IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 18, 2021, accepted March 3, 2021, date of publication March 10, 2021, date of current version March 23, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3065393

A Survey of Correlated High Utility Pattern Mining

RASHAD S. ALMOQBILY[1,2], AZHAR RAUF[2], AND FAHMI H. QURADAA[1,2]
[1]Department of Computer Science, Aden Community College, Aden 2402, Yemen
[2]Department of Computer Science, University of Peshawar, Peshawar 25120, Pakistan

Corresponding author: Rashad S. Almoqbily (rashadsaeedstd@uop.edu.pk)

ABSTRACT Pattern mining is an unsupervised data mining approach aims to find interesting patterns that can be used to support decision-making. High Utility Pattern Mining (HUPM) aims to extract patterns having high utility or importance which has broad applications in domains such as market basket analysis, product recommendation, bioinformatics, e-learning, text mining, and web click stream analysis. However, it has several limitations on real life scenarios; as a consequence, many extensions of HUPM appeared in the literature such as Correlated High Pattern Mining, Incremental Utility Mining, On-Shelf High Utility Pattern Mining, and Concise Representations of High Utility Patterns. The Correlated High Utility Pattern Mining aims to extract interesting high utility patterns by utilizing both Utility and Correlation measures. Several algorithms have been proposed to mine the correlated high utility patterns. These algorithms differ in the measures used to evaluate the interestingness of the patterns, data structures and pruning properties which they use to improve the mining performance. This paper presents a detailed survey on correlated high utility pattern mining, their methods, measures, data structures and pruning properties.

INDEX TERMS Frequent pattern, high utility pattern, interestingness measures, pattern mining, pruning properties.

I. INTRODUCTION

We are living in the data age where a huge amount of data is generated by different devices on the daily basis. Currently, 2.5 quintillion bytes of data are generated and by 2025, it is estimated that 463 exabytes of data will be generated on the daily basis globally [1]. A report by Intel says that 82% of commercial transactions need to be analyzed [2]. Due to the exponentially explosive growth of data, data mining has received a great deal of attention in order to turn such data into useful information [3]. Different types of data mining approaches have been proposed in order to analyze data [3]. Pattern mining is a type of unsupervised data mining approach which aims to find meaningful, useful, interesting and sometimes unexpected patterns that can be used to support decision-making [4], [5]. Numerous types of patterns can be extracted from the data using different types of pattern mining algorithms. Popular types of patterns are frequent patterns [6], high utility patterns [7], sequential patterns, trends, outliers, and graph structures [3], [7].

The associate editor coordinating the review of this manuscript and approving it for publication was Zeshui Xu.

The main task of frequent pattern mining is to find itemsets that frequently appear together in transactions of the database [8]. Although frequent patterns mining was intended for market basket analysis, it has broad applications in domains such as product recommendation, bioinformatics, e-learning, text mining, and web click stream analysis [4], [9], [10]. For instance, frequent patterns may be the words that co-occur frequently in a text, the products that are frequently purchased by customers such as {milk, bread}, or sequences of events that frequently lead to failures in a complex system. Recently, several methods have been proposed for frequent patterns mining [11]–[13].

Even though the mining of frequent pattern is useful, it depends on the assumption that each frequent pattern is interesting. Nevertheless, this assumption is not true for several applications [7], [14]. For instance, the pattern {bread, milk} in a transaction database may be extremely frequent but it may not be interesting as it may produce a low profit. In different circumstances, numerous patterns like {champagne, caviar} may yield a higher profit even if it is not frequent [15]. In order to address this limitation of frequent pattern mining, an emerging research area is Utility

Pattern Mining which aims to find high utility or important patterns [3], [7].

One limitation of high utility itemset mining methods is that they usually discover itemsets that have a high utility but the items constituting these patterns are weakly correlated or they may occur together by chance. Those patterns are useless or misleading for marketing decisions [15]–[21]. For example, consider a retail store database, the algorithms of high utility patterns mining may discover that buying a pen and a 60-inch plasma TV is a high-utility itemset, since these items generally create a high profit when purchased together. However, using this pattern to promote pen to customers who buy plasma TV would be a mistake. Since, if we look carefully, these items are weakly correlated and rarely sold together. This limitation is very critical. An experimental study showed that only less than 1% of the high utility patterns are strongly correlated [15], [21].

To solve the above stated issue, a number of algorithms have been proposed for mining patterns that are more interesting by utilizing both utility and correlation measures to find correlated high utility itemsets [16]–[21]. These techniques differ from each other in the measures used to evaluate the interestingness of the extracted patterns, the data structures and pruning properties that they used to reduce the search space and improve the mining performance.

This survey is focused on the Correlated High Utility Mining techniques, their measures, data structures and pruning properties. However, the next two subsections will present an overview of frequent pattern mining and high utility pattern mining respectively, in order to clarify how the pattern mining has been extended from frequent pattern to high utility pattern and then to correlated high utility pattern mining.

A. FREQUENT PATTERN MINING

Frequent pattern mining aims to find the associations among items in large transactional data sets. Let D be a transactional database. Let $I = \{i1, i2, \ldots, im\}$ be an itemset. Each transaction Tq in D is a set of items such that $Tq \subseteq I$, where q is a unique identifier for the transaction. The pattern (a set of items) is considered as frequent pattern if its support is equal or greater than the minimum support threshold.

Definition 1 (Support): The support of a pattern X in the transactional database D is denoted by $sup(X)$ and is defined as the proportion of transactions in the database that are matched by X. $Support(X) = count(X)/n$, where n is the total number of transactions in the database.

For example, Table 2 shows the frequent patterns for the transactional database in Table 1 with *minsup=0.4*.

Many frequent pattern mining algorithms have been developed in the last two decades such as Apriori [22], ECLAT [23], Frequent Pattern (FP)-growth [24] and negFIN [11]. All these methods use the *Support* measure to evaluate the interestingness of the patterns, while the Apriori property is used as a pruning property to reduce the search space.

TABLE 1. A transactional Database.

TID	Transaction
T1	(a,2), (b,6), (c,2), (d,2), (e,6)
T2	(b,5), (c,3), (d,3), (e,5)
T3	(b,2), (c,4), (d,2)
T4	(a,3), (b,2), (c,3), (d,2)
T5	(a,4), (c,4), (d,4), (e,12), (f,12)
T6	(a,3), (c,2), (d,2), (e,3), (g,3)
T7	(a,3), (c,4), (d,3), (f,4)
T8	(d,3), (g,1)
T9	(a,2), (c,3), (d,2), (f,3)
T10	(a,6), (c,5)
T11	(a,2), (c,2)

TABLE 2. The frequent itemsets for *minsup=0.4*.

1-itemset	Support	2-itemset	Support	3-itemset	Support
a	0.73	ac	0.73	acd	0.55
c	0.91	ad	0.55		
d	0.82	cd	0.73		

Property 1 (Apriori Property [22]): All nonempty subsets of a frequent itemset must also be frequent. This is because of the anti-monotonicity of support measure, which holds an upper bound property. That is, if an itemset is infrequent, all its supersets will be infrequent (The support of each itemsets cannot be greater than the support of any of its subsets).

The support measure is null-variant or it is affected by the total number of transactions in the database. Hence, when the minimum support threshold (*minsup*) is set to low value, many frequent itemsets containing weakly correlated items, are usually generated. Furthermore, a large portion of the extracted patterns are uninformative or redundant. In such case, setting a large value of *minsup* may overcome this issue; however a number of interesting itemsets will be pruned. Therefore, to solve this problem and in order to mine frequent correlated patterns many correlation measures were proposed in the literature such as bond, all-confidence, any-confidence [25], [26], coherence [27] and Kulczynsky [28], [29].

However, FPM methods depend on the assumption that each frequent pattern is interesting; this assumption may not be true for several applications. To address this limitation of the FPM an emerging research area called High Utility Pattern mining aims to extract patterns having high utility or importance.

B. HIGH UTILITY PATTERN MINING (HUPM)

HUPM is an emerging data mining task, which consists of extracting patterns having a high importance in quantitative databases. The utility of a pattern can be defined in terms of different objective criteria like its profit, risk, interest, significance, satisfaction or usefulness [15]. HUPM extends the problem of FPM by taking into account item quantities and profit.

Definition 2 (Quantitative database): Let $I = \{i1, i2, \ldots, im\}$ be a set of items and for each item $ip \in I(1 \leq p \geq m)$ profit unit (External Utility) denoted as $pr(ip)$, in each

transaction each item is associated with internal utility (Quantity) denoted as $q(ip,\ Td)$. A quantitative database $D = \{T1, T2, T3,\ldots Tn\}$ contains a set of transactions. Table 3 shows external utilities for the items in Table 1.

TABLE 3. External Utility.

a	b	c	d	e	f	g
3	4	4	1	5	4	1

Definition 3: Utility of an item *ip* in each transaction T*d* is denoted by $u(ip, Td)$ and is defined as $u\,(ip,\ Td) = q\,(ip,\ Td) \times pr(ip)$. Where $pr(ip)$is the external utility of an item *ip* and $q\,(ip,\ Td)$is the quantity of an item *ip* in transaction *Td*. For example, $u\,(b,\ T1) = 6 \times 4 = 24$.

Definition 4: Utility of an itemset X in the transaction Td is denoted by $u\,(X,\ Td)$ and is defined as $u\,(X,\ Td) = \sum_{ip \in X\ \&\ X \in Td} u(ip,\ Td)$. That is, the sum of the utilities of all items inside the pattern X in the transaction T*d*. For example, $u\,(bc,\ T1) = u\,(b,\ T1) + u\,(c,\ T1) = (6 \times 4) + (2 \times 4) = 32$.

Definition 5: Utility of an itemset X in the database D is denoted by $u\,(X)$ and is defined as $u\,(X) = \sum_{X \in Td\ \&\ Td \subseteq D} u(X,\ Td)$. That is, the sum of the utilities of the itemset X in all transactions containing it. For example, $u\,(bc) = u\,(bc,\ T1) + u\,(bc,\ T2) + \mathcal{C}u\,(bc,\ T3) + u\,(bc,\ T4) = 32+\ 32+\ 24+\ 20 = 108$.

