 3 : Transaction Database with Total Utility

| TID | Transaction | TU  |
|-----|-------------|-----|
| T1  | (C : 5) (D : 20) | 70  |
| T2  | (C : 1) (F : 40) | 42  |
| T3  | (A : 1) (B : 1) (C : 2) (G : 10) | 20  |
| T4  | (A : 1) (B : 1) (C : 2) | 10  |
| T5  | (A : 5) (C : 10) | 45  |

http://www.iaeme.com/IJCET/index.asp editor@iaeme.com

Electronic copy available at: https://ssrn.com/abstract=3538147
An Efficient Algorithm for Mining High Utility Rare Item sets over Uncertain Databases

Let us have a closer look at the basic definitions related to utility of items in a dataset.

$I = \{i_1, i_2, i_3, \ldots, i_m\}$ each item has external utility as $pr(i_p)$. Itemset $X$ of length $k$ is $X = \{i_1, i_2, i_3 \ldots, i_k\}$ where for $j \in 1 \ldots k$, $i_j \in I$. A Transaction database $D = \{T_1, T_2, \ldots, T_n\}$. Every item $i_p$ in a transaction $T_d$ has a quantity $q(i_p, T_d)$ associated with it.

Def.1: The utility of an item $i_p$ in a transaction $T_d$ is the product of the profit of the item and its quantity in the transaction i.e. $u(i_p, T_d) = q(i_p, T_d) \times pr(i_p)$.

Def.2: The utility of an itemset $X$ in a transaction $T_d$ is denoted as $u(X, T_d)$ and defined as $\sum_{x \subseteq T_d, i_p \in X} u(i_p, T_d)$.

Def.3: The utility of a transaction $T_d$ is denoted as $\sum_{i_p \in T_d} u(i_p, T_d)$.

Def.4: The utility of an itemset $X$ in database $D$ is denoted as $u(X)$ and defined as $\sum_{x \subseteq T_d, T_d \in D} u(X, T_d)$.

Def.5: An itemset is called a high utility itemset if its utility is no less than a user-specified minimum threshold denoted by $\text{min\_util}$.

Def.6: The support of an itemset $X$ denoted by $\text{sup}(X)$ is the number of transactions in database $D$ which contain itemset $X$.

Def.7: An itemset $X$ is called a rare itemset, if $\text{sup}(X) < \text{max\_sup\_threshold}$.

If $\text{max\_sup\_threshold}=2$, for the example shown in Table 3 Transaction Database and corresponding Utility Table 4, it uses the concept of ARIMA and generate the Rare Itemsets. Table 5 shows the Rare Itemsets in example database.

| Itemsets | List of Rare Itemsets |
|----------|-----------------------|
| 1-itemset| $\{D\}, \{F\}, \{H\}$ |
| 2-itemset| $\{AG\}, \{BH\}, \{CD\}, \{CF\}, \{CH\}, \{EG\}, \{EH\}$ |
| 3-itemset| $\{ABG\}, \{ACG\}, \{BCH\}, \{BEH\}, \{BEG\}, \{CEG\}, \{CEH\}$ |
| 4-itemset| $\{ABCG\}, \{BCEG\}, \{BCEH\}$ |

Table 4: Utility Table

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

Table 5: Rare Itemsets

Electronic copy available at: https://ssrn.com/abstract=3538147
High utility rare itemsets fall below a maximum support value but above a user provided high utility threshold \( \text{min\_util} \) [24][25]. Jyothi et al. [23] proposed influential work in an extension of High Utility work in terms of a two-phase algorithm which is used to find high utility rare itemsets from transaction databases. The two phase work accomplished as the rare itemsets are mined in the first phase and utility of rare itemsets are calculated in the second phase and high utility rare itemsets. Jyothi et al. [26] proposed an approach for handing profitable transactions along with high utility rare itemsets from a transaction database [27]. A UP-Rare Growth, which uses a UP-Tree data structure [38] to find high utility rare itemsets has been further proposed by Vikram et al in [12]. If \( \text{min\_util} = 30 \) and \( \text{max\_sup\_threshold}=2 \), Table 6 shows the High Utility Rare Itemsets in example database.

