A Survey of Utility-Oriented Pattern Mining

Wensheng Gan, Jerry Chun-Wei Lin, *Senior Member, IEEE*, Philippe Fournier-Viger, Han-Chieh Chao, *Senior Member, IEEE*, Vincent S. Tseng, *Senior Member, IEEE*, and Philip S. Yu, *Fellow, IEEE*

Abstract—The main purpose of data mining and analytics is to find novel, potentially useful patterns that can be utilized in real-world applications to derive beneficial knowledge. For identifying and evaluating the usefulness of different kinds of patterns, many techniques and constraints have been proposed, such as support, confidence, sequence order, and utility parameters (e.g., weight, price, profit, quantity, satisfaction, etc.). In recent years, there has been an increasing demand for utility-oriented pattern mining (UPM, or called utility mining). UPM is a vital task, with numerous high-impact applications, including cross-marketing, e-commerce, finance, medical, and biomedical applications. This survey aims to provide a general, comprehensive, and structured overview of the state-of-the-art methods of UPM. First, we introduce an in-depth understanding of UPM, including concepts, examples, and comparisons with related concepts. A taxonomy of the most common and state-of-the-art approaches for mining different kinds of high-utility patterns is presented in detail, including Apriori-based, tree-based, projection-based, vertical-/horizontal-data-format-based, and other hybrid approaches. A comprehensive review of advanced topics of existing high-utility pattern mining techniques is offered, with a discussion of their pros and cons. Finally, we present several well-known open-source software packages for UPM. We conclude our survey with a discussion on open and practical challenges in this field.

Index Terms—Data science, economics, utility theory, utility mining, high-utility pattern, application

1 Introduction

Data mining [1], [2] focuses on extraction of information from a large set of data and transforms it into an easily interpretable structure for further use. It is an interdisciplinary field focused on scientific methods, processes, and systems to extract knowledge or insights from data in various forms, either structured or unstructured. Mining interesting patterns from different types of data is quite important in many real-life applications [1], [3], [4], [5], [6]. In recent decades, the task of interesting pattern mining [e.g., *frequent pattern mining* (FPM) [7], [8], *association rule mining* (ARM) [9], [10], *frequent episode mining* (FEM) [11], [12], [13], [14], and *sequential pattern mining* (SPM) [5], [15], [16], [17]] has been extensively studied. These are important and fundamental data mining techniques [1] that satisfy the requirements of real-world applications in numerous domains. Most of them aim at extracting the desired patterns using frequency or co-occurrence [7], [8], [9], [10], as well as other properties and interestingness measures [18], [19], [20], [21]. Despite the wide use of pattern mining techniques, most of these algorithms do not allow for the discovery of utility-oriented patterns, i.e., those that contribute the most to a predefined utility threshold, an objective function, or a performance metric. In general, some implicit factors, such as the utility, interestingness, or risk of objects/patterns, are commonly seen in real-world situations. The knowledge that is actually important to the user may not be found by traditional data mining algorithms. Therefore, a novel utility mining framework, called *utility-oriented pattern mining (UPM)* or *high-utility pattern mining (HUPM*[1]*)* [22], [23], [24], which considers the relative importance of items (*utility-oriented* [25]), has become an emerging research topic in recent years. In UPM, the *utility* (i.e., importance, interest, satisfaction, or risk) of each item can be predefined based on a user's background knowledge or preferences.

According to Wikipedia,[2] in economics, utility is a measure of preferences over some set of goods (including services, i.e., something that satisfies human wants). In a perspective, it represents satisfaction experienced by the consumer of a good. Hence, utility is a subjective measure. This definition indicates that a subjective value is associated with a specific value in a domain to express user preference. In practice, the value of utility is assigned by the user

- *W. Gan is with the Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China, and also with the University of Illinois at Chicago, Chicago, IL 60607 USA. E-mail: wsgan001@gmail.com.*
- *J. C. W. Lin is with the Western Norway University of Applied Sciences, Bergen 5063, Norway. E-mail: jerrylin@ieee.org.*
- *P. Fournier-Viger is with the Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China. E-mail: philfv8@yahoo.com.*
- *H. C. Chao is with the National Dong Hwa University, Hualien 974, Taiwan. E-mail: hcc@ndhu.edu.tw.*
- *V.S. Tseng is with the Department of Computer Science, National Chiao Tung University, Hsinchu City 30010, Taiwan. E-mail: vtseng@cs.nctu.edu.tw.*
- *P.S. Yu is with the University of Illinois at Chicago, Chicago, IL 60607 USA. E-mail: psyu@uic.edu.*

Manuscript received 21 May 2018; revised 5 Aug. 2019; accepted 15 Sept. 2019. Date of publication 20 Sept. 2019; date of current version 5 Mar. 2021. (Corresponding author: Jerry Chun-Wei Lin.)
Recommended for acceptance by L. Chen.
Digital Object Identifier no. 10.1109/TKDE.2019.2942594

1. The terms of UPM and HUPM can be interchangeably used but we will use UPM in the rest of this manuscript.
2. https://en.wikipedia.org/wiki/Utility

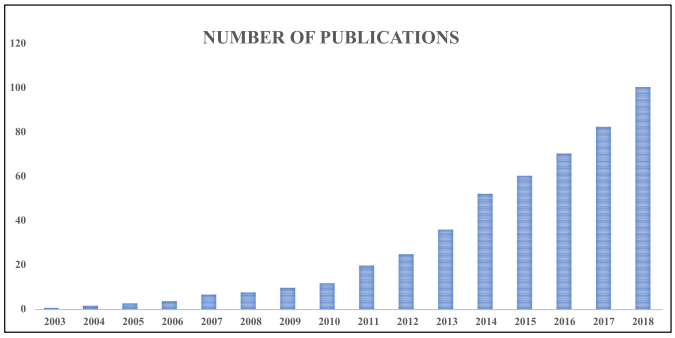


Fig. 1. Number of published papers that use "High Utility Pattern Mining" in sub-areas of data science and analytics. These publication statistics are obtained from Google Scholar. Note that the search phrase is defined as the sub-field named with the exact phrase "utility pattern," and at least one of utility or itemset/rule/episode/sequential pattern appearing, e.g., "utility itemset," "utility pattern," and "utility sequential pattern".

according to his interpretation of domain-specific knowledge measured by a specific value, such as cost, profit, or aesthetic value. According to the studies of Li et al. [18], interestingness measures can be classified as objective measures, subjective measures, and semantic measures [18], [20], [21]. Objective measures [21], [26], such as *support* or *confidence* for pattern mining, are based only on data itself, whereas subjective measures [27], [28], such as *unexpectedness* or *novelty*, take into account the user's domain knowledge. For the semantic measures [24], such as utility, they consider the data itself, as well as the user's expectation. Hence, utility is a quantitative representation of user preference, and the usefulness of an itemset is quantified in terms of its utility value. Utility can be defined as "A measure of how 'useful' (i.e., profitable) an itemset is" [24], [29]. Formally, a pattern is said to be useful to a user if it satisfies a specific utility constraint. In practice, the utility value of a pattern can be measured in terms of cost, profit, aesthetic value, or other measures of user preference.

To address these issues, *utility-oriented pattern mining* (hereinafter called UPM) has become a useful task and an important topic in data mining. In UPM, each object/item has an unit *utility* (e.g., unit profit) and can appear more than once in each transaction or event (e.g., purchase quantity). The *utility* of a pattern represents its importance or satisfaction, which can be measured in terms of risk, profit, cost, quantity, or other information depending on user preference. In general, the utility of a pattern is based on local transaction utility (also called *internal utility*) and *external utility* [24], [29]. The *internal utility* of an object/item is defined according to the information stored in a transaction/event, such as the quantity of the object/item occurred or sold. The *external utility* can be a measure for describing user preferences. Therefore, the utility of a pattern depends on the *utility function* specified by the user, which can be the *Sum*, *Average*, or *Multiplication* of quantity and profit of this pattern in databases. More specifically, the utility-based method for pattern mining can find various types of patterns that could not be identified using previous theories and techniques. According to previous studies, UPM has a wide range of applications, including website click-stream analysis [30], [31], [32], cross-marketing in retail stores [33], [34], mobile commerce environment [35], [36], gene regulation [37], and biomedical applications [38]. Through 15 years of study and development, many techniques and approaches have been extensively proposed for UPM in various applications. As shown in Fig. 1, there has been a rapid surge of interest of UPM in recent years in terms of the number of academic papers published in several subfields, including high-utility itemsets [29], high-utility rules [39], [40], high-utility sequential patterns [41], [42], and high-utility episodes [43], [44].

