Extraction of Closed High-Utility Itemsets and Generatorsbased on Multiple Minimum Support and Utility

G. Srilatha[1], N Subhash Chandra[2],

[1]Research Scholar, Asst. Professor, Jyothishmathi Institute of Technology & Science, Nustulapr, Karimnagar.
[2]Professor, Department of Computer Science and Engineering, CVR College of Engineering, Vastu Nagar, Mangalpalli(V), Ibrahimpatnam (M), RangaReddy (D).
([*]Corresponding author's e-mail: gajula.srilatha.2017@gmail.com)
Received: xxx, Revised: xxx, Accepted: xxx

Abstract

Extracting high utility Itemsets from transactional data samplesdenotes to the production of high utility Itemsets that generates higher profit. Mining of Closed High-Utility Itemsets (CHUIs) functions like a dense and lossless depiction of High Utility Itemsets (HUIs). In addition, CHUIs and its generators are also beneficial in the recommendation and analytical systems. Even though existing approaches have proposed efficient methodologies for the extraction of CHUI and generators, those techniques pre-dominantly used single utility threshold values and single support values. Thus, in this methodology, we suggested an improved association rule miningapproach using multiple minimum support values and utility values for the extractionof CHUI and High Utility Generator (HUG). The extraction is performed through the construction of Lattice for the generated HUIs swiftly to minimize the consumption period where the size of the exploring domain is very large. The performance of the proposed methodology is tested using three available datasets such as foodmart, retails, and chess. The suggested approach has lesser runtime and memory usage exhibited by experimental outcomes when matched with the prevailing approaches.

Keywords*:*Frequent Itemset Mining, High-Utility Itemset Mining, Multiple Minimum Utility, Multiple Minimum Support, Lattice, Generators.

Introduction

The exploration of valuable and interestingdatabased on the domain construction and application field is accomplished via numerous data mining techniquesthat are hidden in databases. The Association Rule Mining (ARM) [1] is the oneconsidered the topic widely by many researchers [11, 12, 13] in literature. Extracting an ‘‘attracting’’ Itemsetfrom huge transactional data samples is becoming one of the significantjobs in the present data miningresearch domain wherethe association guidelines are extracted in two phases. The Itemsets in the initial phase thatrecurrently co-occurred in the transactions are extracted. In the subsequent phase, the guidelines are obtained from constructed recurrentItemsets. However, the generated association rules do not considerItemsets that are lesser frequent and highercost-effective, i.e. having more margins. Maximum business appliancesoftenclaimfor betterfeasibility in the determination of item utilities likemargins, revenues to generate the fascinating guidelines.

Depending on the support and confidence model, traditional ARM delivers the objective measure forthe guidelinesthat are attracting users. Nevertheless, it does not offer any addedinformation to the superior,excluding the actions that replicate the numericalassociation amongst items. Furthermore, it does not replicateits semantic consequencein the direction of mining information. Alternatively, a support confidence prototype might not measure the worth of a rule compliant with the customer's goal(for instance, profit).The semanticmeasure of anyItemset is describedin terms of its utility values which are characteristicallyrelatedto transactional items, where an individual merely would be attracted to an Itemset if it pleases a specified utility limitation.

In the Traditional Association rule mining approach, whole items are specified withasimilarsignificance considering the survivalof items inthe transactional data without observing the utility of Itemsets. For mining of HUI, many investigations are doneto define the utilities of entire items in the data sample by users. HUI is defined as an Itemset bearing utility not lesser compared to a minimal utility threshold and the issue of extractingHUIis widelyconsidered to be complex compared to the problem of extracting Frequent Itemsets. The downward-closure framework in FIM denotes that support of any itemis anti-monotonic, and therefore supersets of rareItemset are rare and the subsets of a recurrentItemset are recurrent. The framework is dominantrelating tocroppingof exploring the domain. The utility of an Itemset is not monotonic nor anti-monotonic,in HUI Mining; hence higher utility Itemsetmight havea superset or subset with lesser, equivalent, or upper utility. Thus, procedures to cropexploring the domain constructed in FIM could not be straightforwardly employed in HUIMining, and hence numerous current approaches are concentrating on extracting HUI, particularly on candidate removal.

Association rule mining,in general, employs justsingle minimum support for the complete transactional data sample and considers whole items with the equal importance. Presume that tacitly entire items inthe dataset haveidentical characteristics. However,in practical life, every item might have diverse different characteristics, rates, and significance and hence it is essential to different deliberate characteristics. For addressing this issue, extraction of association rule with multiple minimalsupports [2, 5] along with the importance of items [3, 4, 6-8, 9] considering multiple utilitieshas beensuggested.ARM having multiple minimal supports finds complete significantguidelines, including occasionally appearing; however significant guidelines, through employing diverse minimum supports with regard to every item.In ARM, instead of employing single minimal support, it is essential to fix the minimal support less for discovering infrequent association rules. However, it mightlead to severalguidelines along with numerous pointless rules with an increase in the search space. Therefore, in this paper,aframework is suggestedto consider features of real-world datasets, the significance ofevery item in transactions, in the multiple minimum supports a model with multiple utility threshold values for every item.An approach forthe construction of Lattice from HUI using multiple minimum utilities and support value is introduced to efficiently discovering CHUI and HUG with multiple minimal supports and utility through the lattice tree structure.

Literature Survey

One of the initialstudies on HUI is the two-phase methodology [29]. This approach extracts HUIs in two stages. In the initial stage, the approach employs the notion of Transaction Weight-Utility(TWU) to extractcompletely higher TWU-Itemsets. Consequently, in the subsequent step, the approachevaluates genuine utilities and defines HUIs. The methodology performs poorly for huge data samples since it tracks a level-basedcandidate construction and testingapproach. Some additional level basedextracting approaches in research includes UMining and UMining_H [30], FUM and DCG+ [31], and GPA [32]. To addressrestrictions of the level-basedmethods, numerous tree aided techniques are suggested in the literature. Certain distinguishedtree-based HUI methods comprise of IHUP [33], HUC-Prune [34], UP-Growth [35],and UP-Growth+ [36].

For capably mining HUIs, HUI-Miner [37], FHM [38], and HUP-Miner [39] employ vertical database demonstrations. These approachesemploy utility-list data structure for storing Itemsetdata in the course of the mining. Certain, common pruning policies used in these methodologies contain TWU [37], U-Prune [37], EUCS-Prune [38], and LA-Prune [39]. These approaches are recognized to be the utmost effective approaches in literature [38] as extract HUIs in a unique step deprived of producing candidates. D2HUP [37] is the other current utility list aided approach that straightforwardly determines HUIs deprived of producing candidates. The hyperlinked-utility list structure known as CAUL is presented by authors for capablestorageof Itemsets. This approach is in the order of magnitude quicker compared to UP-Growth [35].