Table 6 : Rare High Utility Itemset

| Itemsets | List of Rare High Utility Itemsets |
|----------|----------------------------------|
| 1-itemset | \{D\},\{F\}                     |
| 2-itemset | \{CD\},\{CF\}                   |
| 3-itemset | \{Ø\}                           |
| 4-itemset | \{Ø\}                           |

In this paper, an efficient mining algorithm named High Utility Rare Itemsets Mining in uncertain data (HURIMU) is proposed to effectively discover the High Utility Rare Itemsets (HURIs). The rest of the paper is organized as follows: in Section 2 Comprehensively discusses Preliminaries. In Section 3 Existing approach for Mining High Utility Rare Itemsets is discussed. In Section 4 A Noval Efficient Approach for Mining High Utility Rare Itemsets over Uncertain Databases (HURIMU) Algorithm is proposed. In Section 5 experimental assessment on the uncertain dataset example given by Lan Y et al, (2015) are reported. Finally, we conclude the paper in Section 6.

2. PRELIMINARIES

In this section the basic definitions have discussed which are important to understand the problem statement.

Let \( I = \{i_1, i_2, ..., i_n\} \) be a set of distinct items. Each item \( i_p \) has a unit value \( u(i_p) \). We name a non-empty subset, \( X \), of \( I \) as an itemset. For brevity, we use \( X = x_1,x_2,...,x_n \) to denote itemset \( X = \{x_1, x_2, ...x_n\} \). \( X \) is a l - item set if it has \( l \) items. Given an uncertain transaction database \( D \), each transaction is denoted as \( < \text{tid}, Y > \), where \( \text{tid} \) is the transaction identifier, and \( Y = \{y_1(p_1), y_2(p_2), ..., y_m(p_m)\} \). \( Y \) contains \( m \) units. Each unit has an item \( y_i \) and probability \( p_i \), denoting the possibility of item \( y_i \) appearing in the \( \text{tid} \) tuple. A threshold, the minimum utility threshold, is defined as \( \mu \) [15]. For example the uncertain database is shown in Table 7 and the corresponding utility table is shown in Table 8. The minimum utility threshold is set at \( \mu = 25\% \) [15].

Table 7: An Uncertain Database

| T1  | A (0.2) C (0.3) E (0.2) |
| T2  | B (0.2) D (0.3)         |
| T3  | A (0.1) B (0.2) C (0.1) E (0.3) |
| T4  | C (0.2)                 |
| T5  | B (0.3) D (0.2) E (0.1) |
| T6  | A (0.2) C (0.2) D (0.5) |
An Efficient Algorithm for Mining High Utility Rare Item sets over Uncertain Databases

T7  A (0.1) B (0.1) D (0.4) E (0.1)  
T8  B (0.4) E (0.1)  
T9  A (0.3) C (0.3) D (0.2)  
T10 B (0.2) C (0.3) E (0.1)  

Table 8: An Utility Table

| TID | A  | B  | C  | D  | E  |
|-----|----|----|----|----|----|
| Utility | 4  | 1  | 12 | 6  | 15 |

Definition 1: 
Utility of an Item: The Utility of an item is the quantity of each item from the itemset multiplied by their unit profit. i.e utility of an item \( i_j \) in \( T_d \) can be defined as:

\[
U(i_j, T_d) = p(i_j, T_d) \times u
\]  

For example, the utility of an items (C) in \( T_3 \) is,

\[
U(C, T_3) = p(C, T_3) \times u(C) = 0.1 \times 12 = 1.2
\]

Definition 2:
Utility of an itemset \( \mathbf{X} \) in transaction \( T_d \): The Utility of an itemset \( \mathbf{X} \) in transaction \( T_d \) is denoted by \( U(\mathbf{X}, T_d) \), which can be defined as:

\[
U(\mathbf{X}, T_d) = \sum_{i_j \in \mathbf{X}, \mathbf{X} \subseteq T_d} U(i_j, T_d)
\]

For example, the utility of (BC) in \( T_3 \) is calculated as,

\[
U(BC, T_3) = U(B, T_3) + U(C, T_3) = p(B, T_3) \times U(B) + p(C, T_3) \times U(C) = (0.2 \times 1) + (0.1 \times 12) = 1.4
\]

Definition 3:
Utility of an itemset \( \mathbf{X} \) in Uncertain database \( D \): The Utility of an itemset \( \mathbf{X} \) in Uncertain database \( D \) is denoted as \( U(\mathbf{X}) \), which can be defined as:

\[
U(\mathbf{X}) = \sum_{\mathbf{X} \subseteq T_d, \mathbf{X} \subseteq D} U(\mathbf{X}, T_d)
\]