Definition 6: An itemset X is called High Utility Itemset if $u(X) \geq minUtil$, where *minUtil* is the minimum utility threshold. For example, for the data presented in Table 1 with *minUtil*=90, the set of High Utility Itemsets are shown in Table 4.

Several methods have been developed for mining HUP [30]–[36]. All these methods use Utility measure to evaluate the interestingness of the patterns, while the Transaction Weighted Utilization (TWU) property is used as a pruning property in order to reduce the search space.

Definition 7: Utility of a transaction *Td* is denoted by $tu(Td)$ and is defined as the sum of the utilities of all items inside the transaction *Td*. $tu\,(Td) = \sum_{ip \in Td} u(ip,\ Td)$. For example the utility of the transaction $T2$ is calculated as $tu\,(T2) = u\,(b,\ T2) + u\,(c,\ T2) + u\,(d,\ T2) + u\,(e,\ T2) = 20 + 12 + 3 + 25 = 60$.

Definition 8: The Transaction Weighted Utilization (TWU) of an itemset X in database D is defined as $TWU\ (X) = \sum_{X \in Td\ \&\ Td \subseteq D} tu(Td)$. For example, $TWU\ (bc) = tu\,(T1) + tu\,(T2) + tu\,(T3) + tu\,(T4) = 70 + 60 + 26 + 32 = 188$.

Definition 9: An itemset X is called High Transaction-Weighted Utilization Itemset (HTWUI) if $TWU\ (X) \geq minUtil$, where *minUtil* is the minimum utility threshold. For example with *minUtil* =90, an itemset (bc) is HTWUI.

Property 2 (Transaction-Weighted Upper Bound Property): let X be a k-itemset, and Y be (k-1)-itemset such that Y⊂X. If X is HTWUI, Y is HTWUI as well. This means that if an itemset is Low Transaction-Weighted Utilization Itemset (LTWUI); all its supersets will be LTWUIs as well. Hence, this property can be used to reduce the search space by removing LTWUIs with their supersets from the search space.

Although, High Utility Pattern Mining has several applications; it has some limitations. As a consequence, many extensions of high utility pattern mining appeared in the literature such as Incremental Utility Mining [37]– [39] which aims to extract HUPs from dynamic databases, On-Shelf High Utility Pattern Mining [40]– [42] in which the shelf time of items is considered, Concise Representations of High Utility Patterns (e.g. Maximal itemsets [43], [44] and Closed High Utility Itemsets [45], [46]) that aim to extract a small list of meaningful HUPs.

One of the more critical limitation of the traditional high utility patterns mining algorithms is that they usually extract patterns having high utility but the items inside those patterns are weakly correlated or they may occurred together by chance. An experimental study proved that only 1% of the high utility itemsets are strongly correlated. Hence, many HUPs may be not interesting due to the weak correlations among the items inside patterns [15]. To address this limitation, researchers designed methods to extract correlated high utility patterns [16]–[21].

However, these extensions are based on the algorithms of High Utility Pattern mining. Each extension addresses a specific problem and has its own methods, measurements, data structures and pruning properties. In this survey we have focused on the Correlated High Utility Pattern Mining.

C. CORRELATED HIGH UTILITY PATTERN MINING (COHUPM)

COHUPM aims to mine high utility patterns that are correlated by considering both Utility and Correlation measures.

Definition 10 (Correlated High Utility Itemset): For a given quantitative database D with minimum utility threshold (*minUtil*) and minimum correlation threshold (*minCor*), the Correlated High Utility Itemset is an itemset X such that $u\,(X) \geq minUtil\ \delta\ Cor\ (X) \geq minCor$.

For measuring the correlation among items inside an itemset X, a number of measures were proposed. These measures with their methods are discussed in the next section.

II. ALGORITHMS FOR CORRELATED HIGH UTILITY PATTERN MINING

In order to extract more interesting pattern and to avoid misleading patterns resulted from the traditional methods of HUPM, a number of methods have been proposed to mine correlated high utility patterns by utilizing both utility and correlation measures. This section presents different COHUPM methods, their measures, data structures and pruning properties.

A. HIGH UTILITY INTERESTING PATTERN MINING (HUIPM)

High Utility Interesting Pattern Mining (HUIPM) with strong frequency affinity [17] was the first algorithm developed to

TABLE 4. The set of High Utility Itemsets with *minUtil=90*.

No	Itemset	Utility	No	Itemset	Utility	No	Itemset	Utility	No	Itemset	Utility
1	c	128	9	ce	174	17	adf	112	25	def	112
2	e	130	10	cf	120	18	aef	120	26	acde	172
3	ac	175	11	de	141	19	bcd	117	27	acdf	156
4	ae	132	12	ef	108	20	bce	119	28	acef	136
5	af	103	13	acd	138	21	bde	104	29	adef	124
6	bc	108	14	ace	164	22	cde	185	30	bcde	124
7	be	99	15	acf	147	23	cdf	129	31	cdef	128
8	cd	120	16	ade	140	24	cef	124	32	acdef	140

mine interesting pattern in high utility itemset in which the relation among items is meaningful.

1) FREQUENCY AFFINITY MEASURE

A Frequency Affinity measure has been proposed in this algorithm to be used with Utility measure in order to evaluate the interestingness of the desired patterns.

Definition 11: The Frequency Affinity of an itemset X in a transaction Td is denoted by $\chi(X,\ Td)$ and is defined as the minimum frequency value among items inside X in Td. For example, $\chi\ (bd,\ T2) = 3$ in Table 1.

Definition 12: The Frequency Affinity of an itemset X in the quantitative database D is denoted by $\chi(X)$ and is defined as $\chi\ (X) = \sum_{X\ \subseteq\ \mathrm{Td}\in \mathrm{D}} \chi(X,\ Td)$. It is the sum of the affinitive frequencies of the pattern X in all transaction containing it. For example, $\chi\ (bd) = \chi\ (bd,\ T1) + \chi\ (bd,\ T2) + \chi\ (bd,\ T3) + \chi\ (bd,\ T4) = 2 + 3 + 2 + 2 = 9$.

Definition 13: The interesting utility of an itemset X based on strong frequency affinity is denoted by $UA(X)$ and is defined as the sum of the external utilities of items inside X into the frequency affinity of X.

$UA\ (X) = \sum_{ip\in X} pr\ (ip) \times \chi(X)$. Where $pr\ (ip)$is the external utility of item ip. For example, $UA\ (bd) = pr\ (b) \times \chi\ (bd) + pr\ (d) \times \chi\ (bd) = 4 \times 9 + 1 \times 9 = 45$.

Definition 14: For a given quantitative database D and *minUtil* threshold, an itemset X is called High Utility Interesting Pattern if $UA\ (X) \geq minUtil$. For example, with $minUtil = 90$, {bd} itesmet is not high utility interesting patterns.

2) KNOWLEDGE WEIGHTED UTILIZATION (KWU) PROPERTY

KWU has been proposed in this method as a pruning property to reduce the search space.

Definition 15: The Frequency Affinity at stage i in transaction Td is denoted by $\chi\ (d,\ i)$ and is defined as the minimum frequency value among items at stage i inside transaction Td. A Frequency Affinity-based knowledge $K(d, i)$ consists of all the items at stage i inside transaction Td with $\chi\ (d,\ i)$, while the knowledge $K(d, i + 1)$ is prepared by removing $K(d, i)$ from $T(d, i)$. This process continues until the transaction Td becomes NULL. For example, for the transaction $T1$ in Table 1, $K(1, 1) =$ { *a, b, c, d, e: 2*}, $K(1, 2) =$ {*b, e: 4*}, $K(1, 3) = NULL$.

Definition 16: The knowledge utility of the knowledge $K(d, i)$ is denoted as $ku(Kd, i)$ and is defined as $ku(Kd, i) = \sum_{ip\in \mathrm{Kd},\ \mathrm{i}} pr\ (ip) \times \chi(d, i)$. For example, $ku(K1, 1) = pr\ (a) \times \chi\ (1, 1) + pr\ (b) \times \chi\ (1, 1) + pr\ (c) \times \chi\ (1, 1) + pr\ (d) \times \chi\ (1, 1) + pr\ (e) \times \chi\ (1, 1) = 3 \times 2 + 4 \times 2 + 4 \times 2 + 1 \times 2 + 5 \times 2 = 34$.

Definition 17: The Knowledge Weighted Utilization (KWU) of an itemset X is denoted by KWU(X) and is defined as the sum of knowledge utilities of all stages in each transaction containing X.

$KWU\ (\mathrm{X}) = \sum_{\mathrm{X}\subseteq\ \mathrm{Kd,i}\in \mathrm{Td}\in \mathrm{D}} ku(Kd, i)$. For example, $KWU\ (\mathrm{bd}) = ku(K1, 1) + ku(K2, 1) + ku(K3, 1) + ku(K4, 1) = 34 + 42 + 18 + 24 = 118$.

Definition 18: An itemset X is called high knowledge weighted utilization if $KWU\ (\mathrm{X}) \geq minUtil$.

Property 3: Downward closure property of KWU: Let X and Y be itemsets such that $X \subset \mathrm{Y}$, then the knowledge weighted utilization of Y cannot exceed the knowledge weighted utilization of X. Therefore, an upper bound property is holding on KWU, that is, if $KWU\ (X) < minUtil$, Y cannot be High Knowledge Weighted Utilization Itemset.

Based on Property 3, if an itemset X is low knowledge weighted utilization, all its supersets will be low knowledge weighted utilization. Consequently, X itemset and its supersets can be removed from the search space.

3) UTFA DATA STRUCTURE

Utility Tree based on Frequency Affinity (UTFA) has been introduced in this method as an efficient data structure to store sufficient information required for mining the desired patterns. HUIPM method stores the frequency-based knowledge in UTFA. A header table is used to store the items sorted based on their appearance order with their KWU and their frequency affinity.