The utmosttopical and effective technique familiarized in literature for extracting HUIs is EFIM [40], and ituses a horizontal data depiction for loadingItemset. For competently mining HUIs, EFIMuses the ideas of transaction combination, dataset projection, and quick utility evaluation. LU-Prune and SU-Prune are the two new pruning strategies that were presented. The authors validated that their technique in the order of two to three magnitudes quickercompared to theexisting approaches in the literature. IMHUP [41] method employed an indexed-based utility-list for swift extraction of HUIs and this approach neither stores the transaction identifiers nor accomplishes expensivethe intersection amongst transaction-list. Experimental results showed that this approach performs 2-12 times faster compared to HUI-Miner and FHM. Nevertheless, the EFIM approach is proved to be 2 – 3 times faster compared to HUI-Miner and FHM.In [42], a hybrid approach is suggested that merges tree aided (UP-Growth+) and utility list aided the FHM technique. A heuristic approach with dynamism switching is given from a tree aided to the utility list-basedtechnique. UFH was matched with EFIM and proved to perform well for sparse data samples. But, when matched with EFIM on sparse standard datasets [40], D2HUP and HUP-Miner approach performed better.

Earlier, the issue of mining high utility Itemset was projected formally, andthe wideinvestigation is being present for frequent miningItemsets. Apriori [1] was the initial familiarFIM algorithm which depends on a property called downward-closure property [1]. A more efficient frequent Itemsets mining algorithm named Fp-Growth [22] was then suggested. Fp-Growth uses a tree-like data structure, and it does not require to generate candidates to mine frequent Itemsets. The rest of the FIM approaches are either depends on Apriori or Fp-Growth. Considering the importance of items to the user, weighted association rule mining [23] was proposed. Since the proposal of a weighted association rule, a lot of techniques have been proposed by researchers. By considering the non-binary transaction of items, utility mining [24-28] was then proposed and attained a significant research subject in data mining.

In [18], two-phase algorithmscontaining two mining phasesare suggested. So as to powerfully mine high utility Itemsets, in [19], projected IHUP which utilizes a tree-like data structure. Some other widely studied high utility Itemset mining algorithms are HUP-tree [16] by Lin et al. and UP-growth and UP-growth+ in [17].The MHU-Growth [21] for extracting higher utility Itemsets with multiple minimal support was first proposed by Ryanga et al. The HUIM-MMU [20] for extracting high utility Itemsets with multiple minimum utility thresholds was then suggested. Our study intends to remove the fundamental research gap between MHU-Growth and HUIM-MMU and use multiple minimum support and multiple minimum utility thresholds to professionally discover the entire high utility Itemsets.The MHU-Growth approach [15]elongates CFP-Growth, to extract high utility frequent Itemsets having multiple minimumsupport thresholds. In [14], the HUIM-MMU approach for determining HUIs with multiple minimum utility thresholds is presented. To prune un-needed Itemsets for advancing the discovery of HUIs, twoenhanced TID-index and EUCP techniques are projected.

Preliminaries

Few basic preliminaries related to the extraction of association rules are discussed in this section. Here $I = \{i_1, i_2, i_3, \ldots, i_m\}$ befixedgroup of items, with every item i_ℓ, $1 \le \ell \le m$, having an exterior utility p_ℓ, $1 \le \ell \le m$ in utility table. The subset $X \subseteq I$ is known as an Itemset if Xcomprises of$k$dissimilar items $\{i_1, i_2, i_3, \ldots, i_k\}$, here$i_l \in I$, $1 \le \ell \le k$, known as k-Itemset. Let D be a task related data sample consisting of support, utility and transactionaltable $T = \{t_1, t_2, t_3, \ldots, t_n\}$, comprising of a collection of n transactions, where every transaction $t_d \subseteq I$, $1 \le d \le n$, in the data samplebe accompaniedby a single identifier, such as t_{id}. In each transaction t_d, $1 \le d \le n$, every item i_ℓ, $1 \le \ell \le m$ have anon-negative quantity known as $q(i_\ell, t_d)$ that signifiesprocuredsizedefined as an interior utility of item i_ℓ in transaction t_d.

Definition 1. The utility of any item $i_l$in transaction t_d is indicatedwith$u(i_\ell, t_d)$, and specified through the product of internal$q(i_\ell, t_d)$, and external utility $p_l$such as$u(i_\ell, t_d) = p_l \times q(i_\ell, t_d)$. An instance of the transactional data sample isgiven in Table 1.

Definition 2. The utility of anyItemset$X$enclosed ina transactiont_d, indicated as$u(X, t_d)$and specified through the summation ofthe utility of each item of Xint_d. Alternatively, $u(i_l, t_d) = Pi_l \in X \wedge X \subseteq td\ u(i\ell, td)$.

Definition 3. The utility of an Itemset$X$ in $D$is referred with $u(X)$and given by the summation of utilities of X in the entire transactions including Xin$D$, such as,

$$u(X) = PX \subseteq t_d \wedge t_d \in D \qquad (1)$$

$$u(X, t_d) = PX \subseteq t_d \wedge t_d \in D\ Pi_\ell \in X\ u(i_\ell, t_d) \qquad (2)$$

The group of transactionscomprising an Itemset$X$, in database$D$is known as the projected database of Itemset$X$ and it is referred to as DX.

Definition 4. An Itemset$X$ is known as high utility Itemset if the utility of X has at least the individual defined minimalutility threshold, min_{util}. Or else, it is known as low utility Itemset. Consider H to be a whole group of high utilityItemsets. Further,

$$H = \{X | X \in F, u(X) \geq \min_{\text{util}}\} \qquad (3)$$

Definition 5. The local utility of item x_i in Itemset$X$, referred as $luv(x_i, X)$and given by the summation of utility values of items x_i in the entire transactions comprising X, such as,

$$luv(x_i, X) = PX \subseteq t_d \wedge t_d \in D \; u(x_i, t_d) \qquad (4)$$

Definition 6. The local utility of an Itemset$X$ in the other Itemset$Y$ such that $X \subseteq Y$, referred as $luv(X, Y)$, is thesummation of local utility measure values for every item $x_i \in X$ in Itemset$Y$ that is denoted as

$$luv(X, Y) = Px_i \in X \subseteq Y \; luv\,(x_i, Y) \qquad (5)$$

To evaluate the local utility value of anyItemset$X$in the other Itemset$Y$, a utility unit array is essential to attach to every HUI.

Definition 7. The utility unit array of anyItemset$I = \{i_1, i_2, i_3, \dots, i_m\}$is represented by $U(X) = \{u_1, u_2, u_3, \dots, u_k\}$, where every $u_\ell$is $luv(i_\ell, X), 1 \leq \ell \leq k$.

Property 1. Forgiven Itemset$X$along with its utility unit array $U(X)$, the utility of X is specified as$u(X) = Px_i \in X \; luv(x_i X)$.

Definition 8. An Itemset$Y$is known as the closure of Itemset$X$if there existno other higher supersets of Xcompared to $Y$where $(X) = sup\,(Y)$, referred to as $\gamma(X)$. An Itemset$X$is the high utility closed Itemset if $X = \gamma(X)$ and $u\,(X) \geq \min_{util}$.

Definition 9. An Itemset$X$is known as HUI Generator if it has the high utility Itemset, and there is no other subset $Z$of Xsuch that $sup\,(X) = sup\,(Z)$.

Definition 10. A high utility association rule $R$is an associationamongst two HUIX, $Y \subseteq I$of the form $X \rightarrow Y$, utility confidence of rule R, referred to as $uconf(R)$, is given as

$$(R) = lu\,v\,(X, XY)\,u\,(X) \qquad (6)$$

$R: X \rightarrow Y$is called a high utility association rule if $uconf(R)$is higher or equivalent to minimal utility confidence threshold defined by a user.