For example,

\[
U(C) = U(C, T_1) + U(C, T_3) + U(C, T_4) + U(C, T_5) + U(C, T_6) + U(C, T_10) = 3.6 + 1.2 + 2.4 + 3.6 + 3.6 = 16.8
\]

\[
U(BC) = U(BC, T_3) + U(BC, T_{10}) = 1.4 + 3.8 = 5.2
\]

Definition 4:
Transaction utility of transaction \( T_q \): The transaction utility of transaction \( T_q \) is denoted as \( U(T_q) \), which can be defined as:

\[
tu(T_q) = \sum_{j=1}^{m} U(i_j, T_d)
\]

In which \( m \) is the number of items in \( T_d \).
For example, \( TU(T_3) = U(A, T_3) + U(B, T_3) + U(C, T_3) + U(E, T_3) = 0.4 + 0.2 + 1.3 + 4.5 = 6.3 \)

Definition 5:
Total utility in \( D \): The total utility in \( D \) is the sum of all transaction utilities in \( D \) and is denoted as \( TU \), which can be defined as:

\[
TU = \sum_{T_d \in D} TU(T_d)
\]  

(5)

For example, the transaction utilities for \( T_1 \) to \( T_{10} \) are respectively calculated as \( TU(T_1) = 7.4, TU(T_2) = 1.3, TU(T_3) = 6.3, TU(T_4) = 2.4, TU(T_5) = 3, TU(T_6) = 6.2, TU(T_7) = 4.4, TU(T_8) = 1.9, TU(T_9) = 6.0, TU(T_{10}) = 5.3. \)

The total utility in \( D \) is the sum of all transaction utilities in \( D \), which is calculated as: \( TU = (7.4 + 1.3 + 6.3 + 2.4 + 3 + 6.2 + 4.4 + 1.9 + 6 + 5.3) = 44.2. \)

Definition 6:
High Utility Itemset (HUI) : An itemset \( X \) is defined as a high utility itemset (HUI) if its utility value \( U(X) \) is not less than the minimum utility count as:

\[
\sum_{X \subseteq T_d \land T_d \in D} U(X, T_d) = U(X) \geq \mu \times TU
\]

(6)

For example, suppose that the minimum utility threshold \( \mu \) is set at 25%. An item (A) is not considered as a HUI since \( U(BC) = 5.2 \), which is smaller than \( (0.25 \times 44.2) = 11.05. \) An itemset(C) is considered as a HUI in \( D \) since \( U(C) = 16.8 \), which is larger than the minimum utility count = 11.05.

Definition 7:
Transaction-weighted utility (TWU): The transaction-weighted utility (TWU) of an itemset \( X \) is the sum of all transaction utilities \( TU(T_d) \) containing an itemset \( X \), which is defined as:

\[
TWU(X) = \sum_{X \subseteq T_d \land T_d \in D} TU(T_d)
\]

(7)

Definition 8:
High transaction-weighted utilization itemset (HTWUI) : An itemset \( X \) is considered as a high transaction-weighted utilization itemset (HTWUI) if its

\[
TWU(X) \geq TU \times \mu
\]

Theorem 1 (Downward Closure Property of HTWUI): Let \( X^k \) and \( X^{k-1} \) be the HTWUI from uncertain databases, and \( X^{k-1} \subseteq X^k \). The \( TWU (X^{k-1}) \geq TWU (X^k) \).

Proof: Let \( X^{k-1} \) be a (k-1)-itemset and its superset k itemset is denoted as \( X^k \). Since \( X^{k-1} \subseteq X^k \), thus,

\[
TWU(X^k) = \sum_{X^k \subseteq T_d \land T_d \in D} TU(T_d)
\]

\[
\leq \sum_{X^{k-1} \subseteq T_d \land T_d \in D} TU(T_d)
\]

\[
= TWU(X^{k-1})
\]

(8)
Corollary 1: If an itemset $X^k$ is a HTWUI, every subset $X^{k-1}$ of $X^k$ is a HTWUI.

Corollary 2: If an itemset $X^k$ is not a HTWUI, no superset $X^{k+1}$ of $X^k$ is a HTWUI.

Theorem 2 (HUIs $\subseteq$ HTWUIs): The transaction-weighted downward closure (TWDC) property ensures that HUIs $\subseteq$ HTWUIs, which indicates that if an itemset is not a HTWUI, then none of its supersets will be HUIs.