Consider the first transaction $T1$ in Table 1. The knowledge extracting process from $T1$, according to Definition 15 and Definition 16, is shown in Fig. 1. In $K1, 1,\ \chi\ (1, 1) = 2$ and $ku(K1, 1) = 34$. This knowledge is inserted into UTFA according to the items appearance as shown in Fig. 1(a). Then $K1, 1$ is removed from $T1$. The knowledge $K1, 2$ for the second stage is extracted and inserted into UTFA as shown in Fig. 1(b). The transaction $T1$ then becomes NULL and the knowledge extraction process for the second transaction is started in similar process as shown in Fig. 1(c) and Fig. 1(d).

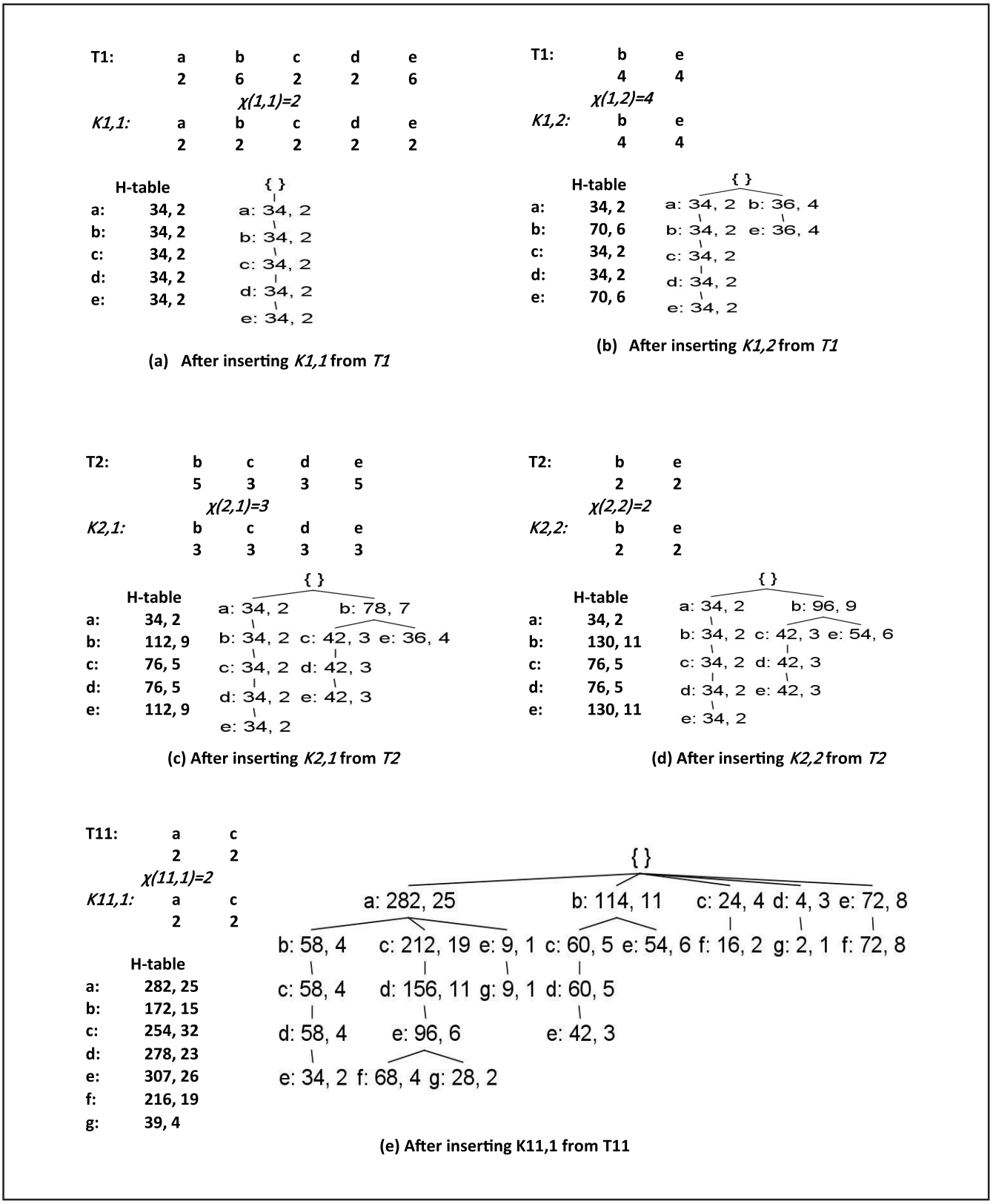

FIGURE 1. The construction process of HUIPM algorithm.

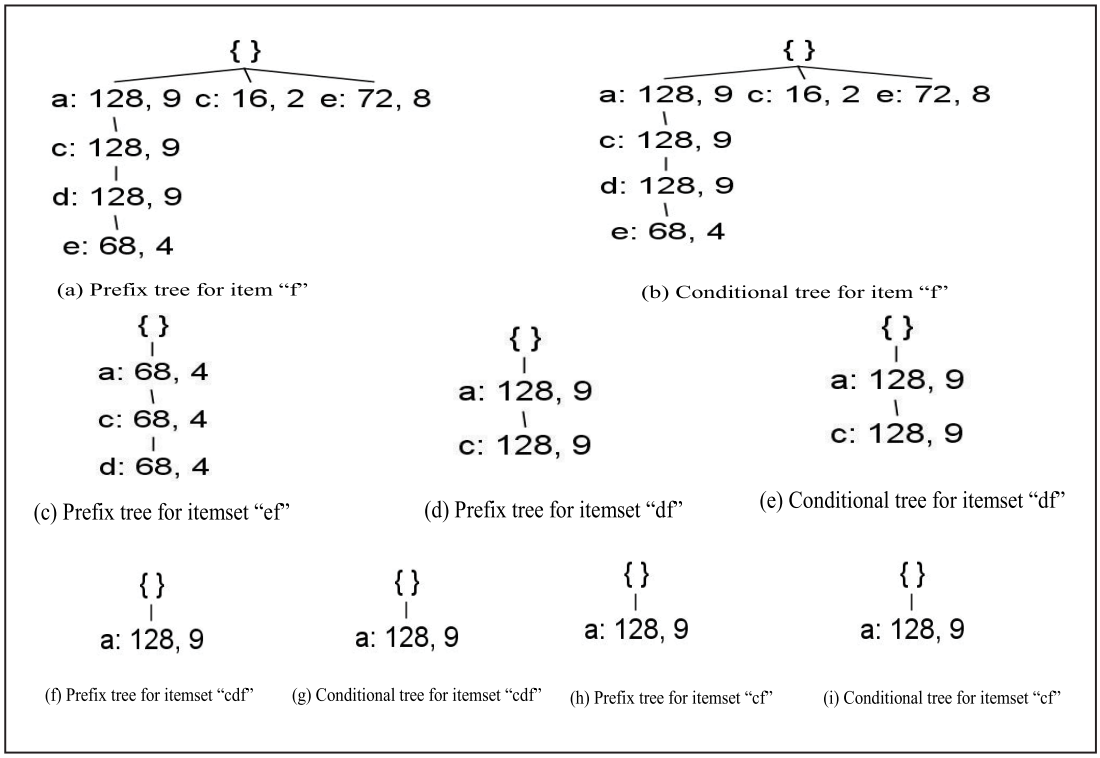

FIGURE 2. Mining process of HUIPM algorithm.

In this manner, all knowledge from all transactions of the database presented in Table 1 is extracted and inserted into the UTFA. The final tree after inserting the last knowledge $K11$, 1 from $T11$ is shown in Fig. 1(e).

4) THE PROCESS OF THE PATTERN MINING IN HUIPM

HUIPM first generates the prefix tree for the last item in the header table. The prefix tree of the item is created by taking all prefix branches with KWU and χ values. Consequently, the conditional tree of the selected itemset is created by removing items with their KWU is less than *minUtil* threshold.

Suppose *minUtil*=90, the HUIPM method will start from "g" item. Because KWU (g) is less than minUtil, "g" item is removed based on the Property 3. The prefix tree of the "f" item is shown in Fig. 2(a). Items "a", "c", "d" and "e" create HKWU itemsets with "f" item. Hence, the conditional tree for item "f" is shown in Fig. 2(b), while {af}, {cf}, {df}, {ef} and {f} candidate patterns are generated. Then the prefix tree of k-itemset is constructed from the conditional tree of (k-1)-itemset as shown in Fig. 2(c), Fig. 2(d), Fig. 2(f) and Fig. 2(h). At the same time the conditional tree of each itemset is constructed from its prefix tree in order to generate the candidate patterns. Once the candidate patterns are generated, the interesting utility is calculated for each candidate to derive HUIPs as shown in Table 5.

HUIPM method recursively creates a number of conditional trees to generate candidates and then derive interesting patterns. This procedure is time-consuming [19].

B. FAST ALGORITHM FOR MINING DISCRIMINATIVE HIGH UTILITY PATTERNS (FDHUP)

FDHUP method [19] has been proposed to improve HUIPM by proposing efficient data structures to store sufficient information for mining the interesting patterns efficiently and by developing new pruning property to reduce the search space.

Frequency Affinity and Utility measures are also used in this method to evaluate the interestingness of patterns.

1) AFFINITIVE UTILITY AND REMAINING AFFINITIVE UTILITY

In this method a new pruning property based on sum of Affinitive Utility (AU) and Remaining Affinitive Utility (RAU) has been proposed to be used with Property 2 (TWU) in order to reduce the search space.

Definition 19: The Affinitive Utility of an itemset X in a transaction Td is denoted as $AU(X, Td)$ and is defined as

$$AU(X,\ Td)\ =\ eu(X) \times \chi(X,\ Td).$$

where, *eu(X)* is the total profit of X or the external utility of X which is defined as:

$$eu\,(X) = \sum_{ip \in X} pr\,(ip)$$

For example, the Affinitive Utility for itemset {ab} in T1 is computed as:

$$eu\,(ab) \times\ \chi\,(ab) = (3 + 4) \times 2 = 14.$$

TABLE 5. Calculation of interesting utility for the candidate patterns of "f" item.