Proposed Multiple Utility and Support based Association Rule Mining Using HUIL

A novel approach is suggested in this section to extract the associations' rules using the Utility-Support Framework of the Item sets dissimilar to the Support-Confidence Framework. Apart from HUI, the Closed High-Utility Itemsets (CHUI) along with their High-Utility Generators (HUG), are being extracted. Usually, most of the existing approaches obtained CHUI and HUG using a single utility threshold and minimum support value. The most significant and interesting part is that, instead of single threshold values, in this approach, multiple minimum utility threshold and multiple minimum support values are employed to extract the association rules from large databases. For this purpose, the proposed approach is segregated into three different phases. They are:

- Determination of Least Minimum Utility (LMU) and Least Minimum Support (LMS) values
- Construction of Lattices using mined HUI known as High-Utility Itemset Lattice (HUIL) structure
- Extraction of CHUI and HUG from constructed HUIL

Determination of Least Minimum Utility and Least Minimum Support values

An ARM with multiple minimal supports and utility discovers the entire significantguidelines, comprisinghardly everhappened but important rules via applying different minimum supports and minimum utility values with respect to every item. Every item has its individualdistinctive minimal utility threshold and support value compared to employing a single minimal utility threshold support values for entire items.

<u>Least Minimal Item Support</u> of an item $i_p$is given as the minimum support threshold of i_p and denotedas $MIS(i_p)$.Minimum support of an Itemset given multiple minimum support values for each item,$X = \{i_1, i_2, \ldots, i_k\}$ refers to the least Minimum ItemsetSupport value ofitems in X, and it is defined as min$[MIS(i_1), MIS(i_2), \ldots, MIS(i_k)]$, where $i_p \in X$ and $1 < p < k$.

<u>Least Minimal Utility Threshold</u> of an item i_j in a data, sample D is stated as$mu(i_j)$. A structure known as MMU-table indicates the user-specified minimum utility thresholdof each item in D and is defined as$MMU - table = \{mu(i_1), mu(i_2), \ldots, mu(i_m)\}$. The minimum utility threshold of a k-Itemset$X = \{i_1, i_2, \ldots, i_k\}$ in D is denoted as $MIU(X)$ and defined as the smallest mu value for items in X, that is: $MIU(X) = min\{mu(i_j)|i_j\ 2\ X, 1\ \ j\ k\}$.

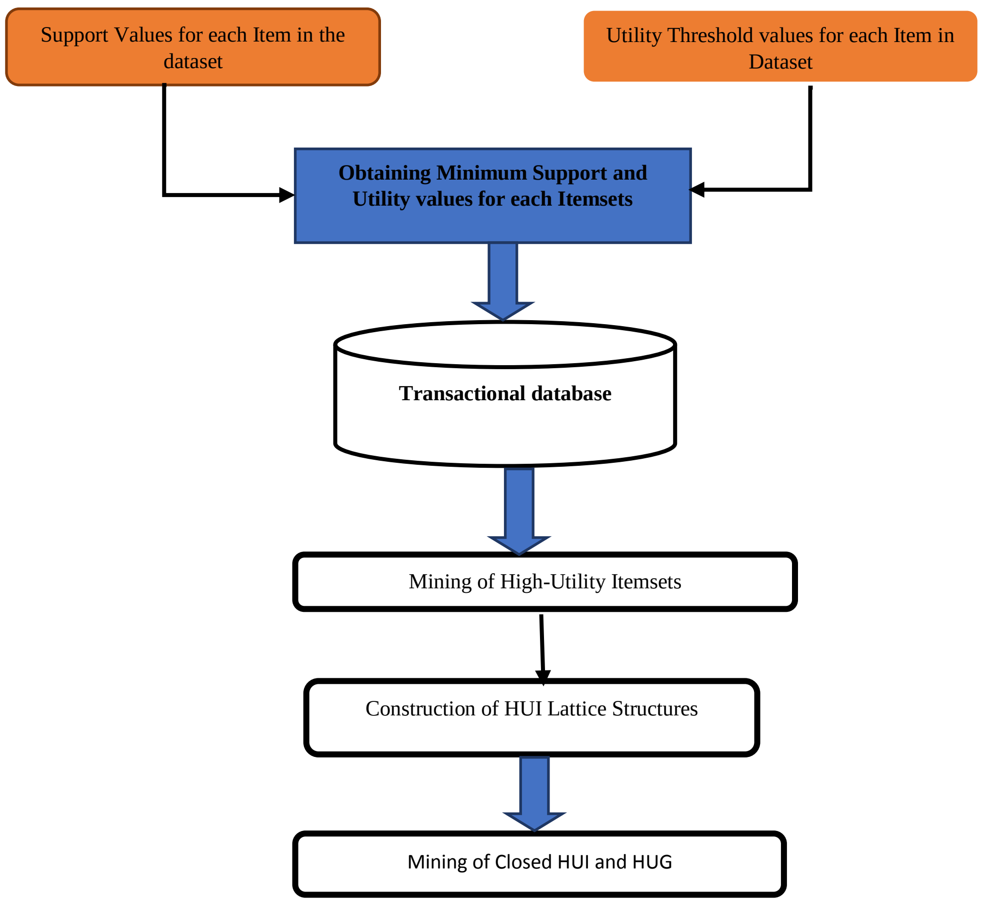

Figure1: Diagrammatic Representation of the Suggested Approach

Constructing a lattice structure using High-Utility Itemsets

The set of HUIs isextracted from Transaction Database D employing considered minimum utility values from multiple utility thresholds of each item. These mined HUIs are employed for the construction of Lattices known as High-Utility Itemset Lattice (HUIL) structure. In this section, the lattice is referred to as a semi-lattice. A detailed description of the construction of lattices is given in Algorithm 1 and lattice representation is given in Figure 2. A lattice structure is build using HUIs where every node comprises HUI,Is_{Gen} flag, andIs_{Closed} flag i.e. the generator flag HUIL structure. This structure includes the root node that is initially an empty set having support and utility values equivalent to 0, links, and children nodes amongst every group of nodes. The associations amongst these nodes are employed to determinethe parental and childrenassociation. Every nodecomprises of data pertaining to the Itemset such asminimum utility threshold, minimum support value, Is_{Closed} and Is_{Gen} flag."The $Is_{Closed}$the flag signifies that Itemset is a CHUI if itsvalue is true. The $Is_{Gen}$the flag signifies that Itemset is a generatorif its value is true. The entity of every node is constructed depending on the group of items"[44].

Initially, the approach calls the Set_{Latice} function for setting up the lattice with the empty rootnode. Also, it scans entire HUIs where every group of HUIs is arrangedusing the size of items. "Forevery HUI, the $Is_{Traverse}$flags ofchildren and root are set and called the function $Push_{Lattic}$ as to push or add the HUI into the lattice. Pertaining to the $Push_{Lattic}$ function, the variable flag is employed which specifies if Itemset{X} could be accumulated directlyinto the present node. If this current root node ($root_{node}$) has child nodes where every child node ($root_{node}$), Xcalls the $Push_{Lattic}$ function repetitively to insertnode {X}into the lattice with every child node as the root node. If there does not existany $child_{node} \in root_{node} \wedge L_c \subset X$, then {X} would be the children node ofthe present root node. To formulatethe data for extracting CHUIs and generators, there are twoflags, I Is_{Closed} and Is_{Gen}, which are fixed to HUI whenever it is inserted intothe lattice"[44,45]. From the outcome of lattice, the CHUI and HUG can be flexibility specified. Further, an approach is suggested to mine entire CHUIs and its related HUG from HUIL, as given in the next section.