In spite of the fact that there are a considerable number of existing published studies and surveys about data mining, especially for pattern mining, none of them discuss UPM. Yet, after more than 15 years of theoretical development, a significant number of new technologies and applications have appeared in the UPM field. Unfortunately, there is no comprehensive survey of utility-oriented pattern mining methods and no study that systematically compares the state-of-the-art algorithms. We believe that now is a good time to summarize the new technologies and address the gap between theory and application. Here, we attempt to find a clearer way to present the concepts and practical aspects of UPM for the data mining research community. In this paper, we provide a systematic and comprehensive survey of the significant advances in UPM. The methods discussed in this article are not only important for high-utility pattern (i.e., itemset [24], [29], rule [39], [40], sequence, episode, etc.) mining but can also serve as inspiration for other data mining tasks [1], [2], including episode mining [11], [12], [13], [14], distributed data mining [45], and incremental/dynamic data mining [46], [47]. The major contributions are listed as follows:

1) This paper first presents the background, motivation, and a comprehensive survey of UPM (Section 1). This survey investigates more than 150 UPM papers published in the last 15 years and summarizes them in a systematic fashion.
2) This survey first introduces an in-depth understanding of UPM, including concepts, examples, comparisons with related studies (e.g., FPM, SPM), applications, and evaluation measures (Section 2). This survey presents a bird's eyes view, and then deeply and comprehensively summarizes the developments of UPM, comparing the state-of-the-art works to earlier works (Section 3).
3) A taxonomy of the most common and the state-of-the-art approaches for UPM is presented, including Apriori-based, tree-based, projection-based, vertical/horizontal-data-format-based, and other hybrid approaches (Section 3). We further analyze the pros and cons of each presented approach.
4) A comprehensive review of advanced topics of utility mining techniques (e.g., dynamic UPM, concise representation of utility patterns, HUSPM, HUEM, UPM in big data, and privacy issue) is presented (Section 4), with a discussion of their pros and cons. Not only the representative algorithms but also the advances and latest progress are reviewed.
5) We further review some well-known open-source software and datasets (Section 5) of UPM and hope that these resources may reduce barriers for future research. Finally, we identify several important issues and research opportunities for UPM (Section 6).

TABLE 1
Summary of Symbols and Their Explanations

Symbol	Definition
I	A set of m distinct items, $I = \{i_1, i_2, \ldots, i_m\}$.
D	A quantitative database, $D = \{T_1, T_2, \ldots, T_n\}$.
QSD	A quantitative sequential database = $\{s_1, s_2, \ldots, s_n\}$.
k-itemset	An itemset with k number of items in itself.
X	A k-itemset having k distinct items $\{i_1, i_2, \ldots, i_k\}$.
$sup(X)$	The support of a pattern X in D or QSD.
$q(i_j, T_q)$	The purchase quantity of an item i_j in transaction T_q.
$pr(i_j)$	The predefined unit profit of an item i_j.
$u(i_j, T_q)$	The utility of an item i_j in transaction T_q.
$u(X, T_q)$	The utility of an itemset X in transaction T_q.
$tu(T_q)$	The sum of the utilities of items in transaction T_q.
$minsup$	A predefined minimum support threshold.
$minconf$	A predefined minimum confidence threshold.
$minutil$	A predefined minimum high-utility threshold.
TWU	The transaction-weighted utility of a pattern.
TWDC	Transaction-weighted downward closure property.
HTWUI	A high transaction-weighted utilization itemset.
HUI	A high-utility itemset.
FPM	Frequent pattern mining.
SPM	Sequential pattern mining.
UPM	Utility-oriented pattern mining.
HUAR	High-utility association rule.
HUSP	High-utility sequential pattern.
HUSR	High-utility sequential rule.
HUE	High-utility episode.
HUIM	High-utility itemset mining.
HUARM	High-utility association rule mining.
HUSPM	High-utility sequential pattern mining.
HUEM	High-utility episode mining.

The remainder of this article is organized as follows. In Section 2, we introduce the necessary background information, the basic concepts and examples, and the applications in this field. In Section 3, we give a high-level overview of emerging UPM problems and survey several popular methods, as well as recent developments. In Section 4, we discuss advanced topics and techniques of UPM. In addition, several well-known open-source software and datasets are summarized in Section 5. We describe some open challenges and opportunities in Section 6. Several future directions are described in Section 7.

2 Basic Concept: Utility-Oriented Pattern Mining

2.1 Preliminary and Types of Utility Patterns

We first present the basic notations, as summarized in Table 1. Then, we introduce related preliminaries of UPM and then define the problem of UPM. Based on pattern diversity, utility-oriented pattern mining can be classified using the following basic criteria and extended patterns.

Definition 1 (Frequent Pattern and Association Rule [9]). *An association rule is an implication of the form, $X \rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \phi$. X (or Y) is a set of* items, *called an* itemset. *Given a database, a pattern (e.g., a set of items, sequences, structures, etc.) is said to be a frequent pattern if it occurs frequently in this database. Support of an itemset denoted as $sup(X)$ is the number of transactions containing X. An association rule R: $X \rightarrow Y$, in which X, Y are disjoint, and Y is non-empty, means that if a transaction includes X, then it also has Y. An association rule consists of frequent itemsets, and its confidence is no less than the minimum confidence sometimes called strong rules. It was first proposed by Agrawal et al. [9] in the context of frequent itemset and association rule mining. For example,* $\{Cheese, Milk\} \rightarrow$ Bread [sup = 5%, conf = 80%]; *this association rule means that 80 percent of customers who buy* Cheese *and* Milk *also buy* Bread, *and 5 percent of customers buy all these products together.*

Definition 2 (High-Utility Itemset, HUI [29], [48]). *The utility of an item i_j appearing in a transaction T_q is denoted as $u(i_j, T_q)$ and defined as $u(i_j, T_q) = q(i_j, T_q) \times pr(i_j)$. The utility of an itemset* X *in T_q is defined as $u(X, T_q) = \sum_{i_j \in X \wedge X \subseteq T_q} u(i_j, T_q)$. The total utility of* X *in a database* D *is $u(X) = \sum_{X \subseteq T_q \wedge T_q \in D} u(X, T_q)$. An itemset is said to be a high-utility itemset (HUI) if its total utility in a database is no less than the user-specified minimum utility threshold (such that $u(X) \geq minutil$); otherwise, it is called a low-utility itemset.*

Definition 3 (High-Utility Association Rule, HUAR [39], [40]). *Since the usefulness of association rule [9] can be defined as a utility function based on the business objective, the utility and confidence can be used to extend the concepts of high-utility itemset and association rule. An association rule R: $X \rightarrow Y$ is considered to have high utility if it meets the* minutil *constraint. Thus, a high-utility association rule (HUAR) consists of high-utility itemsets, and its confidence is no less than the minimum confidence. Generally speaking, discovery of HUIs started as the first phase in the discovery of HUARs, but it has been generalized by formulating a new pattern-mining framework.*

Definition 4 (High-Utility Sequential Pattern, HUSP [41], [42]). *The utility of an item (i_j) in a* q-*itemset* v *is denoted as $u(i_j, v)$, and defined as $u(i_j, v) = q(i_j, v) \times pr(i_j)$, where $q(i_j, v)$ is the quantity of (i_j) in v, and $pr(i_j)$ is the profit of (i_j). The utility of a* q-*itemset v is denoted as $u(v)$ and defined as $u(v) = \sum_{i_j \in v} u(i_j, v)$. The utility of a* q-*sequence $s = <v_1, v_2, \ldots, v_d>$ is denoted as $u(s)$ and defined as $u(s) = \sum_{v \in s} u(v)$. A sequence s in a quantitative sequential database* QSD *is said to be a high-utility sequential pattern (HUSP) if its utility is no less than the minimum threshold of s as $HUSP \leftarrow \{s | u(s) \geq minutil\}$. Considering the ordered sequences, high-utility sequential pattern mining (HUSPM) [41], [42] can discover more informative sequential patterns. This process is more complicated than the traditional UPM or SPM since the order and the utilities of itemsets should be considered together.*

Definition 5 (High-Utility Sequential Rule, HUSR [49]). *A sequential rule R: $X \rightarrow Y$ [50] is a relationship between two unordered itemsets $X, Y \subseteq I$ such that $X \cap Y = \emptyset$ and $X, Y \neq \emptyset$. The interpretation of a rule R: $X \rightarrow Y$ is that if items of X occur in a sequence, then items of Y will occur afterward in the same sequence. Let minsup, minconf $\in$ [0, 1] and minutil be thresholds set by the user and* QSD *be a sequence database.*

A sequential rule R is said to be a high-utility sequential rule (HUSR) [49] iff $u(R) \geq minutil$ and R is a valid rule, in which $u(R)$ is the total utility of R in QSD. Otherwise, it is said to be a low-utility sequential rule. The problem of mining high-utility sequential rules from a sequence database is the discovery of all high-utility sequential rules.