Proof: $\forall X \subseteq D, X$ is an itemset; thus,

$$U(X) = \sum_{X \subseteq T_d \land T_d \in D} U(X, T_d)$$

$$\leq \sum_{X \subseteq T_d \land T_d \in D} TU(T_d)$$

$$= TWU(X)$$

(9)

Based on the above definitions, the problem statement of mining HURIU over uncertain databases had formulated as, for a given uncertain database $D$ with total utility is $TU$, the minimum utility threshold which is set as $\mu$. The problem of mining HURIU over uncertain databases is to mine Rare Itemset by considering those itemsets which have support value less than the max support threshold using Apriori Inverse Concept and the itemsets which are rare but having utility value greater than the min utility threshold value.

3. PROPOSED EFFICIENT APPROACH FOR MINING HIGH UTILITY RARE ITEMSET OVER UNCERTAIN DATABASES

This section will address the proposed efficient algorithm for mining high utility Rare itemsets over uncertain databases using Apriori-inverse [14] and UHUI-apriori [15] ideas. The idea has two distinct phases. We explain the working process of both the phases with the help of an example. Let us take an example of a sample uncertain dataset shown in Table 7 and the profit value associated with each item in Table 8. Let the minimum utility threshold $\mu = 25 \%$ and the maximum frequency support is 2. By applying Apriori-inverse algorithm [14] on uncertain dataset described in Table 7 with maximum frequency support is 2, the list of Rare itemsets generated are listed in Table 9. Apriori-inverse concept is used in HURIU (High Utility Rare Itemset over Uncertain Dataset) algorithm presented and generates HURI over Uncertain Dataset. In first phase, rare itemsets are extracted by considering those itemsets which have support value less than the maximum support threshold using apriori-inverse concept. In second phase the UHUI-apriori algorithm is extended using as an input the utility threshold value according to user’s interest, rare items having utility value greater than the minimum utility threshold are generated using the concept of UHUI-apriori algorithm. The proposed efficient algorithm HURIU Algorithm to mine HURI over uncertain databases is given below.

Algorithm : HURIU algorithm

Require : $D$, an uncertain database, utable, utility table, $\mu$ minimum utility threshold

Ensure: the set of high utility itemsets (HUIs)

1. for each $T_d$ in $D$ $\land$ $i_j$ in $T_d$ do
2. calculate $TWU(i_j)$
3. end for
4. for each $T_d$ in $D$ $\land$ $i_j$ in $T_d$ do
5. if $TWU(i_j) \geq TU \times \mu$ then
6. $HTWUI^1 \leftarrow i_j$
7. end if
8. end for
9. set $k \leftarrow 2$
10. set $X$ as ($k$) – itemset
11. while $HTWUI^{k-1} \neq \text{null}$ do
12. $C_k \leftarrow \text{apriori-inverse}(HTWUI^{k-1})$
13. for each $k$ – itemset $X$ in $C_k$ do
14. scan $D$ to find $TWU(X)$
15. if $TWU(X) \geq TU \times \mu$ then
16. $HTWUI^k \leftarrow X$
17. end if
18. end for
19. $k \leftarrow k + 1$
20. end while
21. $HTWUIs \leftarrow HTWUI^k$
22. for each $k$ – itemset $X$ in $HTWUIs$ do
23. scan $D$ to find $u(X)$
24. if $u(X) \geq TU \times \mu$ then
25. $HURI^k \leftarrow X$
26. end if
27. end for
28. $HURI_s \leftarrow HURI^k$

4. EXPERIMENTAL ASSESSMENT

This section illustrates the experimental results and extensive analysis of Efficient Algorithm for mining high utility rare itemset over uncertain database (HURIU). The HURIU over uncertain databases approach is implemented in Java (jdk 1.6), NetBeans IDE 7.0. These implementations have been done with sample example dataset with varying transactions, with varying Support and Confidence. For the experimental assessment we have referred the same example which is used in [15] by Y. Lan et al. The implementation has been done on sample example dataset with varying Support from lower threshold to higher thresholds. Sample Example Dataset is given below:
Here we have implemented the experiment for the different lower and higher threshold value as 25%, 35%, 40%, 50%, and 60% respectively. We have analyzed in this section the results obtained from the experimentation on HURIU. The experiments are conducted in order to assess the practical feasibility of using different support thresholds from Lower Thresholds to Higher Thresholds for High Utility Itemsets. This implementation has been compared with previous approach presented in [15]. For evaluating the performance of the Improved Mining High Utility Itemset over Uncertain Databases approach i.e. on Improved–UHUI, we have chosen three parameters such as, i) Run Time ii) Memory Utilization iii) Total Number of Rare Itemsets Generated.