<u>*Algorithm 1*</u>: Lattice of High-Utility Itemsets
Input: HUIs arranged using the items level in non-descending (HUIs)
Output: HUIL along with root node ($root_{node}$)

$Latice_{Set}()$
$root_{node} = \emptyset$
$For\ j = 1\ to\ HUIs_{Levels}.Cnt\ do$
    $For\ each$
        $X\ in\ HUIs_{Levels}[j]\ do$
         $root_{Node}.Is_{Traverse} = False$
         $For\ each$
             $children_{Node}\ in\ root_{Node}.Child\ do$
             If
              $children_{Node}.Itemset \subset X$
             $then$
              $child_{Node}.Is_{Traverse} = False$
                 $Reset_{FlagsOn}Lattice_{Nodes}\ (child_{Node})$
             End
         End
         $Push_{Node}(HUI, root_{Node})$
    End
End

$Reset_{FlagsOn}Lattice_{Nodes}\ (L_n)$
{
    $L_n.Is_{Traverse} = False$
    $For\ each$
    $L_c\ in\ L_n.Children\ do$
        $Reset_{FlagsOn}Lattice_{Nodes}(L_c)$
    End
}

$Push_{Nodes}(X, root_{Node})$
{
    If
     $root_{Node}.Is_{Traverse}\ then$
        $Return$
    End
}

$Flag = True$
$root_{Node}.Is_{Traverse} = True$
$For\ each$
$\quad child_{Node}\ in\ root_{Node}.Children\ do$
$\quad If\ child_{Node} \subset X\ then$
$\quad\quad Set\ Flag = False$
$\quad\quad Push_{Node}(X, child_{Node})$
$\quad End$
End
$If\ Flag = True$
$\quad then\ node_{Root}.Children.Add(X)$
$\quad If\ root_{Node}.Support = X.Support\ then$
$\quad\quad X.Is_{Gen} = False$
$\quad\quad root_{Node}.Is_{Closed} = False$
$\quad End$
End

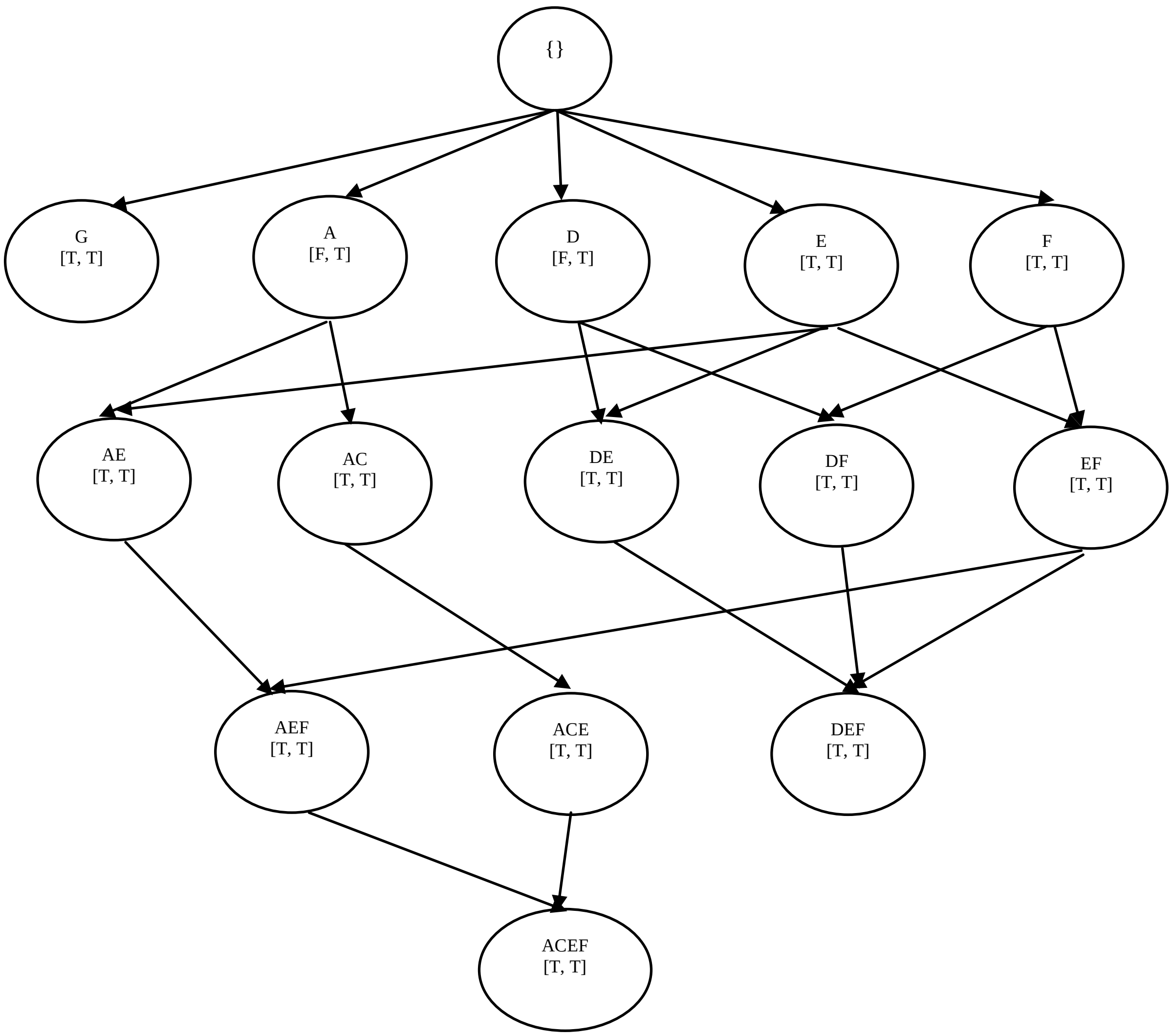

Figure 2: Lattice Representation of for extraction of CHUI and HUGs

Extraction of CHUI and HUG from constructed HUIL

From the constructed HUIL, the beneficial information about CHUIs and HUGs are extracted swiftly. Every node in the lattice structure depending on the results of Alg 1bringsIs_{Closed} and Is_{Gen}flags, definingwhether the Itemset is a CHUI or HUG. The complete description of the approach is given in Algorithm 2. An approach in this section is introduced to mine CHUI anda list of generators using multiple utility and support values known as the HUIL-Miner algorithm. Primarily, the approach passes throughentire children nodes from the roots of the lattice. For every childnode, it calls the function $Extend\_Mining\_CHUI\_HUG(L_c)$. "The function $Extend_Mining_CHUI_HUG(L_c)$will accumulate L_c to CHUIs list if$L_c.Is_{Closed}$ is True. The L_cItemset could be bothCHUI and a generator if it is HUCI and its Is_{gen} *the* flag is True. If L_c is a generator and not a CHUI, $Finding\ _CHUI_And_HUG(L_c)$is known to obtain CHUI that $L_c$pertains to"[44]. In this approach, a queue and list formats are employed, withentire children nodes of L_c as initialized values. If a queue has items, it functions on every Itemset$L_i$in the queue and accumulates L_c to be a generator of L_i if L_i is a CHUI and has similar support as L_c. If L_i has children nodes; further the approach endures to accumulate entire elements into the queue.