No	Candidates Pattern	KWU	FA	UA	Result
1	af	128	9	(3×9) + (4×9)= 63	Pruned
2	cf	144	11	(4×11) +(4×11)= 88	Pruned
3	df	128	9	(1×9) + (4×9)= 45	Pruned
4	ef	140	12	(5×12) + (4×12)= 108	Pass
5	f	216	19	(4×19)= 76	Pruned
6	adf	128	9	(3×9) + (1×9) + (4×9)= 72	Pruned
7	cdf	128	9	(4×9) +(1×9) + (4×9)= 81	Pruned
8	acdf	128	9	(3×9) + (4×9) + (1×9) + (4×9)= 108	Pass
9	acf	128	9	(3×9) + (4×9) + (4×9)= 99	Pass

Definition 20: The affinitive utility of an itemset X in the database D is denoted as $AU(X)$ and is defined as:

$$AU\,(X) = \sum_{\mathrm{x}\in\ \mathrm{T}d\in\cap\mathrm{Td}\in\mathrm{D}} AU(X,\ Td)$$

For example, the affinitive utility of an itemset {b} in the database D is computed as:

$$AU\,(b,\ T1) + AU\,(b,\ T2) + AU\,(b,\ T3) + AU\,(b,\ T4) = 24 + 20 + 8 + 8 = 60.$$

Definition 21: The affinitive utility of an item ip under a pattern X in a transaction Td, is denoted as $\mathrm{AU}(ip \mid X,\ Td)$ and is defined as:

$$\mathrm{AU}\,(ip \mid X,\ Td) = \min\left[\chi\,(X,\ Td)\,,\ q\,(ip, Td)\right] \times pr\,(ip)\,, \quad ip\notin \mathrm{X} \cap ip\in \mathrm{T}d \cap \mathrm{X} \prec ip$$

where $\prec ip$ is the total order adopted in the FDHUP method (sorting items on their TWU ascending order). [b $\prec$ f $\prec$ e $\prec$ a $\prec$ d $\prec$ c].

For example, the affinitive utility of {a} item under the pattern {be} in T1 is computed as $\mathrm{AU}\,(a \mid be,\ T1) = \min\left[\chi\,(be,\ T1)\,,\ q\,(a, T1)\right] \times pr\,(a) = \min\left[6,\ 2\right] \times 3 = 6$.

Definition 22: The remaining affinitive utility of an itemset X in a transaction Td is denoted as $RAU(X, Td)$ and is defined as:

$$RAU\,(X,\ Td) = \sum_{ip\notin \mathrm{X}\ \cap\ ip\in \mathrm{T}d\ \cap\ \mathrm{X}\ \prec ip} \mathrm{AU}\,(ip \mid X,\ Td)$$

For example, the remaining affinitive utility of an itemset {be} in T1 is computed as:

$$\mathrm{RAU}\,(\mathrm{be},\ \mathrm{T1}) = \mathrm{AU}\,(a \mid be,\ T1) + \mathrm{AU}\,(d \mid be,\ T1) + \mathrm{AU}\,(c \mid be,\ T1) = 6+2+8 = 16.$$

Property 4: If the sum of affinitive utility and remaining affinitive utility of an itemset X is less than *minUtil* threshold, then this itemset and all its supersets are not Discriminative High Utility Patterns (DHUP). Thus these itemsets can be removed from the search space.

2) EI-TABLE AND FU-TABLE

Element Information table (EI-table) and Frequency Utility table (FU-table) data structures have been proposed in this method to store required information for mining the DHUP efficiently.

An EI-table of itemset X consists of three fields: TID refers to the transactions ids containing X, the Affinitive Frequency (AF) of X in each transaction, and the Remaining Affinitive Utility of X in each transaction. EI-table of 1-itemsets is shown in Fig. 3.

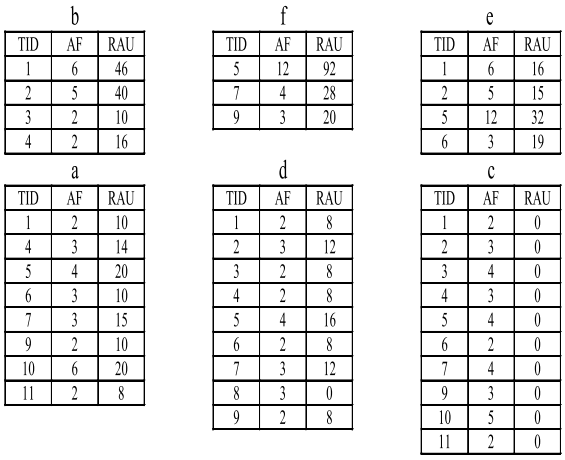

FIGURE 3. EI-table structures of 1-itemsets.

FDHUP method needs one scan for the database to construct EI-table for all 1-itemsets, then for $k \geq 2$ the EI-table of k-itemset is constructed from the EI-tables of (k-1)-itemsets. For instance EI-table of {be} itemset is constructed from EI-tables of {b} and {e} as shown in Fig. 4.

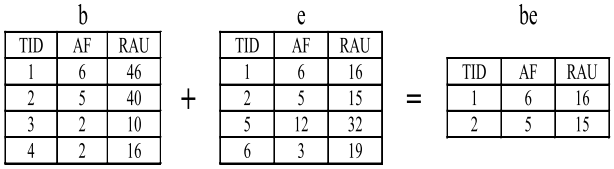

FIGURE 4. EI-table structures of [be] itemset.

After constructing EI-table of an itemset X, FU-table is constructed for that itemset. FU-table consists of four fields:

TABLE 6. The set of High Utility Interesting Patterns (HUIP) resulted by HUIPM and FDHUP algorithms.

No	Itemset	Interesting Utility	No	Itemset	Interesting Utility	No	Itemset	Interesting Utility	No	Itemset	Interesting Utility
1	ef	108	5	ce	99	9	ace	96	13	c	128
2	acdef	108	6	e	130	10	cd	100			
3	acf	99	7	cde	110	11	acd	120			
4	be	99	8	acde	104	12	ac	161			

the name of an itemset X, External Utility of X ($EU(X)$), the Affinitive Utility of X ($AF(X)$) and the summation of the remaining utility of X ($RAU(X)$). Fig. 5(a) shows FU-table for {b} itemset, while Fig. 5(b) shows FU-table for {be} itemset.

{b}

EU	AU	RAU
4	60	112

AU(b)=EU(b) × sum(AF(b) =4×15=60.
RAU(b)= sum(RAU(b)=46+40+10+16=112.

(a) FU-table of {b}

{be}

EU	AU	RAU
9	99	31

AU(be)=EU(be) × sum(AF(be) =9×11=99.
RAU(be)= sum(RAU(be)=16+ 15=31.

(b) FU-table of {be}

FIGURE 5. Constructing the FU-table of itemsetes.

If the TWU(k-itemset)<*minUtil* or $sum(AU + RAU) <$ *minUtil*, this itemset and all its supersets are not HUIP. Otherwise the interesting utility fo the itemset is calculated if $UA(K - itemset) >= minUtil$, this itemset will be considered as HUIP. Table 6 shows the High Utility Interesting Patterns resulted by both HUIPM and FDHUP methods for the data presented in Table 1.

C. FAST CORRELATED HIGH-UTILITY ITEMSET MINER (FCHM)

Fournier-Viger *et al.* [20] developed Fast Correlated high-utility itemset Miner (FCHM) algorithm for integrating the concept of correlation in high-utility itemset mining in order to extract profitable patterns that are highly correlated. Two version of the algorithm have been proposed: FCHM-bond and FCHMall-confidence which are based on bond and all-confidence measures that are already used for measuring frequent correlated patterns [25], [27], [47].

1) BOND MEASURE

To calculate the correlation of an itemset Bond measure has been introduced in this method has been used in FCHMbond method.

Definition 23: the disjunctive support of an itemset X is denoted as $dissup(X)$ and defined as: the total number of transactions containing at least one item from X.

$$dissup\,(X) = |\{T \in D \mid X \cap T \neq \emptyset\}|$$

For example, in Table 1, $dissup\,(ad) = 11$.

Definition 24: the correlation of an itemset X based on the Bond measure is defined as:

$$Bond\,(X) = \frac{supp(X)}{dissup(X)}$$

For example, in Table 1, $Bond\,(ad) = \frac{6}{11}$=0.55.

2) ALL-CONFIDENCE MEASURE

Based on all-confidence, the pattern is correlated if it has confidence equal or greater than the minimum all-confidence threshold.

Definition 25: the correlation of an itemset X based on the all-confidence measure is defined as:

$$all - confidence\,(X) = \frac{supp(X)}{argmax\{support(Y) | \forall Y \subset X \wedge Y = \phi\}}$$

For example, in Table 1, $all - confidence\,(ad) = \frac{6}{9}$=0.67.

3) IUTIL AND RUTIL BASED PROPERTY

Property 5: Beside using the pruning property of TWU, FCHM method also uses the pruning property based on summation of initial utility and remaining utility which has been proposed in HUI-Miner [33] for reducing the search space.

In the preprocessing step, FCHM sorts items in the ascending order based on their TWU. Hence, for the data in Table 1, the total order is [b $\prec$ f $\prec$ $e \prec$ a $\prec$ $d \prec c$].

Definition 26: Given a transaction Td and an itemset X such that $X \subset$ Td, all items that appear after X in Td according to the total order are denoted as T/X.

For example, in Table 1, $T1/\{ea\} = \{dc\}$, according to the total order.

Definition 27: the remaining utility of an itemset X in the transaction Td is denoted as Rutil(X, Td) and is defined as:

$$Rutil\,(X,\ Td) = \sum_{ip \in Td/X} u(ip,\ Td)$$

For example, in Table 1, and based on the total order, $Rutil\,(ea,\ T1) = 2 + 8 = 10$.

Property 6 (Anti-monotonicity of the bond and all-confidence measures): FCHMbond and FCHMall-confidence utilize the Anti-monotonicity of the bond and all-confidence measures respectively as pruning properties to reduce the search space. That is, for any two itemsets X and Y such that $X \subseteq Y$, $bond(X) \geqslant bond(Y)$ and $all - confidence(X) \geqslant all - confidence(Y)$ [47].