4.1. Run Time

In computer science, the analysis of algorithms is for the determination of the amount of resources (such as time and storage) necessary to execute them. Here, comparison analysis between the proposed approaches HURIU with UHUI approach is done with various threshold values on the sample dataset. Results are shown in below graph.
It can be observed in the above figure that the proposed HURIU over Uncertain Databases approach is outperforms compare to the previous approach. The proposed approach is more efficient compare to the previous approach in terms of time utilization on lower and higher threshold both on the sample data example.

![Time Utilization Analysis](image)

It can also be observed that the time utilization for generating High Utility Itemsets and Rare Utility Itemsets generation can be varied with the threshold values. As it can be seen in the above graph that with lower threshold the time utilization is more than in the generation of High Utility Rare Itemset

### 4.2. Memory Utilization

Memory complexity is the size of work memory used by an algorithm. The memory usage of the algorithms was also compared when varying the thresholds from lower to higher, for sample datasets. We can observe easily that our proposed algorithm act differently for various threshold values. Results are shown in below Figure. The memory utilization is greater than the previous approach.
It can also be observed that the memory utilization for Rare Itemset Generation is always less than the memory utilization for High Utility Itemset Generation. It can be seen in the below graph.

4.3. Total No. of Items Generation
It can be clearly observed in the below graph that the Number of Rare Utility Itemsets increases as the threshold value get lower and it is decreases if the threshold value increased. The analysis can be seen in the below graph.
5. CONCLUSION AND FUTURE ENHANCEMENT

In this paper we have discussed an efficient approach for Mining High Utility Rare Itemsets over Uncertain Database using HURIU algorithm. Extensive experimental implementation has been performed to test the performance of this approach over our sample example dataset. It can be concluded that proposed approach HURIU can be suggested for better utilization of Time as it consumes less time for the generation of High Utility Rare Itemsets. It consumes more memory hence not recommended for better utilization of memory. It is also concluded that as the threshold value is increases the generation of HURI is decreases and with lower threshold values it increases the number of HURI generated.

**Table 9: Conclusion of Time and Memory Utilization**

| Sr.No. | Algorithms   | Support (Threshold %) | Run Time Utilization (in seconds) | Memory Utilizations (in bytes) |
|--------|--------------|------------------------|----------------------------------|------------------------------|
| 1      | UHUI Algorithm | 25%                    | 1.092                            | 0.783355                     |
|        |              | 35%                    | 1.306                            | 0.783294                     |
|        |              | 40%                    | 1.246                            | 0.783309                     |
|        |              | 50%                    | 1.156                            | 0.783294                     |
|        |              | 60%                    | 1.188                            | 0.783294                     |
| 2      | HURIU Algorithm | 25%                    | 0.36                             | 2.151176                     |
|        |              | 35%                    | 0.372                            | 2.106758                     |
|        |              | 40%                    | 0.359                            | 2.107735                     |
|        |              | 50%                    | 0.396                            | 2.107285                     |
|        |              | 60%                    | 0.352                            | 2.106361                     |
**Table 10:** Conclusion of Itemsets Generated

| Sr.No. | Type of Itemsets Generated | Support (Threshold %) | No. of Itemsets Generated |
|--------|---------------------------|-----------------------|---------------------------|
| 1      | HURI Itemsets             | 25%                   | 10                        |
|        |                           | 35%                   | 5                         |
|        |                           | 40%                   | 4                         |
|        |                           | 50%                   | 2                         |
|        |                           | 60%                   | 2                         |
| 2      | HUI Itemsets              | 25%                   | 15                        |
|        |                           | 35%                   | 10                        |
|        |                           | 40%                   | 9                         |
|        |                           | 50%                   | 5                         |
|        |                           | 60%                   | 3                         |

Hence as per the above conclusion tables we can conclude that in the proposed efficient approach using HURIU algorithms the time utilization is reduced drastically for both the higher and lower threshold values. It can also be concluded as per Table 9 & Table 10 that if the the threshold values are low the number of high utility rare itemsets will be more and as the threshold values are moving higher the generation of high utility rare itemsets will be less. Hence we strongly recommend the proposed approach for the extraction of High Utility Rare Itemset over Uncertain Dataset for better utilization of time.