Algorithm 2: Mining of CHUIs and their HUGs ($CHUI_{List}$)from HUIL Approach
Input: HUIL with the $root_{Node}$
Output: CHUIs and their HUGs ($CHUI_{List}$)

```
Mining_CHUI_And_HUG ()
{
    For each
    L_c in root_Node.Children do
        Extend_Mining _CHUI_HUG(L_c)
    End
}

Extend_Mining _CHUI_HUG(L_c)
{
    If L_c.Flag = False then
        latticeNode.Is_Flag = True
        If (L_c.Is_Closed = True) then
            CHUI_List.Add_HighUtilityClosedItemset(L_c)
        End
        If (L_c.Is_Generator = True and L_c.Is_Closed = False) then
            Finding_CHUI_And_HUG(L_c)
        Else if (L_c.Is_Gen = True and L_c.Is_Closed = True) then
            CHUI_List.Add_Generator (L_c, L_c)
        End
        For each L_s in L_c.Children do
            Extend_Mining_CHUI_HUG(Ls)
        End
    End
}

Finding _CHUI_And_HUG(L_c)
{
    Queue = ∅, TrackList = ∅
    Add entire L_c.Children into Queue and TrackList
    While Queue = ∅ do
        Li = Queue.Dequeue()
        If L_c.Support = L_i.Support and L_i.Is_Closed = True.then
            Chui_List.Add_Generator(Li, Lc)
        End
```

$For\ each$
$L_s\ in\ L_i.Children\ do$
$If\ (L_s\ \_TrackList)\ then$
$Add\ L_s\ into\ Queue$
$Add\ L_s\ into\ TrackList$
End
End
End

}

Results and Discussion

In the area of HUI mining,none of the research has been done that uses multiple minimum support and multiple minimum utility thresholds at the same time except [1]. MHU-Growth and HUI-MMU are employed to authenticate the efficiency of the suggested algorithm which can provide the benchmark. Experiments areperformed on threepractical data samplescontainingnumerous characteristics. From the SPMF website [43], the foodmart, retail, and chess datasets were achieved. In Tables 3 and 4, it has been shown about the parameters and characteristics of the data sample employed correspondingly. A uniform distribution in [1, 10] is used to discover the internal utility values. A Gaussian (normal) distribution is employed to discover the external utility values.

$$MIS\ (i) = Max\ [\beta \times Sup\ (i), LS] \quad (1)$$
$$MU\ (i) = Max\ [\alpha \times pr\ (i), GLMU] \quad (2)$$

The Minimum Itemset Support (MIS) and Minimum Utility threshold (MU) values are assigned to each item using equation 1 and equation 2. In equation 1, the parameter β is used to control how the MIS values are associatedwith their frequencies where 0≤β≤1. If β=0 then a single MIS value that is LS is assigned to every item. In the equation 2, $pr(i)$ refers to the external utility of item i and to ensure the randomness of MU values the value of α set to different in different datasets such as 20k for foodmart, 80k for retail and 3k for the chess data sample.The comparison of the proposed approach is carried out in two different ways.

Runtime Analysis

The runtime analysis of the proposed association mining approach is matched with the existing association mining techniques in this section. The comparison of runtime analysis from Figure 2, Figure 3, and Figure 4 represent three datasets such as Food Mart, Retail, and Chess respectively and the proposed algorithm has lesser runtime when compared to the existing Utility Itemsets Mining techniques such as HUIApproach, IHUPApproach, and HUP-minerApproach. It is observed that runtime is rising linearly as the dimension of data size rises.