4) UTILITY LIST STRUCTURE

FCHM method uses utility list structure which has been proposed in [33]. Initial utility-lists keeping the utility information about input dataset, it required two scans for the dataset to be constructed. Firstly, TWU of all items are computed by a dataset scan. All 1-itemset having TWU less than *minUtil* are removed and no longer considered according to Property 2.

TABLE 7. The set of Correlated High Utility Patterns (COHUP) resulted by FCHMbond algorithm.

No	Itemset	Bond Value	Utility	No	Itemset	Bond Value	Utility	No	Itemset	Bond Value	Utility
1	c	1	128	4	de	0.44	141	7	ac	0.8	175
2	e	1	130	5	ce	0.4	174	8	cd	0.72	120
3	bc	0.4	108	6	adc	0.54	138				

TABLE 8. The set of Correlated High Utility Patterns (COHUP) resulted by FCHMall-confidence algorithm.

No	Itemset	all-confidence Value	Utility	No	Itemset	all-confidence Value	Utility	No	Itemset	all-confidence Value	Utility
1	c	1	128	5	bc	0.4	108	9	acd	0.8	138
2	e	1	130	6	de	0.44	141	10	ac	0.8	175
3	be	0.5	99	7	cde	0.4	185	11	cd	0.8	120
4	bcd	0.4	117	8	ce	0.4	174				

Utility list of {b}

TID	Iutil	Rutil
1	24	46
2	20	40
3	8	18
4	8	23

Utility list of {f}

TID	Iutil	Rutil
5	48	92
7	16	28
49	12	20

Utility list of {e}

TID	Iutil	Rutil
1	30	16
2	25	15
5	60	32
6	15	19

Utility list of {a}

TID	Iutil	Rutil
1	6	10
4	9	14
5	12	20
6	9	10
7	9	19
9	6	14
10	18	20
11	6	8

Utility list of {d}

TID	Iutil	Rutil
1	2	8
2	3	12
3	2	16
4	2	12
5	4	16
6	2	8
7	3	16
8	3	0
9	2	12

Utility list of {c}

TID	Iutil	Rutil
1	8	0
2	12	0
3	16	0
4	12	0
5	16	0
6	8	0
7	16	0
9	12	0
10	20	0
11	8	0

FIGURE 6. Initial Utility list structure for all 1-itemsets.

Utility list of {a}

TID	Iutil	Rutil
1	6	10
4	9	14
5	12	20
6	9	10
7	9	19
9	6	14
10	18	20
11	6	8

+

Utility list of {d}

TID	Iutil	Rutil
1	2	8
2	3	12
3	2	16
4	2	12
5	4	16
6	2	8
7	3	16
8	3	0
9	2	12

=

Utility list of {ad}

TID	Iutil	Rutil
1	8	8
4	11	12
5	16	16
6	11	8
7	12	16
9	8	12

Utility list of {ad}

TID	Iutil	Rutil
1	8	8
4	11	12
5	16	16
6	11	8
7	12	16
9	8	12

+

Utility list of {ac}

TID	Iutil	Rutil
1	14	0
4	21	0
5	28	0
6	17	0
7	25	0
9	18	0
10	38	0
11	14	0

=

Utility list of {adc}

TID	Iutil	Rutil
1	16	0
4	23	0
5	32	0
6	19	0
7	28	0
9	20	0

FIGURE 7. Constructing utility list structure for k-itemsets.

For dataset in Table 1, suppose the *minUtil*=90 and then FCHM no longer takes item {g} into consideration after the first dataset scan, while the remaining items in all transaction are sorted according to the total order.

The utility list of itemset X consists of three fields: *TID* refers to the transaction id (*Td*) containing itemset X, Iutil refers to the utility of X in transaction *Td* ($Iutil(X, Td)$), and Rutil refers to the Remaining Utility of X in *Td* ($Rutil(X, Td)$). Fig. 6. shows the initial utility list structure for all 1-itemsets.

Then, the utility list structure for k-itemset (k≥2) is constructed from the utility list of (k-1)-itemsets as shown in Fig.7.

To calculate the Iutil of k-itemset (k>=3) in the utility list the following formula is used:

$$\begin{aligned} &u(\{i1 \ldots i(k-2)\ i(k-1)\ ik\},\ Td) \\ &\quad = u(\{i1 \ldots i(k-2)\ i(k-1)\},\ Td) \\ &\qquad + u(\{i1 \ldots i(k-2)\ ik\},\ Td) \\ &\qquad - u(\{i1 \ldots i(k-2)\},\ Td) \end{aligned}$$

For example, $u(\{adc\}, T1) = u(\{ad\}, T1) + u(\{ac\}, T1) - u(\{a\}, T1) = 8 + 14 - 6 = 16$.

During the mining process, if TWU(k − itemset) < *minUtil* or Corr(k − itemset) <*minCorr* or sum(Iutil+ Rutil) <*minUtil*, this itemset and all its supersets are not COHUP and thus can be removed from the search space. Otherwise, the utility of k-itemset is calculated to drive the COHUP. For the running example, Table 7 shows the list of COHUP resulted by FCHMbond algorithm, while Table 8 shows the list of COHUP resulted by FCHMall-confidence algorithm.

D. NON-REDUNDANT CORRELATED HIGH-UTILITY ITEMSET MINING (CoHUIM)

Gan *et al.* [18] developed an algorithm named CoHUIM to extract non-redundant correlated purchase behaviors by considering the correlation and utility measures.

1) KULCZYNSKY MEASURE

The CoHUIM method uses the Kulczynsky (abbreviated as Kulc) measure [28], [29] in conjunction with Utility measure to evaluate the interestingness of the desired patterns.

Definition 28: The correlation between items inside an itemset X based on the Kulc measure is defined as the mean of the conditional probabilities of items:

$Kul\ (X) = \frac{1}{k} \sum_{ip \in X} \frac{sup(X)}{sup(ip)}$, where k is the number of items inside X.

For example, for the data in Table 1, $Kul\ (ab) = \frac{1}{2}\left(\frac{2}{8} + \frac{2}{4}\right) = 0.375$.

2) PROJECTED DATABASE

CoHUIM method uses the projection-based mechanism to reduce the size of the database, while Property 2(TWU) is used to reduce the search space.

Definition 29: given a transaction Td and an itemset X, the projected transaction (Td of X) is the set of postfix items based on the total order (sorted items ascending order on their support) and it is denoted as $Td_{|X}$. the total order is [f $\prec$ b $\prec e$ $\prec$ a $\prec$ d $\prec$ c].

For the data in Table 1, the projected transaction $T2_{|b} =$ {(e, d, c)}.

Definition 30: given a database D and an itemset X. The projected database from D using itemset X is denoted as $D_{|X}$ and is defined as $D_{|X} = \{Td_{|X} | Td_{|X} \in D \wedge Td_{|X} \neq \emptyset\}$. Fig.8. shows the projected databases of 1-itemsets.

$D_{|f}$

TID	Transaction
T5	(e, a, d, c)
T7	(a, d, c)
T9	(a, d, c)

$D_{|b}$

TID	Transaction
T1	(e, a, d, c)
T2	(e, d, c)
T3	(d, c)
T4	(a, d, c)

$D_{|e}$

TID	Transaction
T1	(a, d,c)
T2	(d, c)
T5	(a, d, c)
T6	(a, d, c)

$D_{|a}$

TID	Transaction
T1	(d, c)
T4	(d, c)
T5	(d, c)
T6	(d, c)
T7	(d, c)
T9	(d, c)
T10	(c)
T11	(c)

$D_{|d}$

TID	Transaction
T1	(c)
T2	(c)
T3	(c)
T4	(c)
T5	(c)
T6	(c)
T7	(c)
T8	Null
T9	(c)

FIGURE 8. Projected databases for 1-itemsets.

Then the projected databases of k-itemsets ($k \geq 2$) is constructed by scanning the projected database of (k-1)-itemset without needing to scan the whole database. For example, $D_{|ba}$ can be constructed from $D_{|b}$ as shown in Fig.9.

$D_{|ba}$

TID	Transaction
T1	(d, c)
T4	(d, c)

FIGURE 9. Projected database of ba itemset.

E. CORRELATED UTILITY-BASED PATTERN MINING (CoUPM)

An efficient utility mining approach namely (CoUPM) [21] was proposed by taking positive correlation and profitable value into account. The same measures (Kulc and utility) used in [18] are used in this method to evaluate the interestingness of the patterns. The utility list structure which is used in [48] is utilized as a data structure to store information required to mine the desired patterns in efficient manner, while Property 2 and Property 5 are used for reducing the search space.

F. MINING CORRELATED HIGH UTILITY ITEMSETS IN ONE PHASE (CoHUI-MINER)

Vo *et al.* [49] developed an algorithm named CoHUI-Miner to efficiently mine correlated high-utility itemsets. In this method, the same measures and pruning property used in [18] are used to evaluate the extracted patterns and reduce the search space respectively. Moreover, a new concept of prefix utility of projected transactions is added to the projected database mechanism, to make the Property 4 possible to be used for reducing the search space.

Definition 31: given a projected transaction $Td_{|X}$ the prefix utility of $Td_{|X}$ is denoted by $pru(Td_{|X})$ and is defined as $pru(Td_{|X}) = u(X,\ Td)$.

The projected databases for the 1-itemsets showed in Fig. 8 is updated by adding 'Prefix Utility' column and shown in Fig. 10.