Definition 6 (High-Utility Episode, HUE [43], [44]). *An episode α is a non-empty totally ordered set of simultaneous events (SE) of the form $< (SE_1), (SE_2), \ldots, (SE_k) >$, where SE_i appears before SE_j for all $1 \leq i < j \leq k$. For example, $< (AB), (C) >$ is an episode containing a simultaneous event (AB) and a series event (C). The total utility of an episode α in a single simple or complex event containing a set of sub-events is $u(\alpha)$ [43], [44], and its calculation is more complicated than that of the utility of a sequence [42]. An episode is said to be a high-utility episode (abbreviated as HUE) in complex event sequences if its total utility in these sequences is no less than the minimum utility threshold such that $u(\alpha) \geq minutil$. Otherwise, this episode is a low-utility episode.*

Definition 7 (Utility-Oriented Pattern Mining, UPM). *A general definition of UPM is given below: UPM is a new mining framework that utilizes the utility theory and various mining techniques (e.g., data structure, pruning strategy, upper bound) to discover the interesting patterns (e.g., HUI, HUAR, HUSP, HUSR, HUE), and these derived patterns can lead to utility maximization and high benefit in business or other tasks.*

Based on the above concepts of utility pattern, the UPM framework can be further classified into the following categories, including 1) *high-utility itemset mining* (*HUIM*), 2) *high-utility association rule mining* (*HUARM*), 3) *high-utility sequential pattern mining* (*HUSPM*), 4) *high-utility sequential rule mining* (*HUSRM*), and 5) *high-utility episode mining* (*HUEM*).

2.2 Comparisons with Related Concepts

With the boom in data mining and analysis, all kinds of data have emerged, and a number of concepts (e.g., FPM, SPM, FEM, UPM, etc.) to model various types of data have been proposed. These concepts have similar meanings, as well as subtle differences. Here we compare the UPM framework with its most related concepts.

- *UPM versus FPM*. Frequent pattern mining (FPM) [7], [8], [9], [10] is a common and fundamental topic in data mining. FPM is a key phase of association-rule mining (ARM), but it has been generalized to many kinds of patterns, such as frequent sequential patterns [16], frequent episodes [11], and frequent subgraphs [51]. The goal of FPM is to discover all the desired patterns having support no lower than a given *minimum support* threshold. If a pattern has higher support than this threshold, it is called a frequent pattern; otherwise, it is called an infrequent pattern. Unlike UPM, studies of FPM seldom consider the database having quantities of items, and none of them considers the utility feature. Under the "economic view" of consumer rational choices, utility theory can be used to maximize the estimated profit. UPM considers both statistical significance and profit significance, whereas FPM aims at discovering the interesting patterns that frequently co-occur in databases. In other words, any frequent pattern is treated as a significant one in FPM. However, in practice, these frequent patterns do not show the business value and impact. In contrast, the goal of UPM is to identify the useful patterns that appear together and also bring high profits to the merchants [52]. In UPM, managers can investigate the historical databases and extract the set of patterns having high combined utilities. Such problems cannot be tackled by the support/frequency-based FPM framework.
- *UPM versus WFPM*. In the related areas, the relative importance of each object/item is not considered in the concept of FPM. To address this problem, weighted frequent-pattern mining (WFPM) was proposed [53], [54], [55], [56], [57], [58], [59]. In the framework of WFPM, the weights of items, such as unit profits of items in transaction databases, are considered. Therefore, even if some patterns are infrequent, they might still be discovered if they have high *weighted support* [53], [54], [55]. However, the quantities of objects/items are not considered in WFPM. Thus, the requirements of users who are interested in discovering the desired patterns with high risks or profits cannot be satisfied. The reason is that the profits are composed of unit profits (i.e., weights) and purchased quantities. In view of this, utility-oriented pattern mining has emerged as an important topic. It refers to discovering the patterns with high profits. As mentioned previously, the meaning of a pattern's utility is the interestingness, importance, or profitability of the pattern to users. The utility theory is applied to data mining by considering both the unit utility (i.e., profit, risk, and weight) and purchased quantities. This has led to the concept of UPM [52], which selects interesting patterns based on *minimum utility* rather than *minimum support*.
- *UPM versus SPM*. Different from FIM, sequential pattern mining (SPM) [5], [15], [16], [17], which discovers frequent subsequences as patterns in a sequence database that contains the embedded timestamp information of an event, is more complex and challenging. In 1995, Agrawal and Srikant first extended the FPM model to handle sequences [15]. Consider the sequence $<\{a, e\}, \{b\}, \{c, d\}, \{g\}, \{e\}>$, which represents five events made by a customer at a retail store. Each single letter represents an item (i.e., $\{a\}$, $\{c\}$, $\{g\}$, etc.), and items between curly braces represent an itemset (i.e., $\{a, e\}$ and $\{c, d\}$). Simply speaking, a sequence is a list of temporally ordered itemsets (also called events). Owing to the absence of time constraints in FPM not present in SPM, SPM has a potentially huge set of candidate sequences [16]. In a related area, through 25 years' study and development, many techniques and approaches have been proposed for mining sequential patterns in a wide range of real-world applications [5]. In general, SPM mainly focuses on the co-occurrence of derived patterns; it does not consider the unit profit and purchase quantities of each product/item.

So far, we have reviewed a wide range of pattern-mining frameworks that aim to discover various types of patterns, such as itemsets [9], [53], sequences [15], [16], and graphs [51]. These frameworks, however, only select high-frequency/support patterns. Patterns below the minimum threshold are considered useless and discarded. *Frequency* is the main interestingness measurement, and all objects/items and transactions are treated equally in such a framework. Clearly, this assumption contradicts the truth in

TABLE 2
Various Applications of UPM

Domain	Applications
Business intelligence	Market basket analysis, recommendation, cross marketing, sales intelligence, and risk prediction.
Web mining	Users' click-stream, users' access behaviors, and traversal pattern mining.
Mobile computing	Mobile e-commerce, travel route recommendation, spatial crowdsourcing, and spatial data analytics.
Stream processing	Web-click stream analysis, IoT data analytics, and stream mining in smart transportation.
Biomedicine	Gene expression and gene-disease association.

many real-world applications because the importance of different items/itemsets/sequences might be significantly different. In these circumstances, the frequency-/support-based framework is inadequate for pattern mining and selection. Based on the above concerns, researchers proposed the concept of UPM.

2.3 Why Utility-Oriented Pattern Mining and Analysis

With the rapid advancement of research on UPM, numerous applications in different domains have been proposed in recent years. We next describe several important applications, as summarized in Table 2.

• *Market Basket Analysis*. In market basket analysis, each transaction recorded with a customer contains several products/items, annotated with their purchase time, purchase quantities and the selling price. An important technique is based on the theory that if customers buy a certain set of items, customers are more (or less) likely to buy another set of items. For the problem of mining-characterized association rules from market basket data, the goal is to not only discover the buying patterns of customers but also the highly profitable patterns and customers. In some existing frameworks [33], [34], [60], [61], [62], the utility (i.e., importance, interest, or risk) of each product can be predefined based on users' background knowledge or preferences. As a result, UPM is able to offer richly detailed information about users' purchasing behaviors.

• *Web Mining*. There is much rich information in web data. For example, users' click-stream and purchase behaviors are recorded in web logs. In such data, a user's click operation (there are one or many clicks in one session) and browsing time on a web page can be expressed as the *internal utility* of the web page. Obviously, each web page has different importance depending on users' different preferences (i.e., *external utility*). Thus, UPM technology can be used to discover utility-oriented patterns from web logs, such as high-utility access patterns [63] and high-utility traversal patterns [64]. The derived results are quite useful for electronic commerce for such things as improving website services, providing some navigation suggestions for web browsing, and improving the design of web pages.

• *Mobile Computing*. With the explosive growth of the Internet of Things (IoT) [65] technologies in the Big Data era, such as smart-phones, wireless networks, and GPS devices, information about users' mobile behavior (e.g., locations and payment records) can be acquired and integrated in data analytics. In this scenario, utility-oriented mining technologies can be used to discover valuable user behaviors. Shie et al. first proposed a new framework to mine high-utility mobile sequences [35], [36] in mobile environments. It can extract associations between customers' purchase behaviors and location trajectories. The discovered high-utility patterns can be utilized for many applications essentially to mobile e-commerce, such as location-based advertisement or recommendations, navigational services, spatial crowdsourcing, and utility-based recommendation systems.