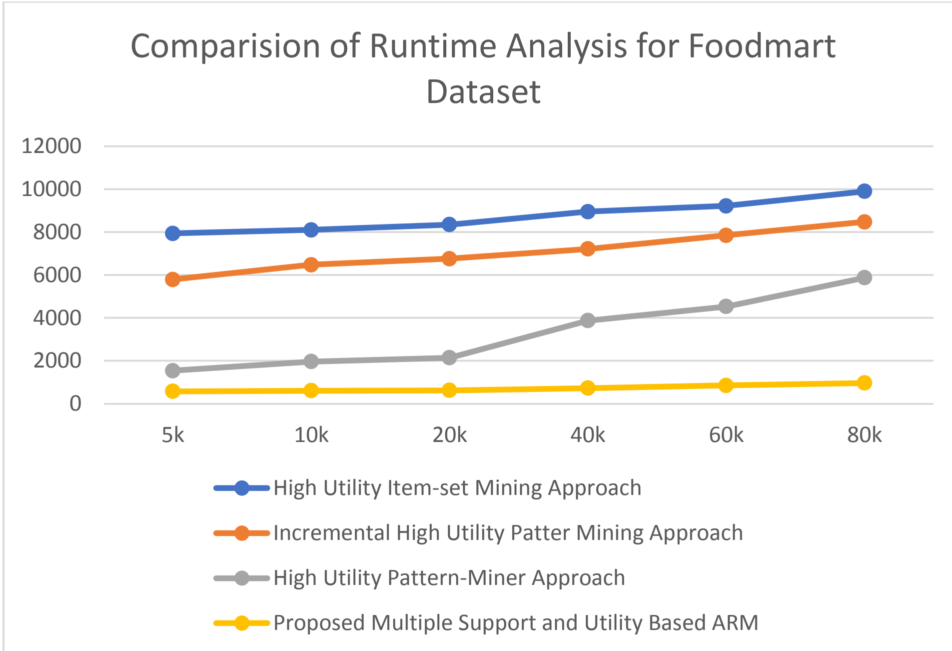

Figure 3: Comparison of Runtime Analysis for Food Mart Dataset

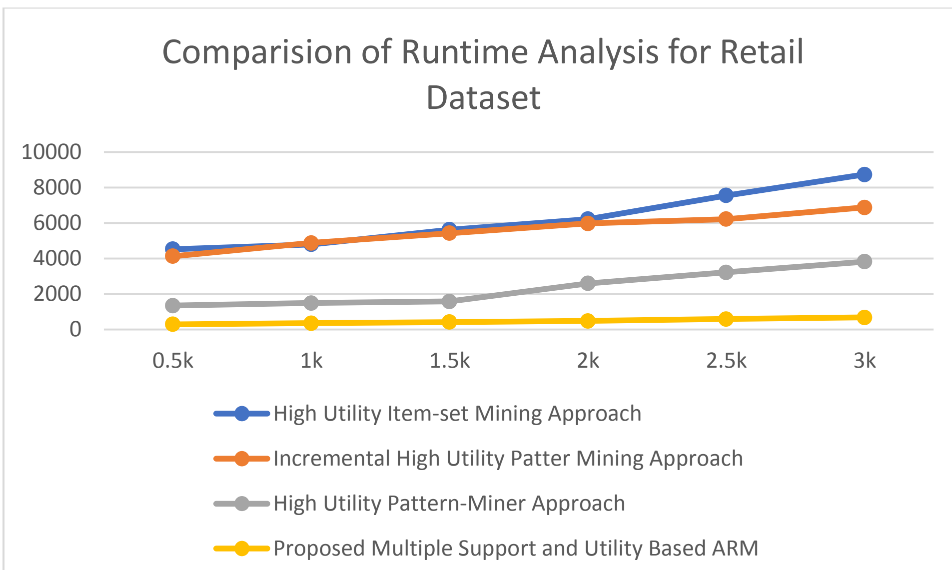

Figure 4: Comparison of Runtime Analysis for Retails Dataset

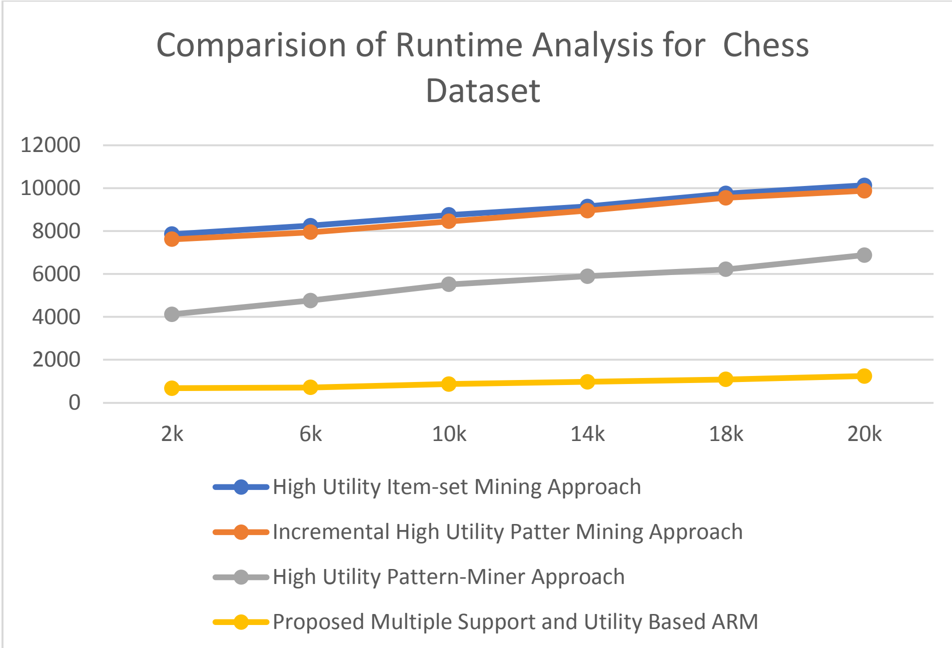

Figure5: Comparison of Runtime Analysis for Chess Dataset

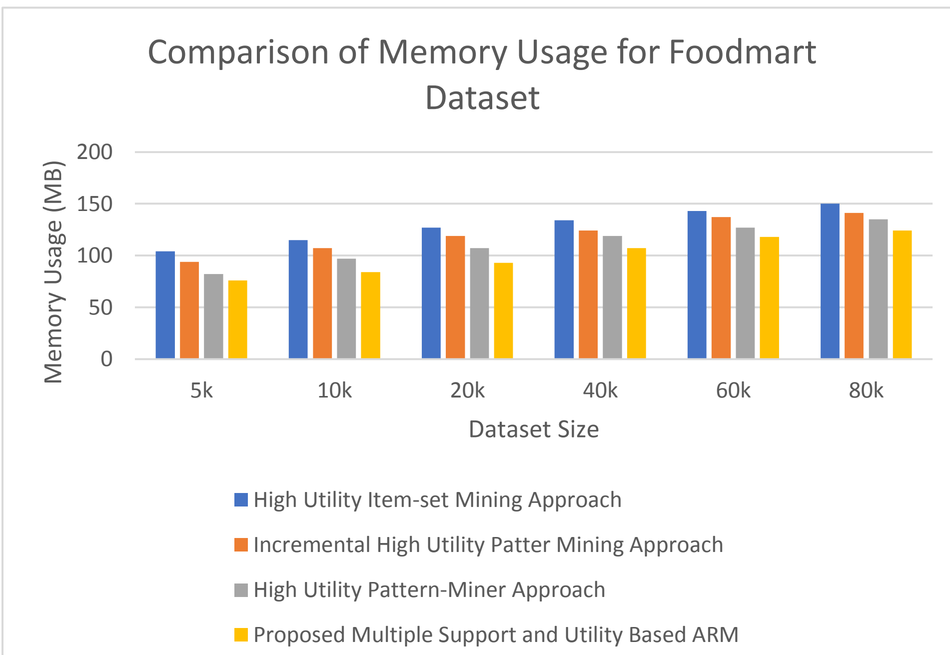

Figure 6: Comparison of Memory Usage for Food Mart Dataset

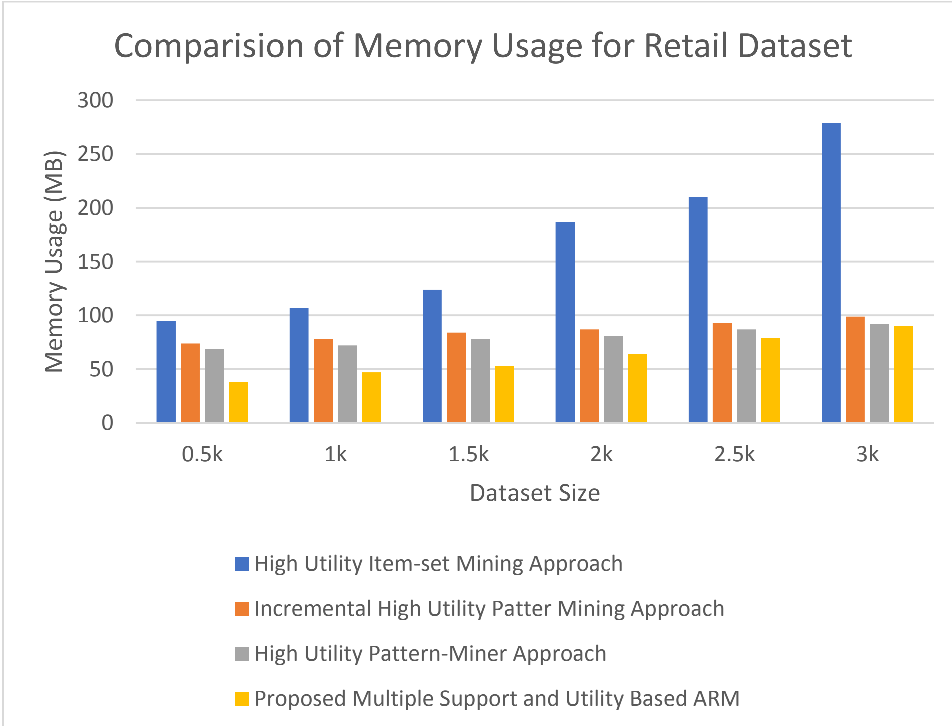

Figure 7: Comparison of Memory Usage for Retail Dataset

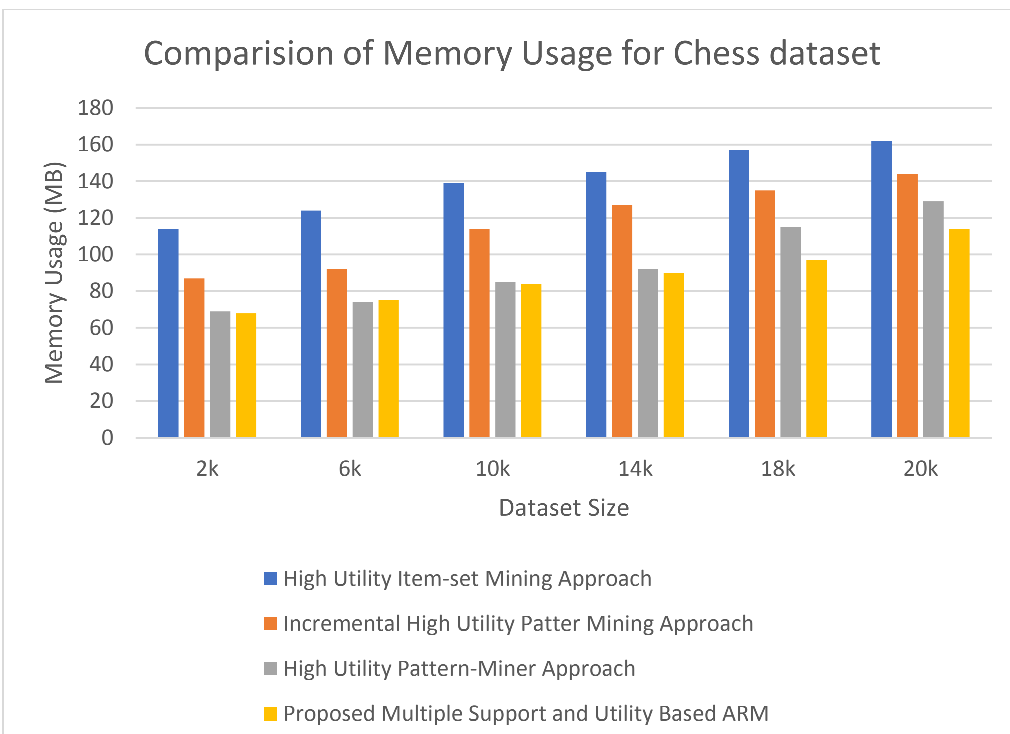

Figure 8: Comparison of Memory Usage for Chess Dataset

Memory Usage

The memory usage of the proposed association mining approach is compared with the existing association mining techniques in this section. For three datasets such as Food Mart, Retail and Chess respectivelyFigure 6, Figure 7 and Figure 8 signifies the evaluation of memory usage analysis andFigure 3, Figure 4 and Figure5 signifies that the suggested procedure has lesser memory usage when matched with the existing Utility Itemsets Mining techniques such as HUI Approach, IHUPApproach, and HUP-minerApproach. As the dimension of the data size rises the memory usage risesincreases linearly as observed.

Conclusions

An efficient mining approach is suggested in this paper, to extract the closed HUI and high utility generators from constructed high-utility Itemset lattice. For the extraction of CHUI and HUG in preference to single utility and support values for all Itemsets, multiple minimum utility threshold, and multiple minimum support values for each Itemset are used. The proposed approach is implemented in three different stages such as determination of Least Minimum Utility (LMU) and Least Minimum Support (LMS) values, Construction of Lattices using mined HUI known as High-Utility Itemset Lattice (HUIL) structure and Extraction of CHUI and HUG from constructed HUIL. The experimental results for the suggested approach are carried out using three practical data samples having diverse characteristics like Foodmart, Retails, and Chess. The proposed approach runtime and memory usage are matched with existing approaches and it has better performance values as shown.