$D_{|f}$

TID	Transaction	Prefix utility
T5	(e, a, d, c)	48
T7	(a, d, c)	16
T9	(a, d, c)	12

$D_{|b}$

TID	Transaction	Prefix utility
T1	(e, a, d, c)	24
T2	(e, d, c)	20
T3	(d, c)	8
T4	(a, d, c)	8

$D_{|e}$

TID	Transaction	Prefix utility
T1	(a, d,c)	30
T2	(d, c)	25
T5	(a, d, c)	60
T6	(a, d, c)	15

$D_{|a}$

TID	Transaction	Prefix utility
T1	(d, c)	6
T4	(d, c)	9
T5	(d, c)	12
T6	(d, c)	9
T7	(d, c)	9
T9	(d, c)	6
T10	(c)	18
T11	(c)	6

$D_{|d}$

TID	Transaction	Prefix utility
T1	(c)	2
T2	(c)	3
T3	(c)	2
T4	(c)	2
T5	(c)	4
T6	(c)	2
T7	(c)	3
T8	Null	3
T9	(c)	2

FIGURE 10. Projected databases with prefix utility for 1-itemsets.

Once the projected databases of 1-itemsets are constructed, the projected database of k-itemset (k$\geq$2)can be constructed from the projected database of (k-1)-itemset whiteout needing to scan the whole database. The Prefix Utility combined with pruning properties increased the performance of CoHUI-Miner algorithm. An experiment proved that CoHUI-Miner is two times faster than the CoHUIM algorithm [34].

Table 9 shows Correlated High Utility Patterns (COHUPs) resulted by the last three methods CoHUIM, CoUPM and CoHUI-Miner for the data presented in Table 1.

Table 10 shows a summary of the pattern mining methods and their features.

Different measures have been used to evaluate the interestingness of the patterns and thus different patterns have been extracted. Refer to Tables 4, 6, 7, 8 and 9 for the data presented in Table 1 having $minUtil$=90 and $minCor$=0.4. Thirty two patterns are considered as HUPs when the Utility measure is used. Out of these 32 HUPs, thirteen patterns are extracted as interesting when the Utility with FA measures are used, eight patterns are considered as interesting when the Utility with Bond measures are used, eleven patterns are

TABLE 9. The set of Correlated High Utility Patterns (COHUP) resulted by CoHUIM, CoUPM and CoHUI-Miner methods.

No	Itemset	Kulc Value	Utility	No	Itemset	Kulc Value	Utility	No	Itemset	Kulc Value	Utility
1	ac	0.9	175	9	ae	0.56	132	17	cd	0.84	120
2	acd	0.76	138	10	af	0.69	103	18	cde	0.62	185
3	acde	0.44	172	11	bc	0.7	108	19	cdf	0.54	129
4	acdf	0.5	156	12	bcd	0.61	117	20	ce	0.7	174
5	ace	0.48	164	13	bce	0.4	119	21	cf	0.65	120
6	acf	0.56	147	14	bde	0.4	104	22	de	0.72	141
7	ade	0.49	140	15	be	0.5	99	23	e	1	130
8	adf	0.57	112	16	c	1	128				

TABLE 10. Summary of the pattern mining methods and their features.

No of Phases	Algorithm	Patterns	Measures	Data Structures	Pruning Properties
Two Phases	Join-Based [22], [50] Tree-Based [23], [51]	Frequent Patterns	Support	None Set enumeration tree	Apriori property
One Phase	Pattern Growth [11], [24], [52]			Different	
Two Phases	A Two-Phase Algorithm [34]	High Utility Pattern	Utility	None	TWU
	UP Growth [54]			Prefix tree	(1)TWU (2)Sum of Iutil and Rutil
One Phase	EFIM [36]			Projected Database	TWU
	HUI-Miner [33]			Utility list	(1)TWU (2) Sum of Iutil and Rutil
	FHM [30]				
Two Phases	HUIPM [17]	COHUP	Utility and FA	UTFA	KWU
	FDHUP [19]			EI-table with FU table	(1)TWU (2)Sum of AU and RAU
One Phase	FCHMbond [20]		Utility and Bond	Utility list	(1)TWU (2)Sum of Iutil and Rutil (3)Antimonotonicity of Bond
	FCHMall-confidence [20]		Utility and all-confidence	Utility list	(1)TWU (2)Sum of Iutil and Rutil (3)Antimonotonicity of all-con
	CoHUIM [18]		Utility and Kulczenski	Projected database	TWU
	CoUPM [21]			Utility list	(1)TWU (2)Sum of Iutil and Rutil
	CoHUI-Miner [49]			Projected database with prefix utility	

extracted as interesting when the Utility with all-confidence measures are used, while twenty three patterns have been extracted when Utility with Kulc measures are used.

Based on the used measures, some of interesting patterns are missed due to mathematical formulas of the measures. On the other hand some misleading patterns are extracted, though they are uninteresting patterns. Extracting all interesting patterns accurately and avoiding misleading patterns are open research challenges with pattern mining approaches. The accuracy of the different extracted patterns will be discussed in section III-A.

III. A COMPARISON OF PATTERN MINING METHODS

This section presents a comparison of different pattern mining methods on their patterns mined, data structures, pruning properties, measures and the accuracy of the extracted patterns.

A. THE ACCURACY OF THE EXTRACTED PATTERNS

This section discusses the accuracy of different COHUPM methods. As discussed earlier, different COHUPs were extracted by different methods. The reason is being that these methods are based on different correlation measures. Table 11 shows the set of the patterns that are extracted from different pattern mining methods for the data presented in Table 1.

Table 11 shows that out of 32 HUPs, 13 patterns were extracted by HUIPM and FDHUP methods which use the FA measure to evaluate the interestingness of HUPs. FA measure however, cannot evaluate the real inherent correlation of

TABLE 11. Summary of the patterns extracted by each method.

No	Patterns	Methods used Utility measure [7], [30], [33], [34], [36], [54]	Methods used Utility + Frequency Affinity measure [17], [19]	Methods used Utility+ Bond measure [20]	Methods used Utility+ all-confidence measure [20]	Methods used Utility+ Kulc measure [18], [21], [49]
1	ac	√	√	√	√	√
2	acd	√	√	√	√	√
3	acde	√	√			√
4	acdef	√				
5	acdf	√	√			√
6	ace	√	√			√
7	acef	√				
8	acf	√	√			√
9	ade	√				√
10	adef	√				
11	adf	√				√
12	ae	√				√
13	aef	√				
14	af	√				√
15	bc	√		√	√	√
16	bcd	√			√	√
17	bcde	√				
18	bce	√				√
19	bde	√				√
20	be	√	√		√	√
21	c	√	√	√	√	√
22	cd	√	√	√	√	√
23	cde	√	√		√	√
24	cdef	√				
25	cdf	√				√
26	ce	√	√	√	√	√
27	cef	√				
28	cf	√				√
29	de	√		√	√	√
30	def	√				
31	e	√	√	√	√	√
32	ef	√	√			

itemset because the minimum value among items' quantities in the itemset is considered as a correlation value. Hence, if items' quantities are high by chance, these items will be considered as COHUPs even if they occurred together in limited number of transactions. For instance, itemset {ef} was considered as interesting, though {e} and {f} occurred together only in a single transaction. On the other hand, other patterns were considered as non-COHUPs, though they are correlated. For example, patterns {ade}, {adf}, {ae}, {bc}, {bcd}, {bce}, {bde} and {de} were considered as non-COHUPs while in actuality they are correlated, because they appear in many transactions together as obvious in Table 1.

The FCHMbond algorithm which is based on the Bond measure has identified 8 patterns as COHUPs out of 32 HUPs. Since the Bond measure is affected by the disjunctive support, some correlated patterns were considered as non-COHUPs. For example, patterns {acde}, {ace}, {ade}, {adf}, {ae}, {bcd}, {bce}, {bde}, {be} and {cde} were considered as non-CHUPs; though these patterns are correlated.

The FCHMall-confidence algorithm identified 11 patterns as COHUPs out of 32. The all-confidence measure is not suitable to assess the correlation on the dataset containing unstable transactions [21]. Hence, some interesting patterns are missed such as {acde}, {ace}, {ade}, {ae}, {bc}, {bcd}, {bce} and {bde}.

Similarly, CoHUIM, CoUPM and CoHUI-Miner methods in which the Kulc measure has been used, 23 patterns have been extracted as COHUPs. Kulc measure considers the mean of the conditional probabilities of items as a correlation value. Hence some imbalanced items were considered as COHUPs even if they are not correlated for the reason of relatively imbalanced items. For example, the pattern {cf} is relatively imbalanced because it contains unequal distribution of items; item {c} occurred in 10 transactions, whereas item {f} occurred in 3 transactions. Relatively imbalanced patterns are misleading to the decision making (e.g., It would be a mistake to promote item {f} to customers who buy item {c}). Similarly, patterns {acdf}, {acf}, {af}, and {cdf} were considered as CHUPs, while they are relatively imbalanced patterns.

TABLE 12. Performance analysis of Correlated High Utility Pattern Mining.

Algorithms	Datasets used in experiments	Compared Algorithms	Experiment's results
HUIPM	Retail, BMS-POS and kosarak [55]- [57] with synthetic utility values and a real-life dataset (Chain-store) using real utility values [58]	-Two-phase [43], [59]-FUM and DCG+ [60]	Their obtained results show that HUIPM significantly outperforms the other algorithms in terms of the execution time and memory usage.
FDHUP	Real life dataset [55] and synthetic dataset [61]	HUIPM	The result shows that the FDHUP outperforms the state-of-the-art HUIPM algorithm with respect to execution time, memory consumption, and scalability.
FCHM (FCHMBond and FCHMall-confidence)	Mushroom, retail, kosarak and foodmart datasets that are available in as part of the SPMF open source data mining library [62]	FHM [30]	The authors concluded that the FCHM algorithm can filter a huge amount of weakly correlated itemsets encountered in real datasets and thus run faster. Their results also show that the FCHMall-confidence algorithm is more efficient in terms of memory consumption as compared to the FCHM-bond and FHM algorithms.
CoHUIM	Real-life datasets (foodmart [63] and retail chess mushroom [55]) and two synthetic datasets (T10I4D100K and T5I2N2KD100K [63])	FDHUP	Their obtained results state that with a higher min-Corr threshold the CoHUIM outperforms the FDHUP algorithm for all non-dense datasets. The authors also concluded that the introduced sorted downward closure property and projection mechanism in the CoHUIM algorithm can prune unpromising candidates efficiently to reduce memory consumption. Hence, the CoHUIM algorithm is more feasible and acceptable in real-life scenarios.
CoUPM	Real life datasets (foodmart, chess, mushroom, retail and BMSPOS2) [62] and one synthetic dataset (T10I4D100K) [63]	CoHUIM	The results show that the CoUPM algorithm has significantly better performance than the state-of-the-art CoHUIM algorithm with respect to of execution time and memory usage.
CoHUI-Miner	Chess, Connect, mushroom, Accident, Kosarak and Chainstore that are available in as part of the SPMF open source data mining library [62]	CoHUIM	The experimental results show that the performance of the CoHUI-Miner algorithm in terms of running time and memory usage is much better than CoHUIM especially, with the dense and moderately dense datasets.