• *Stream Processing*. The majority of data is born as continuous streams [66], [67]: sensor events, user activity on a website, financial trades, and others; all these data are created as a series of events over time. In general, some stream data contain rich and important features that are similar to the general static data. In contrast to the support-based pattern mining technologies, the utility-oriented pattern mining technologies can be applied to extract useful pattens and knowledge from stream data, i.e., website click-streams [68]. Some preliminary studies have been carried out on this issue, such as [30], [32], [69], [70].

• *Biomedicine*. In gene expression data, each row represents a set of genes and their expression levels (i.e., internal utility) under an experimental condition. In addition, each gene has a degree of importance for biological processes (i.e., external utility). In bioinformatics, UPM technology can discover useful relationships between genes. For example, Liu et al. [71] applied a UPM method to successfully discover interesting gene regulation patterns from a time-course of comparative gene expression data. By analyzing the discovered results, medical researchers can find new drugs for the treatment of diseases. Recently, Zihayat et al. proposed a utility model by considering both the gene-disease association scores and their degrees of expression levels in a biological investigation [37].

• *Other Applications*. Since the "utility" of a pattern measures the importance of the pattern to the user (i.e., risk, weight, cost, and profit), UPM has broad real-life applications; several examples are described below. In risk prediction, the risk that events may occur is indicated by occurrence probabilities and risk. For example, the event $<(A,\ 1,\ 80);\ (D,\ 5,\ 15);\ (E,\ 3,\ 125);\ 90\%>$ indicates that this event consists of three sub-events $\{A, D, E\}$ with occurrence frequencies {1, 5, 3}, while their risk {80, 15, 125}, respectively, has a 90 percent probability of occurring. In e-commerce business, this manifests as identifying customers who visit web pages a number of times by taking pages visited as a utility parameter. In financial analysis, e.g., online banking fraud detection, the transfer of a large amount of money to an unauthorized overseas account may appear once or many times in several million transactions, yet it has a substantial business impact.

2.4 Evaluation Measures of Utility

Here, we briefly describe several key measures that have been used in the literature to determine utility-oriented relationships in UPM. In Section 2.1, the theoretical foundations

TABLE 3
Evaluation Measures of Utility in UPM

Measure	Description
Utility	The commonly used utility measure in many UPM models and algorithms and its definition has been given from Definitions 2 to 7, and details can be referred to [42], [52].
Average utility	The average utility is $au(X, T_q) = \sum_{i_j \in X \wedge X \subseteq T_q} q(i_j, T_q) \times pr(i_j)/\|X\|$ [72], where k is the number of items in X. It considers the length of pattern as a major factor.
Expected/ potential utility	Measures both probability and utility of a pattern in uncertain databases [73]; the expected support [74] is measured as $expSup(X) = \sum_{i=1}^{\|D\|}(\prod_{X_i \in X} p(X_i, T_q))$.
Affinitive utility	The affinitive utility [75] of a pattern X in T_q is defined as $au(X, T_q) = eu(X) \times af(X, T_q)$, where $eu(X) = \sum_{i_j \in X} pr(i_j)$ and $af(X, T_q)$ denote the affinitive frequency of X in T_q that is $af(X, T_q) = min\{q(i_1, T_q), q(i_2, T_q), \ldots, q(i_j, T_q)\}, i_j \in X$ [75].
Utility occupancy	It depends on the contribution of a unit item. The utility occupancy [76] of a pattern X in T_q and D are defined as $uo(X, T_q) = u(X, T_q)/tu(T_q)$ and $uo(X) = \sum_{X \subseteq T_q \wedge T_q \in D} uo(X, T_q)/\|\Gamma_X\|$, respectively.

of several UPM frameworks were analyzed. Based on the utility theory [25], many evaluation measures of utility have been proposed. Some commonly used evaluation measures of the utility of a pattern in the UPM field are summarized in Table 3.

The most commonly adopted evaluation measure for UPM is the general *utility* concept [42], [52]. It is based on *external utility* (e.g., profit, unit price, risk) and *internal utility* (e.g., quantity). As described in Definition 2, the overall utility of a pattern in the processed database is the cumulative utilities of this pattern in each transaction where it appears in. The *average utility* [72] is divide by the length of pattern, and used to avoid the effect of overall utility increasing with the length of pattern. Expected/potential utility determines both uncertainty and utility of a pattern in uncertain data [73]. Thus, for UPM, this measure is suitable for dealing with uncertain data. Affinitive utility [75] is proposed to address the special task of correlated UPM, but not used for the general task of UPM. The utility occupancy [76] is more suitable than the *utility* concept and *average utility* for discovering the high-utility patterns which have high utility contribution.

3 Basic Approaches for High-Utility Pattern Mining

3.1 Overview of Proposed Categorization

The development of the utility-oriented algorithms has always been an important issue in data mining area. During the past decades, a significant number of utility-oriented algorithms has been proposed to mine utility-based patterns from various types of data (i.e., transaction data [9], sequential data [15], episode data [11], stream data [66], [67], etc). Considering that it is infeasible to go through all existing algorithms within a limited space, in this review we select some representative HUIM algorithms. According to the different mining principles and data structures, Fig. 2 presents a rough overview of techniques to address the UPM problem. Specifically, to facilitate our discussion, we classify these efforts into the following categories: 1) Apriori-based approaches; 2) tree-based approaches; 3) projection-based approaches; and 4) vertical-/horizontal-data-based approaches.

3.2 Apriori-Based Approaches

In 1994, Agrawal and Srikant proposed the well-known downward closure property, also known as the Apriori property [9], which states that all non-empty subsets of a

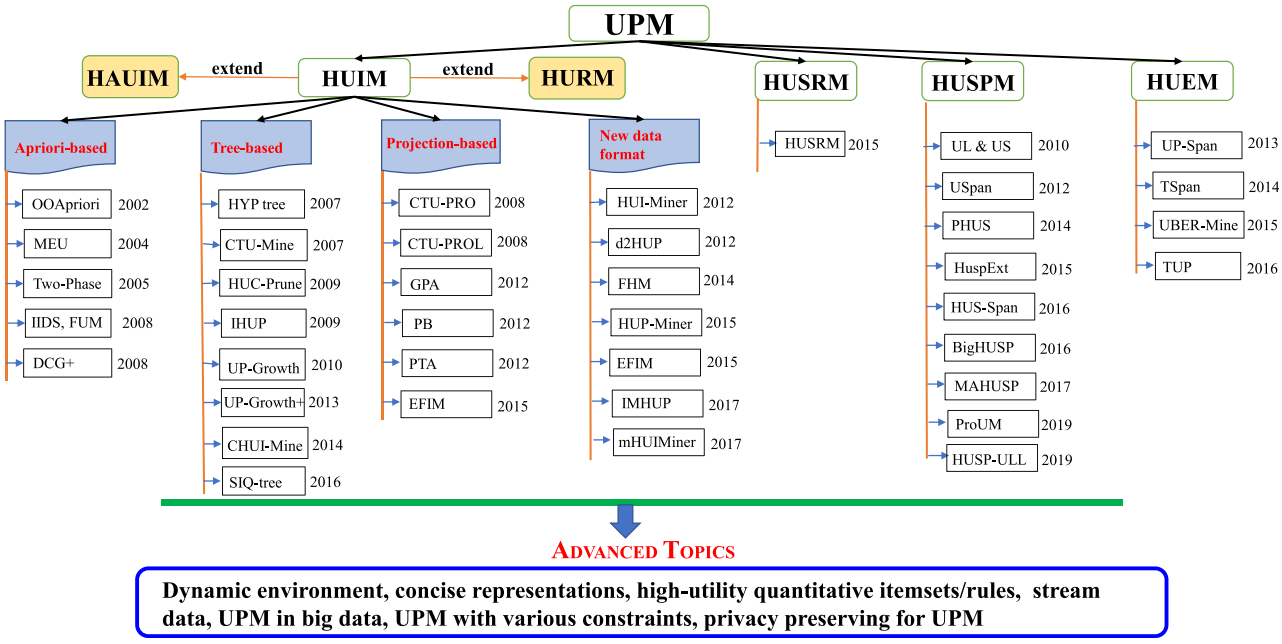


Fig. 2. Taxonomy of UPM algorithms.