References

[1] Agrawal, R., & Srikant, R. (1994)‘Fast algorithms for mining association rules in large databases’. In Proceedings
of the 20th International Conference on Very Large Data Bases. Vol. 1215, pp. 487–499. Morgan Kaufmann Publishers Inc.

[2] C.-H. Chen, T.-P. Hong and V.S. ‘Tseng, Genetic-fuzzy mining with multiple minimum supports based on fuzzy
clustering’, Soft Computing, Vol. 15 No. 12, pp. 2319-2333, 2011.

[3] G. Lee, U. Yun and K. Ryu, ‘Sliding window based weighted maximal frequent pattern mining over data streams’,
Expert Systems with Applications. Vol. 41 No. 2, pp. 694–708, 2014.

[4] C.-W. Lin, G.-C. Lan and T.-P. Hong, ‘An incremental mining algorithm for high utility Itemsets’, Expert Systems with Applications, vol. 39 no. 8, pp. 7173–7180, 2012.

[5] Y.-C. Liu, C.-P. Cheng and V.S. Tseng, ‘Discovering relational-based association rules with multiple minimum
supports on microarray datasets’. Bioinformatics.Vol. 27 No. 22, pp. 3142–3148, 2011.

[6] V.S. Tseng, B.-E. Shie, C.-W. Wu and P.S. Yu, ‘Efficient algorithms for mining high utility Itemsets from transactional databases’, IEEE Transactions on Knowledge and Data Engineering. Vol. 25 no. 8, pp. 1772–1786, 2013.

[7] C.-W. Wu, P. Fournier-Viger, P.S. Yu and V.S. Tseng, Efficient mining of a concise and lossless representation of high utility Itemsets, The 11th IEEE International Conference on Data Mining (ICDM 2011). 2011, pp. 824–833.

[8] U. Yun, G. Lee and K. Ryu.‘Mining maximal frequent patterns by considering weight conditions over data streams’. Knowledge-Based Systems.Vol. 55. pp. 49–65, 2014.

[9] U. Yun and K. Ryu. ‘Efficient mining of maximal correlated weight frequent patterns’. Intelligent Data Analysis. Vol. 17, no. 5.pp. 917–939, 2013.

[10] H. Mannila.‘Database methods for data mining’, In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 1998). 1998.

[11] V.P. Álvarez and J.M. Vázquez.‘An evolutionary algorithm to discover quantitative association rules from huge databases without the need for an Apriori discretization’. Expert Systems with Applications. Vol. 39. No. 1. 2012.