B. PERFORMANCE ANALYSIS OF CORRELATED HIGH UTILITY PATTERN MINING

This section presents a brief analysis of the performance of Correlated High Utility Pattern Mining algorithms. In the area of Correlated High Utility Pattern Mining, a number of experiments were conducted to evaluate the performance of the proposed algorithms with other stat-of-the-art algorithms on different datasets. Table 12 shows the detail of these experiments including datasets, algorithms, and results obtained.

IV. OPEN ISSUES AND RESEARCH OPPORTUNITIES

High utility pattern mining is an active research area in data mining. Different methods have been developed in last two decades to mine high utility patterns. Correlated High Utility Pattern Mining aims to address one of the core limitations of traditional HUPM methods. In the previous section we reviewed the key methods that were proposed in literature. This section presents key issues in the current COHUPM methods and discusses research opportunities.

A. INTERESTINGNESS OF THE EXTRACTED PATTERNS

Most of the current measures fail to properly extract accurate patterns, for the different reasons. For instance, Frequency Affinity measure fails to evaluate the inherent correlation among items, because only the minimum value among the quantities is considered as a correlation value. That is, if the quantities of items are high by chance, these items will be considered as highly correlated based on Frequency Affinity measure even if they occur together in limited number of transactions. The Bond and the all-confidence measures fail to evaluate the correlation when presented with a huge dataset containing many inconsistent transactions. The Kulc measure fails to evaluate the inherent correlation when it is presented with relatively imbalanced items, because it assumes balanced distribution of items when it calculates the correlation. Hence, in order to extract accurate patterns, new measures are required to evaluate the correlation among items for both balanced and imbalanced data.

B. DATA STRUCTURES

Efficiency is still a challenge for mining the correlated high utility patterns. The current data structures are not efficient for mining the desired patterns especially with the production of huge amount of Big data. Hence, it would be useful to develop more efficient data structures to store required information for mining the patterns efficiently.

C. PRUNING PROPERTIES

We discussed several pruning properties adopted in the current COHUPM methods. There is a need to explore new

pruning properties to reduce the search space that can significantly improve the mining performance.

V. CONCLUSION

Pattern mining is an unsupervised data mining approach that aims to find interesting patterns from a huge amount of data. The popular types of patterns mined are frequent patterns, high utility patterns, sequential patterns, trends, outliers, and graph structures. High Utility Pattern Mining has been extended to Correlated High Utility Pattern Mining (COHUP) which aims to extract interesting patterns for real life scenarios by utilizing both Utility and Correlation measures.

The interestingness of the extracted patterns and the time efficiency of the Correlated High Utility Pattern Mining methods have attracted the researchers' attention in order to find the most important patterns for decision making in efficient manner. For measuring the interestingness of the Correlated High Utility Patterns, a number of measures have been used in the literature. Several data structures and pruning properties have been proposed for reducing the database size and the search space respectively to make the mining process efficient. This survey paper analyzed and compared the COHUPM methods in the literature. It reviewed the state-of-the-art methods, key measures, data structures and the pruning properties in details. Furthermore, we stated the current issues in COHUPM methods and the future research opportunities.

REFERENCES

[1] J. Desjardins. (Apr. 15, 2019). *How Much Data is Generated Each Day?* [Online]. Available: https://www.visualcapitalist.com/how-much-data-is-generated-each-day/

[2] A. A. Saeed, A. Rauf, S. Khusro, and S. Mahfooz, "Compressed bitmaps based frequent itemsets mining on Hadoop," in *Proc. 10th Int. Conf. Informat. Syst.*, 2016, pp. 159–165.

[3] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts Techniques*. Amsterdam, The Netherlands: Elsevier, 2011.

[4] P. Fournier-Viger, J. C.-W. Lin, B. Vo, T. T. Chi, J. Zhang, and H. B. Le, "A survey of itemset mining," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 7, no. 4, Jul. 2017, Art. no. e1207.

[5] P. Fournier-Viger, J. C.-W. Lin, R. U. Kiran, Y. S. Koh, and R. Thomas, "A survey of sequential pattern mining," *Data Sci. Pattern Recognit.*, vol. 1, no. 1, pp. 54–77, 2017.

[6] C.-H. Chee, J. Jaafar, I. A. Aziz, M. H. Hasan, and W. Yeoh, "Algorithms for frequent itemset mining: A literature review," *Artif. Intell. Rev.*, vol. 52, no. 4, pp. 2603–2621, Dec. 2019.

[7] J.-F. Qu, M. Liu, and P. Fournier-Viger, "Efficient algorithms for high utility itemset mining without candidate generation," in *High-Utility Pattern Mining*. Cham, Switzerland: Springer, 2019, pp. 131–160.

[8] C. C. Aggarwal, M. A. Bhuiyan, and M. Al Hasan, "Frequent pattern mining algorithms: A survey," in *Frequent Pattern Mining*. Cham, Switzerland: Springer, 2014, pp. 19–64.

[9] C. C. Aggarwal, "Applications of frequent pattern mining," in *Frequent Pattern Mining*. Cham, Switzerland: Springer, 2014, pp. 443–467.

[10] S. Naulaerts, P. Meysman, W. Bittremieux, T. N. Vu, W. V. Berghe, B. Goethals, and K. Laukens, "A primer to frequent itemset mining for bioinformatics," *Briefings Bioinf.*, vol. 16, no. 2, pp. 216–231, Mar. 2015.

[11] N. Aryabarzan, B. Minaei-Bidgoli, and M. Teshnehlab, "NegFIN: An efficient algorithm for fast mining frequent itemsets," *Expert Syst. Appl.*, vol. 105, pp. 129–143, Sep. 2018.

[12] H. Bui, B. Vo, H. Nguyen, T.-A. Nguyen-Hoang, and T.-P. Hong, "A weighted N-list-based method for mining frequent weighted itemsets," *Expert Syst. Appl.*, vol. 96, pp. 388–405, Apr. 2018.

[13] Z. Zhang, W. Pedrycz, and J. Huang, "Efficient frequent itemsets mining through sampling and information granulation," *Eng. Appl. Artif. Intell.*, vol. 65, pp. 119–136, Oct. 2017.

[14] S. Zida, P. Fournier-Viger, J. C.-W. Lin, C.-W. Wu, and V. S. Tseng, "EFIM: A fast and memory efficient algorithm for high-utility itemset mining," *Knowl. Inf. Syst.*, vol. 51, no. 2, pp. 595–625, May 2017.

[15] P. Fournier-Viger, J. C.-W. Lin, T. Truong-Chi, and R. Nkambou, "A survey of high utility itemset mining," in *High-Utility Pattern Mining*. Cham, Switzerland: Springer, 2019, pp. 1–45.

[16] W. Gan, J. Chun-Wei, H.-C. Chao, T.-P. Hong, and P. S. Yu, "CoUPM: Correlated utility-based pattern mining," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 2607–2616.

[17] C. F. Ahmed, S. K. Tanbeer, B.-S. Jeong, and H.-J. Choi, "A framework for mining interesting high utility patterns with a strong frequency affinity," *Inf. Sci.*, vol. 181, no. 21, pp. 4878–4894, May 2011.

[18] W. Gan, J. C.-W. Lin, P. Fournier-Viger, H.-C. Chao, and H. Fujita, "Extracting non-redundant correlated purchase behaviors by utility measure," *Knowl.-Based Syst.*, vol. 143, pp. 30–41, Mar. 2018.

[19] J. C.-W. Lin, W. Gan, P. Fournier-Viger, T.-P. Hong, and H.-C. Chao, "FDHUP: Fast algorithm for mining discriminative high utility patterns," *Knowl. Inf. Syst.*, vol. 51, no. 3, pp. 873–909, Jun. 2017.

[20] P. Fournier-Viger, Y. Zhang, J. C.-W. Lin, D.-T. Dinh, and H. B. Le, "Mining correlated high-utility itemsets using various measures," *Log. J. IGPL*, vol. 28, no. 1, pp. 19–32, Jan. 2020.

[21] W. Gan, J. C.-W. Lin, H.-C. Chao, H. Fujita, and P. S. Yu, "Correlated utility-based pattern mining," *Inf. Sci.*, vol. 504, pp. 470–486, Dec. 2019.

[22] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. 20th Int. Conf. Very Large Data Bases (VLDB)*, vol. 1215, 1994, pp. 487–499.

[23] M. J. Zaki, "Scalable algorithms for association mining," *IEEE Trans. Knowl. Data Eng.*, vol. 12, no. 3, pp. 372–390, May/Jun. 2000.

[24] J. Han and J. Pei, "Mining frequent patterns by pattern-growth: Methodology and implications," *ACM SIGKDD Explor. Newslett.*, vol. 2, no. 2, pp. 14–20, Dec. 2000.

[25] E. R. Omiecinski, "Alternative interest measures for mining associations in databases," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 1, pp. 57–69, Jan. 2003.

[26] S. Bouasker and S. B. Yahia, "Key correlation mining by simultaneous monotone and anti-monotone constraints checking," in *Proc. 30th Annu. ACM Symp. Appl. Comput.*, Apr. 2015, pp. 851–856.