TABLE 4
Apriori-Based Algorithms for High-Utility Pattern Mining

Name	Description	Pros.	Cons.	Year
MEU [52]	The first theoretical model and strict definitions of high-utility itemset mining.	MEU uses a heuristic to determine candidates and usually overestimates.	It cannot maintain the downward closure property of Apriori [9], and the derived results are incomplete.	2004
UMining [29] & UMining_H [29]	The general HUIM model with several mathematical properties of the utility measure.	UMining uses the utility upper bound and UMining_H utilizes a heuristic pruning strategy.	It generates a large amount of candidate patterns) and suffers from excessive candidate generations and poor scalability.	2006
Two-Phase [77]	The TWDC property was proposed to discover HUIs in two phases.	It can greatly prune a large amount of candidate patterns.	It has the problem of candidates level-wisely generation-and-test [9], and requires multiple database scans.	2005
IIDS [34], FUM [34], DCG+ [34]	By applying IIDS to ShFSM and DCG, two methods - FUM and DCG+ - were implemented.	For any existing level-wise utility mining method, it can reduce the number of candidates and improve performance.	It has the same performance issues as Apriori [9].	2008

frequent itemset must also be frequent, and any superset of an infrequent itemset cannot be frequent. For example, assuming $\{a, b, c\}$ is frequent, all of its sub-itemsets, such as $\{a, c\}$ and $\{b, c\}$, are also frequent. If $\{d, e\}$ is infrequent, its supersets, such as $\{a, d, e\}$ and $\{d, e, f\}$, are not frequent. Some Apriori-based approaches for HUIM have been further developed. The core step of these Apriori-based UPM algorithms is the generation of candidate k-itemsets C_k from high-utility (k-1)-itemsets (denoted as HUI_{k-1}), and it consists of two operations: join and prune. In join step, the conditional join of two HUI_{k-1} patterns is used to generate candidate set C_k. The prune step then reduces the size of C_k by using the utility upper bound (which is similar to the Apriori property [9]).

• *OOApriori & Top-k Closed Utility Mining* [22], [23]. In 2002, Shen and Yang proposed an objective-oriented association (OOA) mining approach [22]. They integrated the utility constraint into OOApriori (a variant of Apriori [9]) to prune candidates for deriving the OOA rules. The interestingness of OOA rules are measured in terms of probabilities and utilities in supporting the user's objective. The utility constraint for OOA rules is neither monotone nor anti-monotone. In 2003, Chan et al. first defined the concept of utility mining and proposed an objective-directed mining algorithm to mine the top-k closed-utility patterns [23]. This was the first time the term "utility mining" was presented and used to identify both frequent and high-utility itemsets based on business objectives. In this utility-based mining framework, a pruning strategy based on a weak but anti-monotonic condition was developed to reduce search space.

• *Mining with Expected Utility (MEU)* [52]. In 2005, Yao et al. proposed a utility mining model, called mining with expected utility (MEU) [52], which considers both the purchase quantities (called *internal utility*) and unit profits (called *external utility*) of items to mine HUIs. Note that the term "mining high-utility itemsets" first appeared in [23], but their concept and definitions were quite different from the definitions of high-utility itemset mining today. It is widely believed that utility-based itemset mining, sequence mining, and web mining originated in [52]. Researchers in the field of UPM consider the MEU model as the first theoretical model and strict definition of high-utility itemset mining. MEU uses a heuristic to determine candidates and usually overestimates. However, it cannot maintain the downward closure property of Apriori [9], and the derived results are incomplete.

• *UMining and UMining_H* [29]. Yao et al. then proposed UMining and heuristic UMining_H [29] for finding HUIs based on several mathematical properties of the utility measure. The utility constraint is characterized by a property giving the upper bound of the utility value of an itemset. In UMining, the property of utility upper bound is used as a pruning strategy. UMining_H utilizes another pruning strategy based on a heuristic method [29]. However, some HUIs may be erroneously pruned by this heuristic method. Furthermore, neither of them have the downward closure property of Apriori [9], and they overestimate too many patterns. Therefore, they suffer from excessive candidate generation and poor scalability.

• *Two-Phase* [77]. Note that the downward closure property (w.r.t. the Apriori property [9]) of the support measure does not hold for the utility. To address the challenge that the utility measure is neither monotone nor anti-monotone, Liu et al. proposed the well-known Two-Phase algorithm [77]. Two-Phase introduces a novel concept named the transaction-weighted downward closure (TWDC) property (for any itemset X, if X is not a HTWUI, any superset of X is not an HUI) and used it to discover HUIs in two phases. Phase 1: it finds each itemset X such that $TWU(X) \geq minutil$ using the *TWU* upper bound to prune the search space. Initially, it scans a database once to get all 1-itemset $HTWUI_1$; then generates (k+1)-level candidate itemsets (with length k+1) from length-k candidates $HTWUI_k$ (where $k > 0$). For each iteration, it needs to examine the *TWU* values of candidates by scanning the database once. Finally, it is terminated when no candidate can be generated. Phase 2: it scans the database again to calculate the exact utility of each candidate in the set of $HTWUI_k$ and then outputs the desired HUIs.

• *IIDS, FUM, and DCG+* [34]. The itemset share mining problem [60] can be converted to the utility mining problem by replacing the frequency value of each item in every transaction by its total profit (i.e., multiplying the frequency value by its unit profit). The isolated items discarding strategy (IIDS) [34] reduces the number of candidates in every database scan. By discarding isolated items to reduce the number of candidates and to shrink the database scan in each pass, IIDS can improve the level-wise, multi-pass candidate-generation process. Applying IIDS to ShFSM [60] and DCG [60], Fast Utility Mining (FUM) [34] and Direct Candidates Generation (DCG)+ [34] were further developed. The results showed that FUM and DCG+ [60] are better than MEU, UMining, UMining_H and Two-Phase. However, both still suffer from the problem of generating and testing candidates in a level-wise way and require multiple database scans.

Discussions. In summary, all of the early UPM approaches improved on the Apriori work [9]. As shown in Table 4, an important drawback is that all of them need to generate a huge amount of candidates since they rely on a loose upper bound on the utilities of candidates. As a result, these approaches may suffer from long execution times (computationally expensive) and consume a huge amount of memory. Moreover, all these algorithms suffer from the same limitations as Apriori-based ARM algorithms, which are to generate candidates not appearing in the database and to perform multiple database scans to mine the desired information. The computational complexity of these Apriori-based UPM techniques depends on the level-wise manner that generates a huge number of candidates. These techniques may have quadratic complexity if the processed data containing long transactions or a low *minutil* threshold value is used.

3.3 Tree-Based Pattern-Growth Approaches

Many early HUIM approaches perform a level-wise exploration of the search space to find HUIs. To avoid the drawback of an Apriori-based level-wise search, several tree-based HUIM algorithms were then proposed to efficiently mine HUIs based on the TWDC property and the pattern-growth-mining approach. Generally speaking, the tree-based UPM algorithms are inspired by the traditional tree-based FPM algorithms, i.e., FP-Growth [7]. Some mathematical formalism of the pattern-growth method with FP-tree can be referred to [7].

• *HYP Tree [78] and HUC-Prune [79]*. In 2007, Hu et al. proposed an approximation method that identifies the contribution of the predefined utility, objective function, and performance metric, and can take advantage of item attributes [78]. It identifies high-utility combinations and then finds HUIs through a high-yield partition (HYP) tree [78]. In contrast to the traditional FPM and ARM techniques, its goal is to find segments of data and combinations of items/rules that satisfy certain conditions and maximize a predefined objective function. Different from the former UPM approaches, it conducts "rule-discovery" with respect to individual attributes and the overall criterion for the discovered results. It aims at mining groups of patterns that when combined, contribute the most to an objective function [78]. Since the Apriori-like HUIM algorithms suffer from the candidate generation-and-test problem, Ahmed et al. proposed a novel tree-based algorithm named High-Utility Candidate Prune (HUC-Prune). The proposed HUC-tree is a prefix tree storing the candidate items in descending order of *TWU* values. Each node in the HUC-tree consists of the item's name and its *TWU* value. Similar to IHUP [31], HUC-Prune replaces the level-wise candidate generation process by a pattern-growth mining approach. It needs at most three database scans for mining the HUIs, and has better performance than the Apriori-based algorithms.

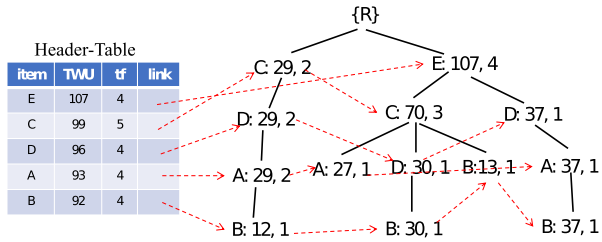


Fig. 3. Example $IHUP_{TWU}$-tree structure (used by IHUP [48]).