[12] R. Chaves, J. Ramírez, J.M. Górriz and C.G. Puntonet. 'Association rule-based feature selection method for Alzheimer's disease diagnosis', Expert Systems with Applications. Vol. 39. No. 14. pp. 11766–11774, 2012.
[13] G. Pyun, U. Yun, and K. Ryu, 'Efficient frequent pattern mining based on linear prefix tree', Knowledge-Based
Systems vol. 55, pp. 125–139, 2014.
[14] Lin, J.C.W., Gan, W., Fournier-Viger, P., Hong, T.P. 'Mining high-utility Itemsets with multiple minimum utility thresholds'. In: International C* Conference on Computer Science and Software Engineering, pp. 9–17. 2015.
[15] Ryang, H., Yun, U., Ryu, K. 'Discovering high utility Itemsets with multiple minimum supports'. Intelligent Data Analysis. Vol. 18 no. 6, pp.1027–1047. 2014.
[16] C. W. Lin, T. P. Hong, and W. H. Lu, 'An effective tree structure for mining high utility Itemsets'. Expert System Applications. Vol. 38, No. 6. pp. 7419–7424, 2011.
[17] V. S. Tseng, B. Shie, C. Wu, and P. S. Yu, 'Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases'. IEEE Transaction Knowledge Data Engineering. Vol. 25 No. 8, pp. 1772–1786, 2013.
[18] S. Zida, P. Fournier-Viger, J. C. Lin, C. Wu, and V. S. Tseng, 'EFIM : A Highly Efficient Algorithm for High-Utility Itemset Mining'. Lecture Notes in Computer Science. Vol. 9413, pp. 530–546, 2015.
[19] C. F. Ahmed, S. K. Tanbeer, B. Jeong, and Y. Lee, 'Mining High Utility Patterns in Incremental Databases'. Proceeding of 3rd International Conference on Ubiquitous Information Management and Communication. Vol. 21, No. 12, pp. 656–663, 2009.
[20] J. C. Lin, W. Gan, P. Fournier-viger, and T. Hong, 'Mining High-Utility Itemsets with Multiple Minimum UtilityThresholds'. Proceeding of the Eighth International Conference on Computer Science & Software Engineering. pp. 9-17. 2015.
[21] H. Ryang, U. Yun, and K. Ho.'Discovering high utility Itemsets with multiple minimum supports'. Intelligent Data Analysis, vol. 18. pp. 1027–1047, 2014.
[22] J. Han, J. Pei, and Y. Yin, 'Mining frequent patterns without candidate generation'. ACM SIGMOD Rec. vol. 29, no. 2, pp. 1–12, 2000.
[23] K. Sun and F. Bai, 'Mining Weighted Association Rules without Pre-assigned Weights'. vol. 20, no. 4. pp. 489–495. 2008.
[24] C. W. Lin, G. C. Lan, and T. P. Hong, "An incremental mining algorithm for high utility Itemsets," Expert System Appliances. vol. 39, no. 8, pp. 7173–7180, 2012.
[25] C. W. Lin, T. P. Hong, and W. H. Lu, 'An effective tree structure for mining high utility Itemsets'. Expert System Appliances. Vol. 38, no. 6. pp. 7419–7424, 2011.
[26] V. S. Tseng, B. Shie, C. Wu, and P. S. Yu, "Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases'. IEEE Transactions Knowledge Data Engineering. Vol. 25, no. 8, pp. 1772–1786, 2013.
[27] P. Fournier-Viger, C. W. Wu, S. Zida, and V. S. Tseng, 'FHM: Faster high-utility itemset mining using Estimatedutility co-occurrence pruning'. Lecture Notes in Computer Science, vol. 8502, pp. 83–92, 2014.
[28] S. Zida, P. Fournier-viger, J. C. Lin, C. Wu, and V. S. Tseng, 'EFIM: A Highly Efficient Algorithm for High-Utility Itemset Mining'. Lecture Notes in Computer Science, vol. 9413, pp. 530–546, 2015.
[29] Liu, Y., Liao, W.-k., & Choudhary, A. 'A two-phase algorithm for fast discovery of high utility Itemsets'. In T. Ho, D. Cheung, & H. Liu (Eds.), Advances in knowledge discovery and data mining. In Lecture Notes in Computer Science: 3518, pp. 689–695, 2005.
[30] Yao, H., & Hamilton, H. J. 'Mining itemset utilities from transaction databases'. Data & Knowledge Engineering, vol. 59 no. 3, pp. 603–626. 2006. http://dx.doi.org/10. 1016/j.datak.2005.10.004.
[31] Li, Y.-C. , Yeh, J.-S. , & Chang, C.-C. 'Isolated items discarding strategy for discovering high utility Itemsets'. Data & Knowledge Engineering. Vol. 64, no. 1, pp. 198–217, 2008.
[32] Lan, G.-C. , Hong, T.-P. , & Tseng, V. S. 'Discovery of high utility Itemsets from on-shelf time periods of products'. Expert Systems with Applications, vol. 38, no. 5, pp. 5851–5857. 2011.
[33] Ahmed, C. F., Tanbeer, S. K., Jeong, B.-S. , & Lee, Y.-K. 'Efficient tree structures for high utility pattern mining in incremental databases'. IEEE Transactions on Knowledge and Data Engineering, vol. 21 no. 12, pp. 1708–1721. 2009.

[34] Ahmed, C. F.,Tanbeer, S. K. , Jeong, B.-S. , & Lee, Y.-K. 'HUC-Prune: An efficient candidate pruning technique to mine high utility patterns'. Applied Intelligence, vol. 34, no. 2, pp. 181–198. 2011.
[35] Tseng, V. S., Wu, C.-W. , Shie, B.-E. , & Yu, P. S. 'UP-Growth: An efficient algorithm for high utility itemset
mining'. In Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining. pp. 253–262. 2011.
[36] Tseng, V. S. , Shie, B.-E. , Wu, C.-W. , & Yu, P. S. (2012). Efficient algorithms for mining high utility Itemsets from transactional databases. IEEE Transactions on Knowledge and Data Engineering, vol. 25. No. 8. pp. 1772–1786. 2012.
[37] Liu, M., & Qu, J. 'Mining high utility Itemsets without candidate generation'. In Proceedings of the 21st ACM international conference on information and knowledge management of ACM. In CIKM '12. pp. 55–64. 2012.
[38] Fournier-Viger, P. , Wu, C.-W. , Zida, S., & Tseng, V. S. 'FHM: Faster high-utility itemset mining using estimated utility co-occurrence pruning'. In International symposium on methodologies for intelligent systems. pp. 83–92, 2014.
[39] Krishnamoorthy, S. 'Pruning strategies for mining high utility Itemsets'. Expert Systems with Applications, vol. 42, no. 5, pp. 2371–2381. 2015.
[40] Zida, S, Fournier-Viger, P, Lin, J. C.-W. , Wu, C.-W. , & Tseng, V. S. 'EFIM: A fast and memory-efficient algorithm for high-utility itemset mining. Knowledge and Information Systems', vol. 51, pp. 595–625. 2017.
[41] Ryang, H., & Yun, U. 'Indexed list-based high utility pattern mining with utility upper-bound reduction and pattern combination techniques'. Knowledge and Information Systems, vol. 51, no. 2, pp. 627–659. 2017.
[42] Dawar, S. , Goyal, V. , & Bera, D. 'A hybrid framework for mining high-utility Itemsets in a sparse transaction database'. Applied Intelligence,pp. 1–19. 201.
[43] SPMF: an open-source data mining library. http://www.philippe-fournier-viger. com/spmf/.
[44] Thang Mai, Loan T. T. Nguyen. "An efficient approach for mining closed high utility Itemsets and generators", Journal of Information and Telecommunication, 2017
[45] Djenouri Y., Fournier-Viger P., Belhadi A., Chun-Wei Lin J. (2019) Metaheuristics for Frequent and High-Utility Itemset Mining. In: Fournier-Viger P., Lin JW., Nkambou R., Vo B., Tseng V. (eds) High-Utility Pattern Mining. Studies in Big Data, vol 51. Springer, Cham