**REFERENCES**

[1] Adda M., Wu L., Feng Y. Rare itemset mining. InMachine Learning and Applications. ICMLA 2007. Sixth International Conference, Dec 13, 2007, pp.73-80.

[2] Adda M., Wu L., White S., Feng Y. Pattern detection with rare item-set mining. arXiv preprint arXiv:1209.3089. 2012 Sep 14.

[3] Aggarwal C.C., Li Y. and Wang J. Frequent pattern mining with uncertain data. in SIGKDD 2009, pp.29-38.

[4] Agrawal R. and Srikant R. Fast algorithms for mining association rules. in VLDB 1994, pp.487–499.

[5] Ahmed CF., Tanbeer SK., Jeong BS. and Lee YK. Efficient tree structures for high utility pattern mining in incremental databases. IEEE Trans.on Knowl. and Data Eng. 21(12), Dec. 2009, pp. 1708–1721.

[6] Ahmed CF., Tanbeer SK., Jeong BS. and Lee YK. HUCPrune: An Efficient Candidate Pruning Technique to mine high utility patterns. Appl Intell, 2011, PP. 181–198.

[7] Bhattacharya S. and Dubey D. High Utility Itemset Mining. International Journal of Emerging Technology and Advanced Engineering, 2, 8 August 2012.

[8] Chan R., Yang Q. and Shen Y. D. Mining high utility itemsets. in Third IEEE International Conference on Data Mining, Nov 2003, pp. 19–26.

[9] Chui C.K., Kao B., and Hung E. Mining frequent itemsets from uncertain data. in PAKDD 2007, pp. 47–58.

[10] Dai X., Yiu M.L., Mamoulis N., Tao Y. and Vaitis M. Probabilistic spatial queries on existentially uncertain data. In: SSTD Lecture Notes in Computer Science, Springer, 3633, 2005, pp. 400–417.
S. Zanzote Ninoria and S. S. Thakur

[11] Erwin A., Gopalan R. P., and Achuthan N. R. Efficient Mining of High Utility Itemsets from Large Datasets. Springer Berlin Heidelberg, 2008, pp. 554–561.

[12] Goyal V., Dawar S. and Sureka A. High utility rare itemset mining over transaction databases. In International Workshop on Databases in Networked Information Systems, Springer Cham, Mar 23 2015, pp. 27-40.

[13] Han J., Pei J. & Yin Y. Mining frequent patterns without candidate generation. In ACM SIGMOD 2000, pp. 1–12.

[14] Koh YS. and Rountree N. Finding sporadic rules using apriori-inverse. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, Berlin, Heidelberg, May 18 2005, pp. 97-106.

[15] Lan Y., Wang Y., Wang Ya, Shengwei Yi S. and Yu D. Mining High Utility Itemsets over Uncertain Databases. 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, IEEE, Xi'an, 2015, pp. 235-238.

[16] Lee D., Park S.H., and Moon S. Utility-based association rule mining: A marketing solution for cross-selling. Expert Systems with Applications, 40 (7), 2013, pp. 2715–2725.

[17] Li H.F., Huang H.Y., Cheng Chen Y., Liu Y. and Lee S. Fast and memory efficient mining of high utility itemsets in data streams. In Eighth International Conference of Data Mining, Dec 2008, pp. 881–886.

[18] Liu J.P., Wang Y. and F.D. Incremental Mining algorithm Pre-FP in Association Rule Based on FP-tree. Networking and Distributed Computing International Conference, 2010, pp. 199-203.

[19] Liu M. and Qu J. Mining High Utility Itemsets without Candidate Generation. CIKM’12, USA, ACM, October 29-November 2, 2012.

[20] Ninoria SZ. and Thakur SS. A Survey on High Utility Itemsets Mining. International Journal of Computer Applications, 975(8887), Oct 2017, pp. 44-50.

[21] Ninoria SZ. and Thakur SS. Review On Rare Itemset Mining. International Journal of Computer Sciences and Engineering, NCRTI, Feb 2019, submitted.

[22] Philippe F.V., Cheng W.W., Souleymane Z. and Vincent S.T. FHM: Faster High-Utility Itemset Mining using Estimated Utility Co-occurrence Pruning, Proc. 21st International Symposium on methodologies for Intelligent Systems (ISMIS 2014), Springer, LNAI, 2014, pp. 83-92.