[27] M. Barsky, S. Kim, T. Weninger, and J. Han, "Mining flipping correlations from large datasets with taxonomies," 2011, *arXiv:1201.0233*. [Online]. Available: http://arxiv.org/abs/1201.0233

[28] S. Kulczynski, "Die pflanzenassoziationen der pieninen," Imprimerie de l'Universit, 1928.

[29] T. Wu, Y. Chen, and J. Han, "Re-examination of interestingness measures in pattern mining: A unified framework," *Data Mining Knowl. Discovery*, vol. 21, no. 3, pp. 371–397, Nov. 2010.

[30] P. Fournier-Viger, C.-W. Wu, S. Zida, and V. S. Tseng, "FHM: Faster high-utility itemset mining using estimated utility co-occurrence pruning," in *Proc. Int. Symp. Methodol. Intell. Syst.* Cham, Switzerland: Springer, 2014, pp. 83–92.

[31] J. C.-W. Lin, W. Gan, P. Fournier-Viger, T.-P. Hong, and V. S. Tseng, "Fast algorithms for mining high-utility itemsets with various discount strategies," *Adv. Eng. Informat.*, vol. 30, no. 2, pp. 109–126, Apr. 2016.

[32] Y. C. Lin, C.-W. Wu, and V. S. Tseng, "Mining high utility itemsets in big data," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Cham, Switzerland: Springer, 2015, pp. 649–661.

[33] M. Liu and J. Qu, "Mining high utility itemsets without candidate generation," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 55–64.

[34] Y. Liu, W.-K. Liao, and A. Choudhary, "A two-phase algorithm for fast discovery of high utility itemsets," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Berlin, Germany: Springer, 2005, pp. 689–695.

[35] U. Yun, H. Ryang, and K. H. Ryu, "High utility itemset mining with techniques for reducing overestimated utilities and pruning candidates," *Expert Syst. Appl.*, vol. 41, no. 8, pp. 3861–3878, Jun. 2014.

[36] S. Zida, P. Fournier-Viger, J. C.-W. Lin, C.-W. Wu, and V. S. Tseng, "EFIM: A highly efficient algorithm for high-utility itemset mining," in *Proc. Mex. Int. Conf. Artif. Intell.* Cham, Switzerland: Springer, 2015, pp. 530–546.

[37] P. Fournier-Viger, J. C.-W. Lin, T. Gueniche, and P. Barhate, "Efficient incremental high utility itemset mining," in *Proc. ASE BigData SocialInform.*, 2015, pp. 1–6.

[38] H. Ryang and U. Yun, "High utility pattern mining over data streams with sliding window technique," *Expert Syst. Appl.*, vol. 57, pp. 214–231, Sep. 2016.
[39] U. Yun and H. Ryang, "Incremental high utility pattern mining with static and dynamic databases," *Int. J. Speech Technol.*, vol. 42, no. 2, pp. 323–352, Mar. 2015.
[40] G.-C. Lan, T.-P. Hong, J.-P. Huang, and V. S. Tseng, "On-shelf utility mining with negative item values," *Expert Syst. Appl.*, vol. 41, no. 7, pp. 3450–3459, Jun. 2014.
[41] G.-C. Lan, T.-P. Hong, and V. S. Tseng, "Discovery of high utility itemsets from on-shelf time periods of products," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5851–5857, May 2011.
[42] T.-L. Dam, K. Li, P. Fournier-Viger, and Q.-H. Duong, "An efficient algorithm for mining top-*k* on-shelf high utility itemsets," *Knowl. Inf. Syst.*, vol. 52, no. 3, pp. 621–655, Sep. 2017.
[43] B.-E. Shie, P. S. Yu, and V. S. Tseng, "Efficient algorithms for mining maximal high utility itemsets from data streams with different models," *Expert Syst. Appl.*, vol. 39, no. 17, pp. 12947–12960, Dec. 2012.
[44] C. Manike and H. Om, "Modified GUIDE (LM) algorithm for mining maximal high utility patterns from data streams," *Int. J. Comput. Intell. Syst.*, vol. 8, no. 3, pp. 517–529, May 2015.
[45] T.-L. Dam, K. Li, P. Fournier-Viger, and Q.-H. Duong, "CLS-miner: Efficient and effective closed high-utility itemset mining," *Frontiers Comput. Sci.*, vol. 13, no. 2, pp. 357–381, Apr. 2019.
[46] C.-W. Wu, P. Fournier-Viger, J.-Y. Gu, and V. S. Tseng, "Mining closed+ high utility itemsets without candidate generation," in *Proc. Conf. Technol. Appl. Artif. Intell. (TAAI)*, Nov. 2015, pp. 187–194.
[47] N. B. Younes, T. Hamrouni, and S. B. Yahia, "Bridging conjunctive and disjunctive search spaces for mining a new concise and exact representation of correlated patterns," in *Proc. Int. Conf. Discovery Sci.* Berlin, Germany: Springer, 2010, pp. 189–204.
[48] P. Fournier-Viger, J. C.-W. Lin, T. Dinh, and H. B. Le, "Mining correlated high-utility itemsets using the bond measure," in *Proc. Int. Conf. Hybrid Artif. Intell. Syst.* Cham, Switzerland: Springer, 2016, pp. 53–65.
[49] B. Vo, L. V. Nguyen, V. V. Vu, M. T. H. Lam, T. T. M. Duong, L. T. Manh, T. T. T. Nguyen, L. T. T. Nguyen, and T.-P. Hong, "Mining correlated high utility itemsets in one phase," *IEEE Access*, vol. 8, pp. 90465–90477, 2020.
[50] J. Singh, H. Ram, and D. J. S. Sodhi, "Improving efficiency of Apriori algorithm using transaction reduction," *Int. J. Sci. Res. Publications*, vol. 3, no. 1, pp. 1–4, 2013.
[51] M. Song and S. Rajasekaran, "A transaction mapping algorithm for frequent itemsets mining," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 4, pp. 472–481, Apr. 2006.
[52] I. Feddaoui, F. Felhi, and J. Akaichi, "EXTRACT: New extraction algorithm of association rules from frequent itemsets," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2016, pp. 752–756.
[53] H. Q. Pham, D. Tran, N. B. Duong, P. Fournier-Viger, and A. Ngom, "NUCLEAR: An efficient methods for mining frequent itemsets and generators from closed frequent itemsets," *Inf. Technol. Ind.*, vol. 7, no. 2, Jan. 2021.
[54] V. S. Tseng, B.-E. Shie, C.-W. Wu, and P. S. Yu, "Efficient algorithms for mining high utility itemsets from transactional databases," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 8, pp. 1772–1786, Aug. 2013.
[55] B. Goethals. *Frequent Itemset Mining Dataset Repository*. Accessed: 2008. [Online]. Available: http://fimi.cs.helsinki/fi/data
[56] T. Brijs, G. Swinnen, K. Vanhoof, and G. Wets, "Using association rules for product assortment decisions: A case study," in *Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 1999, pp. 254–260.
[57] Z. Zheng, R. Kohavi, and L. Mason, "Real world performance of association rule algorithms," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2001, pp. 401–406.
[58] J. Pisharath. (2008). *NU-MineBench Version 2.0 Source Code and Datasets*. [Online]. Available: http://cucis.ece.northwestern.edu/projects/DMS/MineBench.html
[59] Y. Liu, W.-K. Liao, and A. Choudhary, "A fast high utility itemsets mining algorithm," in *Proc. 1st Int. Workshop Utility-Based Data Mining*, 2005, pp. 90–99.
[60] Y.-C. Li, J.-S. Yeh, and C.-C. Chang, "Isolated items discarding strategy for discovering high utility itemsets," *Data Knowl. Eng.*, vol. 64, no. 1, pp. 198–217, Jan. 2008.
[61] R. Agrawal and R. Srikant. *Quest Synthetic Data Generator*. [Online]. Available: http://www.Almaden.ibm.com/cs/quest/syndata.html
[62] P. Fournier-Viger, A. Gomariz, T. Gueniche, A. Soltani, C.-W. Wu, and V. S. Tseng, "SPMF: A java open-source pattern mining library," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3389–3393, 2014.
[63] Microsoft. *Example Database Foodmart of Microsoft Analysis Services*. [Online]. Available: http://www.Almaden.ibm.com/cs/quest/syndata.html

RASHAD S. ALMOQBILY received the B.S. degree in computer science from the Aden Community College, Aden, Yemen, in 2008, and the M.S. degree in computer science from the University of Peshawar, Peshawar, Pakistan, in 2018, where he is currently pursuing the Ph.D. degree with the Computer Science Department, Faculty of Numerical and Physical Sciences. Since 2012, he has been a Lecturer with the Department of Computer Science, Aden Community College. His research interests include data mining, data privacy, and data science.

AZHAR RAUF received the M.Sc. degree in computer science from the Department of Computer Science, University of Peshawar, Pakistan, in 1994, and the M.S. and Ph.D. degrees in computer science from Colorado Technical University, Colorado Springs, CO, USA, in 2002 and 2006, respectively. He has more than 24 years of experience that includes 17 years of teaching and research and seven years of IT industry experience. He has worked for different companies, including Bearing Point, Greatwest Healthcare, Oracle Corporation, Actiontech Electronics, Colorado, CO, USA, and Wilfrid Laurier University, Waterloo, ON, Canada. He is currently working as a Professor with the Department of Computer Science, University of Peshawar. He is the author or coauthor of more than 50 research publications. He has supervised a number of Ph.D. and M.S. students. His research interests include data mining, data warehousing, databases, data privacy, and big data.

FAHMI H. QURADAA received the B.S. degree in computer science from the Faculty of Administration and Computer and Information System, Thamar University, Thamar, Yemen, in 2001, and the M.S. degree in information and computer science from the College of Computer Science and Engineering, King Fahd University of Petroleum and Mineral, Dhahran, Saudi Arabia, in 2011. He is currently pursuing the Ph.D. degree with the Computer Science Department, Faculty of Numerical and Physical Sciences, University of Peshawar, Peshawar, Pakistan. Since 2011, he has been a Lecturer with the Department of Computer Science, Aden Community College, Aden, Yemen. His research interests include software engineering and data mining.