• *IHUP (Incremental High-Utility Pattern Mining)* [31]. The tree-based algorithm named IHUP with three tree structures ($IHUP_{PL}$-tree, $IHUP_{TF}$-tree and $IHUP_{TWU}$-tree) was proposed for incremental and interactive high-utility pattern mining [31]. Each node in the IHUP-tree represents an itemset and consists of the itemset's name, a TWU value, and a support count. The IHUP algorithm has three steps: 1) construction of IHUP-tree, 2) generation of HTWUIs [77], and 3) identification of high-utility itemsets. Step 1 is similar to the construction of FP-tree [7] and a complete illustrated example had been given in [31], [47]. In each transaction, the set of $HTWUI_1$ are sorting in *TWU*-descending order and then continuously inserted into the prefix-based IHUP-tree. Fig. 3 shows the constructed IHUP-tree for the example database given in [48]. In step 2, all $HTWUI_k$ are generated from the constructed IHUP-tree using the FP-Growth [7] approach. In step 3, all HUIs and their utilities can be identified from the set of HTWUIs by scanning the database once. Thus, IHUP can avoid generating candidates in a level-wise way. Although IHUP significantly outperforms IIDS [34] and Two-Phase [77], it still produces too many HTWUIs in step 1. Note that both IHUP and Two-Phase use the *TWU* concept to overestimate the utilities of itemsets. Thus, they produce the same huge number of HTWUIs in step 1. Such a large number of HTWUIs substantially degrades the mining performance in terms of execution time and memory consumption. Moreover, the performance of step 2 is affected by the number of HTWUIs. The reason is that the more HTWUIs is generated in step 1, the longer execution time required for mining HUIs in step 2.

• *UP-Growth [80] and UP-Growth+ [48]*. Tseng et al. designed a more compressed utility-pattern tree (UP-tree) and proposed the well-known utility pattern-growth algorithm (UP-Growth) [80] to efficiently mine HUIs. UP-Growth is inspired by the frequency-based FP-Growth method. It integrates four novel strategies, named DLU (Discarding Local Unpromising items), DLN (Decreasing Local Node utilities), DGU (Discarding Global Unpromising items during the construction of a global UP-tree), and DGN (Decreasing Global Node utilities during the construction of a global UP-tree). After two scans of the original database,

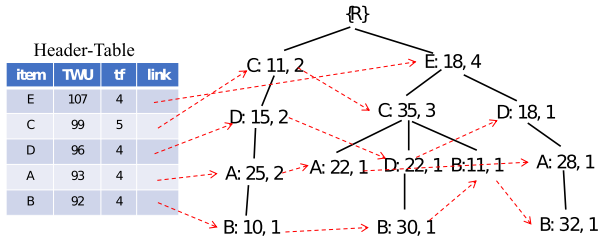


Fig. 4. Example UP-tree structure (used by UP-Growth algorithm [48]).

the UP-tree can be constructed. In the first scan, the utility of each transaction and TWU of each single item are calculated. Discarding global unpromising items, those unpromising items that are not HTWUIs are removed from each transaction, and utilities are eliminated again. Then, the remaining promising items in each transaction are sorted in the descending order of TWU. In the second scan, transactions are inserted into UP-tree by using DGU and DGN strategies. After building the complete UP-tree, as shown in Fig. 4, the potential HUIs (PHUIs) are generated from the global UP-tree with DLU and DLN strategies. In summary, the framework of UP-Growth consists of three steps: 1) scan the database twice to construct a global UP-tree with the DLU and DLN strategies; 2) recursively generate PHUIs from global UP-tree and local UP-trees by UP-Growth with the DGU and DGN strategies; and 3) identify final HUIs from the set of PHUIs. As an improvement of UP-Growth, UP-Growth+ [48] was then developed by utilizing the minimal utilities of each node in each path in the UP-tree. Compared with UP-Growth, the enhanced UP-Growth+ can decrease the overestimated utilities of PHUIs and greatly reduce the number of candidates.

- *CHUI-Mine* [81]. Later, Song et al. proposed the Concurrent High-Utility Itemset Mine (CHUI-Mine) algorithm for mining HUIs by dynamically pruning the CHUI-tree structure. The CHUI-tree is introduced to capture the important utility information of the candidate itemsets. By recording changes in support counts of candidates during the tree construction process, it uses dynamic CHUI-tree pruning. CHUI-Mine makes use of a concurrent strategy, enabling the simultaneous construction of a CHUI-tree and the process for discovering HUIs. Thus, it can reduce the problem of huge memory usage for tree construction and traversal in tree-based HUIM algorithms. Concurrent processes generally interact through the following two mechanisms: shared variables and message passing [81]. Extensive experimental results show that CHUI-Mine is both more efficient and more scalable than Two-Phase [77], FUM [34], and HUC-Prune [79]. However, the faster tree-based algorithms, including IHUP [31], UP-Growth [80], and UP-Growth+ [48] were not all compared.
- *Sum of Item Quantities-Tree (SIQ-Tree)* [82]. Tree construction with a single database scan is significant since a database scan is a time-consuming task. In utility mining, an additional database scan is necessary to identify actual high-utility patterns from candidates. A novel tree structure, namely Sum of Item Quantities tree (SIQ-tree) [82], was developed to capture database information through a single pass. Moreover, a restructuring method is proposed with strategies for reducing overestimated utilities. It can construct the SIQ-tree with only a single scan and decrease the number of candidate patterns effectively with the reduced overestimation utilities, through which mining performance is improved. This approach is faster than IHUP [31] on most datasets, but the faster UP-Growth [80] and UP-Growth+ [48], were not compared.

Discussions. Characteristics and differences of these tree structures are presented in Table 5. In addition, there are various other utility-based mining methods based on tree structures [83], [84]. These tree-based algorithms comprise three steps: 1) construction of trees, 2) generation of candidate HUIs from the trees using the designed pattern-growth approach, and 3) identification of HUIs from the set of candidates. Although these trees are compact, they may not be minimal and still occupy a large memory space. The mining performance is closely related to the number of conditional trees constructed during the entire mining process and the construction/traversal cost of each conditional tree. When using these algorithms on a large database with a low-utility threshold, the storage and traversal costs of numerous conditional trees are high. Thus, one of the performance bottlenecks of these algorithms is the generation of a huge number of conditional trees, which has high time and space costs.

In summary, to address the disadvantages of the Apriori-like UPM algorithm, and to improve efficiency, the advantages of pattern-growth tree-based techniques are as follows: 1) only need two or three passes over dataset; 2) "compresses" datasets into the tree structure; 3) no candidate generation; and 4) much faster than Apriori-like approaches. However, they still have some disadvantages: 1) the constructed tree may not fit in memory; 2) the designed tree is expensive to build; 3) it is time-consuming to recursively process all conditional prefix trees to generate candidates; and 4) the constructed tree is sensitive to the *minutil* parameter.

3.4 Projection-Based Pattern-Growth Approaches

In the past, some of projection-based techniques have been commonly used in data mining, i.e., FP-Growth [7] for FPM and PrefixSpan [16] for SPM. The general idea of projection-pattern mining is to use target items to recursively project the processed database into some smaller projected sub-databases, and then grow the itemset or subsequence fragments in each projected sub-database. To overcome the disadvantages of the tree-based HUIM approaches, some projection-based techniques have been developed for UPM. Some basic mathematical formalism of projection and projected sub-databases are skipped here, and details can be referred to [16], [85].