[23] Pillai J., Vyas OP., and Muyeba M. Huri—a novel algorithm for mining high utility rare itemsets. In Advances in Computing and Information Technology, Springer, Berlin, Heidelberg, 2013, pp. 531-540.

[24] Pillai J. and Vyas OP. High Utility Rare Item Set Mining (HURI): An Approach for Extracting High Utility Rare Item Sets. Journal on Future Engineering and Technology, 7(1), 1 Aug 1 2011.

[25] Pillai J. and Vyas OP. Overview of itemset utility mining and its applications. International Journal of Computer Applications, 115, Aug 2010, pp. 9-13.

[26] Pillai J. and Vyas OP. Transaction profitability using huri algorithm [tphuri]. International Journal of Business Information Systems. 2013 Nov; 2(1).

[27] Pillai J. and Vyas OP. User centric approach to itemset utility mining in Market Basket Analysis. International Journal on Computer Science and Engineering, 2011 Jan, 3(1):393-400.

[28] Yen S.J. and Lee Y.S. Mining High Utility Quantitative Association Rules. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 283–292.

[29] Shankar S., Purusothaman T.P., Jayanthi S. and Babu N. A fast algorithm for mining high utility itemsets. In: Proceedings of IEEE International Advance Computing Conference (IACC 2009), Patiala, India, 2009, pp. 1459-1464.

http://www.iaeme.com/IJCET/index.asp

editor@iaeme.com
An Efficient Algorithm for Mining High Utility Rare Item sets over Uncertain Databases

[30] Shie B.E., Hsiao H.F., Tseng V.S., and Yu P. S. Mining High Utility Mobile Sequential Patterns in Mobile Commerce Environments. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 224–238.

[31] Shih S.C., Cheng K. Huang, Zhe-Min Lin. New and efficient knowledge discovery of partial periodic patterns with multiple minimum supports The Journal of Systems and Software, ELSEVIER, 84, 2011.1638–1651.

[32] Singh K., Singh S.S., Kumar A., Shakya HKB. CHN: an efficient algorithm for mining closed high utility itemsets with negative utility. IEEE Transactions on Knowledge and Data Engineering. Nov 20 2018.

[33] Szathmary L., Napoli A. and Valtchev P. Towards rare itemset mining. InTools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on 2007 Oct 29, pp. 305-312. IEEE.

[34] Thakur SS and Ninoria SZ. An Improved Progressive Sampling based Approach for Association Rule Mining. International Journal of Computer Applications. 2017;165(7).

[35] Toivonen H. Sampling large databases for association rules. InVLDB , Sep 3, 196, Vol. 96, pp. 134-145.

[36] Tong Y., Zhang X., and Chen L. Tracking frequent items over distributed probabilistic data. 2015, pp. 1-26.

[37] Tong Y., Chen L., and She J. Mining frequent itemsets in correlated uncertain databases. Journal of Computer Science and Technology, 30,(4),2015, pp. 696–712.

[38] Tseng V. S., Wu C.W., Shie B.E. and Yu P.S. UP-Growth: an efficient algorithm for high utility itemset mining. InProceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2010 Jul 25, pp. 253-262.

[39] Tseng V. S., Shie B.E., Wu C.W. and Yu P.S. Efficient algorithms for mining high utility itemsets from transactional databases, IEEE Trans. on Knowl. and Data Eng., 25(8), Aug. 2013, pp. 1772–1786.

[40] Yao H., Hamilton H.J and Geng L.A unified framework for utility based measures for mining itemsets, in: proceedings of the ACM international conference on utility-based Data Mining Workshop (UBDM), 2007, pp. 28-37.

[41] Yao H., Hamilton H.J. Mining itemset utilities from transaction databases. in Data and Knowledge Engineering 59, 2006,pp.603-626.

[42] Yao H., Hamilton J., Butz C.J. A foundation approach to mining itemset utilities from databases. in: Proceedings of the Third SIAM International Conference on Data Mining, Orlando, Florida, 2004, pp.482-486.

[43] Yue L. Review of algorithm for mining frequent patterns from uncertain data. International Journal of Computer Science and Network Security (IJCNS), 15(6), Jun 1,2015.

[44] Zihayat M. and An A. Mining top-k high utility patterns over data streams. Information Sciences,285, 2014, pp.138 – 161.