- *CTU-PRO and CTU-PROL* [33]. In 2007, Erwin et al. first proposed a projection-based CTU-PRO algorithm for HUIM. It mines HUIs by bottom-up traversal of a compressed utility pattern tree (CUP-tree) [33], which is a variant of CTU-tree [83]. The mining of a subdivision from CUP-tree consists of three steps: 1) Construction of ProItem-Table, 2) construction of ProCUP-tree, and 3) mining by Pro-Cup-tree traversal. CTU-PRO creates a GlobalCUP-tree from the transaction database after identifying the individual HTWUIs [77] with the concept of TWU. For each HTWUI, a smaller projection tree called the LocalCUP-tree is extracted from the GlobalCUP-tree for returning all high-utility itemsets with that item as prefix. CTU-PRO constructs parallel sub-divisions on disk that can be mined

TABLE 5
Tree-Based Pattern-Growth Algorithms for High-Utility Pattern Mining

Name	Description	Pros.	Cons.	Year
HYP tree [78]	An approximation method identifies the utility contribution.	It can find segments of data through combinations of few items/rules.	The built HYP tree is huge, and the memory is costly.	2007
CTU-Mine [83]	A pattern-growth approach based on a compact data representation named CTU-tree for utility mining.	The pattern growth, which avoids candidate generation-and-test, is suitable for dense data.	The CTU-tree is complex and stores too much information, which may consume much memory.	2007
HUC-Prune [79]	High-utility candidate prune without level-wise candidate generation-and-test.	It replaces the level-wise candidate generation process by a pattern-growth mining approach.	The upper bound is high, and the constructed tree is huge.	2009
IHUP [31]	A tree-based approach for incremental and interactive high-utility pattern mining.	The IHUP-tree is more compact than previous trees, and IHUP is significantly faster than IIDS and Two-Phase.	It still uses the TWU and produces too many HTWUIs in phase 1.	2009
UP-Growth [80]	The utility pattern growth algorithm with more compressed utility pattern tree (UP-tree).	UP-tree is more compact than IHUP-tree, and the strategies are powerful to reduce the number of candidates.	It is time-consuming to recursively process all conditional prefix trees for candidate generation.	2010
UP-Growth+ [48]	An improved version of UP-Growth with two pruning strategies.	The enhanced UP-Growth+ can decrease the overestimated utilities of PHUIs and greatly reduce the number of candidates.	It is time-consuming to recursively process all conditional prefix trees for candidate generation.	2013
CHUI-Mine [81]	Concurrent High-Utility Itemset Mine (CHUI-Mine) by dynamically pruning the CHUI-tree.	Two mechanisms, shared variables and message passing, can make the CHUI-tree more compact.	Recursively processing all conditional prefix trees is time-consuming.	2014
SIQ-tree [82]	Sum of Item Quantities.	Constructs the SIQ-tree with only a single scan and decreases the number of candidate patterns.	Recursively processing all conditional prefix trees is time-consuming.	2016

independently. The performance of CTU-PRO is better than Two-Phase [77] and CTU-Mine [83]. CTU-PROL introduces two new concepts, compressed transaction utility-prol and CUP-tree, which are used for parallel projection of the transaction database. Note that the anti-monotone property of TWU is used to prune the search space of sub-divisions in CTU-PROL. However, unlike Two-Phase, it avoids a rescan of the database to calculate the actual utilities of HTWUIs. The results show that CTU-PROL outperforms Two-Phase [77] and CTU-Mine [83].

- *GPA and PB* [85]. Since the tree-based pattern-growth approaches recursively perform tree traversal and generate a series of sub-tree structures, Lan et al. proposed two alternative efficient projection-based utility mining approaches, named Gradual Pruning Approach (GPA) [85] and PB (Projection-Based mining approach) [85]. Compared with the level-wise techniques, the property of a projection-based technique is more suitable for improving the utility upper bound. The general idea is to use the overestimated HTWUIs [77] to recursively project item/sequence databases into some smaller projected databases and grow item/subsequence fragments in each projected sub-database. In addition, PB applies a novel pruning strategy and an indexing mechanism to speed up the runtime and reduce the memory requirement of the mining process. The indexing mechanism imitates traditional projection algorithms (i.e., PrefixSpan [16]) by projecting sub-databases. Using projection, GPA and PB can significantly reduce database size when deriving larger itemsets and outperform Two-Phase [77].
- *PTA* [85]. Different from PB [85] and GPA [85], *pruning* and *filtering* strategies are proposed to tighten the upper bounds of utility values in the projection-based upper-bound tightening approach (abbreviated as PTA). The framework of PTA includes the following: 1) finds HTWUIs and high-utility 1-itemsets; 2) performs the pruning strategy and the indexing strategy; 3) projects transactions required by the prefix itemsets to be processed; and 4) finds k-HTWUIs and high-utility k-itemsets. An effective index mechanism is applied to reduce the time cost of searching relevant transactions that need to be projected in sub-databases. Thus, PTA only needs one database scan. Through experiments, the results show that PTA outperforms the other existing algorithms (i.e., Two-Phase [77], GPA [85], PB [85], CTU-PRO [33], IHUP_{PL} [31], IHUP_{TWU} [31], and IHUP_{TF} [31]) in terms of pruning unpromising itemsets, memory usage, and runtime, respectively.

Discussions. In summary, the above UPM approaches, which utilize the database projection mechanism, have the following advantages: 1) mine the complete set of high-utility patterns but reduce the effort of candidate generation; 2) prefix-projection reduces the size of the projected sub-database and leads to efficient processing; and 3) bi-level projection and pseudo-projection may improve mining efficiency, as summarized in Table 6.

3.5 New Data-Format-Based Approach

To achieve more efficiency than the tree-based UPM approaches, some algorithms that mine high-utility itemsets using a vertical or horizontal data structure with a single

TABLE 6
Projection-Based Pattern-Growth Approaches for UPM

Name	Description	Pros.	Cons.	Year
CTU-PRO [33] & CTU-PROL [33]	Two projection-based algorithms with Compressed Utility Pattern-tree (CUP-tree).	They construct parallel sub-divisions on disk that can be mined independently and have good performance.	TWU is adopted as the upper bound, which generates many redundant candidates.	2008
GPA and PB [85]	Two projection-based mining approaches, GPA (Gradual Pruning Approach) and PB (Projection-Based mining approach).	Using projection, they can speed up the runtime and reduce database size when deriving larger itemsets.	TWU is adopted as the upper bound, which generates many redundant candidates.	2012
PTA [85]	A projection-based upper-bound tightening approach.	Two effective strategies, named pruning and filtering, are proposed to tighten the upper bounds of utility values.	The projection of sub-databases is sometimes time-consuming.	2012

TABLE 7
Utility-List-Based Algorithms for High-Utility Pattern Mining

Name	Description	Pros.	Cons.	Year
HUI-Miner [86]	The first one-phase model to mine high-utility itemsets.	It first introduced the concept of the remaining utility and the vertical data structure w.r.t. utility-list.	The join operations between utility lists of (k+1)-itemsets and k-itemsets is time-consuming.	2012
d2HUP [88]	Another algorithm that can directly discover HUIs without maintaining candidates.	It efficiently obtains the utility of each enumerated itemset and the upper bound on utilities using the CAUL.	The tree structure and CAUL consume more memory.	2012
FHM [87]	An improved version of HUI-Miner with a pruning strategy named EUCP.	It not only has the advantages of HUI-Miner but also reduces the join operations between utility lists.	It consumes slightly more memory than HUI-Miner and has poor performance on dense datasets.	2014
HUP-Miner [89]	An improved version of HUI-Miner with two new pruning strategies (PU-Prune and LA-Prune).	The two new pruning strategies can reduce the join operations between utility lists.	It needs to explicitly set the number of dataset partitions, while these partitions cannot always improve the efficiency.	2015
EFIM [90]	Uses projection and transaction-merging techniques for reducing the cost of database scans.	It consumes less memory, and its complexity is roughly linear with the number of items in the search space.	Sometimes the recursive projection is time-consuming and uses a lot of memory.	2015
IMHUP [91]	A novel utility-list-based algorithm for HUPs mining without any candidate generation.	It uses the indexed utility list to reduce the join operations between utility lists.	The upper bound on utilities is not tight enough.	2017
mHUIMiner [92]	An efficient one-phase algorithm that combines some ideas from HUI-Miner and IHUP.	It utilizes utility list and remaining utility and performs well on sparse datasets.	It still suffers from some problems similar to those of HUI-Miner and IHUP.	2017

phase were proposed recently, such as HUI-Miner [86], FHM [87], d2HUP [88], HUP-Miner [89], and EFIM [90]. Both d2HUP and EFIM use a horizontal database, while others use the vertical data structure. All these algorithms cannot only avoid the disadvantages of Apriori-based approaches but also avoid the disadvantages of the tree-based HUIM approaches. Details are shown in Table 7 and described below.

• *HUI-Miner [86] and FHM [87].* High-Utility Itemset Miner (HUI-Miner) [86] is the first one-phase algorithm to discover HUIs. It proposes a vertical data structure named utility-list [86] and the concept of *remaining utility* [86], which have been widely extended in many other newly UPM algorithms. As a compact data structure, utility-list can store utility information for the potential patterns that may have high utility value. The utility-list of an itemset X in a database D is a set of tuples corresponding to the transactions in which X appears. Each tuple is defined as $< tid, iu, nu >$ for every transaction T_q containing X, in which the *tid* element is the transaction identifier of T_q, the *iu* element is the utility value of X in T_q, and *ru* element is the *remaining utility* value of X in T_q. More details about the *remaining utility*, utility-list structure, and its construction can be referred to [86]. The construction process of utility-list is quite efficient and consumes little memory. By keeping necessary information from the transaction database in memory, HUI-Miner can directly mine HUIs by spanning the search space w.r.t. a set-enumeration tree [93]. As an enhanced version of HUI-Miner [86], Fast High-Utility Miner (FHM) [87] utilizes a novel pruning strategy named Estimated Utility Co-occurrence Pruning (EUCP) to reduce the costly join operations of utility-lists. EUCP is based on the Estimated

Utility Co-Occurrence Structure (EUCS) [87]. Using utility-list [86], HUI-Miner and FHM need only two database scans to construct a series of utility-lists of $HTWUI_1$. Then, utility-lists of (k+1)-itemsets can be obtained by performing the join operations of utility-lists of k-itemsets. They can directly discover HUIs by keeping utility-list in memory, and utilizes the upper bound of the remaining utility. HUI-Miner and FHM outperform than the all previous algorithms on most datasets, in terms of running time (almost two orders of magnitude faster) and memory cost. FHM is more faster than HUI-Miner [86], especially for dense databases, but not efficient for databases that are sparse. However, the drawback is that both of them need to perform costly join operations among a series of utility-lists, which can be time costly. Note that some quantitative results are already reported on the same benchmark datasets in [86], [87].

• *d2HUP* [88]. d2HUP is also able to directly discover HUIs without candidate generation. It utilizes another novel data structure, named Chain of Accurate Utility Lists (CAUL) [88] to store the necessary information. In contrast to HUI-Miner, it enumerates an itemset as a prefix extension of its prefix itemset. In fact, the search space of d2HUP is a variant of set-enumeration tree [93]. It can efficiently calculate the utility of each enumerated itemset and the upper bound on utilities of the prefix-extended itemsets. In fact, d2HUP also utilizes the similar concept of *remaining utility* to tighten the utility upper bound, which is much tighter than TWU. This upper bound is tightened by iteratively filtering out irrelevant items when constructing CAUL. More specifically, it requires less memory than different kinds of tree structures used in the above-mentioned algorithms. d2HUP was shown to be more efficient than Two-Phase [77], UP-Growth [80], and HUI-Miner [86], but the performance was not compared with some recent algorithms, such as FHM [87] and HUP-Miner [89].

• *HUP-Miner*. HUP-Miner [89] is an improvement algorithm based on HUI-Miner [86]. Two new pruning strategies, PU-Prune (based on dataset partition) and LA-Prune (based on the concept of *lookahead pruning*), are introduced in HUP-Miner to limit the search space for mining HUIs [89]. It needs to set the number of dataset partitions K, which determines how many partitions processed internally. However, the optimal value of K is hard to find empirically for a given dataset. Based on the concept of remaining utility [86], LA-Prune provides a tighter utility upper bound of any k-itemset. Thus, a huge number of unpromising k-itemset ($k \geq 2$) that have low utility can be pruned. It has been shown that HUP-Miner is significantly faster than HUI-Miner. In fact, the PU-Prune strategy based on dataset partition does not always have an effect on runtime and memory consumption. In addition, a shortcoming is that the number of partitions is required to be set explicitly by users, since it is an additional parameter.

• *Index High-Utility Itemsets Mine (IHUI-Mine)* [94]. As mentioned before, these candidate generation-and-test approaches suffer from the drawbacks of having an immense candidate pool and requiring several database scans. Meanwhile, methods based on pattern growth tend to consume large amounts of memory to store conditional trees. IHUI-Mine uses the subsume index [95], a data structure for efficient frequent itemset mining, to enumerate the desired HUIs and prune the search space. The experimental results show that IHUI-Mine outperforms some popular algorithms, including Two-Phase [77], FUM [34], and HUC-Prune [79], but it has not been compared with the state-of-the-art algorithms.

• *IMHUP* [91]. In the framework of list-based high-utility pattern mining, there are a number of comparison and join operations of entries within lists causing enormous execution time costs. Based on the indexed utility-list (IU-list) [91], two techniques were developed in Indexed list-based Mining of High-Utility Patterns (IMHUP) to reduce utility upper-bounds that satisfy the anti-monotonic property. IMHUP-RUI and IMHUP-CHI [91] generate high-utility patterns without any construction of additional local-lists when the current lists only contain information of the same revised transactions. They further utilize the upper-bound utilities in IU-lists to decrease the search space.

• *EFIM* [90]. The projection-based EFficient high-utility Itemset Mining (EFIM) algorithm introduces several new ideas, including two new upper bounds named revised *subtree utility* and *local utility*, and a array-based utility computing technique. To reduce the cost of database scans, EFIM further proposes the database projection and transaction merging techniques named High-utility Database Projection (HDP) and High-utility Transaction Merging (HTM). As larger itemsets are explored, both projection and merging reduce the size of the database. The main ideas of HDP and HTM are described in [90]. The time and space complexity of EFIM is roughly linear with the number of distinct items in the search space. The competitive results show that EFIM is in general 2 to 3 orders of magnitude faster than the state-of-the-art algorithms (UP-Growth+ [48], HUI-Miner [86], FHM [87], d2HUP [88], and HUP-Miner [89]) on dense datasets and performs quite well on sparse datasets.

• *mHUIMiner*. mHUIMiner [92] is a hybrid algorithm that combines some ideas from HUI-Miner [86] and IHUP-tree [31]. It adopts the utility-list and remaining utility. It utilizes a tree structure to guide the itemset expansion process, and thus the itemsets that are nonexistent in the database can be avoided. Unlike current techniques, it does not have a complex pruning strategy that requires expensive computational overhead. It was shown to well perform on sparse datasets, and provide the best runtime on sparse datasets, while having a comparable performance than other state-of-the-art algorithms (e.g., HUI-Miner [86], FHM [87], and EFIM [90]) on dense datasets.

Discussions. All the algorithms discussed in this subsection utilize the new data structure to store necessary information about each itemset. By spanning the search space w.r.t. a set-enumeration tree [93], they can easily calculate the total utility of an itemset by performing join operations of the built utility-lists. Moreover, an upper bound on the overall utilities of itemsets called the remaining utility is calculated using utility-lists. It can be used to determine if each pattern and its extensions are not high-utility itemsets (to reduce the search space). The upper bound with remaining utility is equivalent to the upper bound proposed in d2HUP [88]. Although a pattern-growth approach in d2HUP can avoid considering itemsets not appearing in the database, the used hyper-structure still consumes a considerable amount of memory [88]. Some competitive results of these

UPM methods have been compared and summarized in recent studies [90], [92].

As shown at Table 7, the HUIM algorithms are based on a combination of vertical or horizontal data formats and typical approaches. These hybrid algorithms combine different techniques to mine high-utility patterns in such a way that the strengths of each technique are utilized to maximize their efficiency. The properties of these one-phase algorithms are as follows:

1) *Complete result*: The completeness is guaranteed as the traversal of the search space w.r.t. set-enumeration tree.
2) *Stable result*: The result is stable as all exact utility information is stored in a vertical or horizontal data structure. Depth-first searching is also used to quickly calculate the utilities.
3) *Efficiency*: The algorithm is efficient relative to algorithms that traverse the complete search space. Moreover, the sort order of items in set-enumeration tree affects the mining efficiency, but not the final mining results of patterns.
4) *Parameter sensitivity*: These algorithms, except for HUP-Miner, only have *minutil* as the parameter, and are sensitive to it.

4 Advanced Topic of UPM

4.1 Mining High Average Utility Itemsets

A main challenge in HUIM is that the exponential search space for HUIM is extremely large when the number of distinct items or the size of the database is too large. The other challenge is that existing HUIM methods overlook the fact that longer itemsets result in higher utility values. A large itemset may have an unreasonable estimated profit as opposed to its actual value. Therefore, the concept named high average-utility itemset mining (HAUIM) is proposed [72]. HAUIM discovers utility patterns by considering both their utilities and lengths, thus providing a different utility measure than traditional HUIM. HAUIM divides the utility of an itemset by its length (the number of items that the itemset contains). Up to now, some interesting works have been extensively studied, such as Apriori-based algorithms [72], projection-based PAI [96], utility-list based HAUI-Miner [97], [98], and other hybrid algorithms with different upper-bound models [98], [99].

4.2 HUIM in Dynamic Environments

In a wide range of applications, the processed data may be commonly dynamic but not static. The dynamic data are more complicated and difficult to handle than the static data. Most algorithms process a static database to mine HUIs. In real-world applications, records/transactions are dynamically changed (i.e., inserted, deleted, and modified) in the original database. Some preliminary studies have been done on this issue for UPM.

• *Case 1: HUIM with Record Insertion*. Data mining is an iterative process, and incremental data mining [100], [101] provides the ability to continuous analyze and mine the data by using previous data structure and mining results. Up to now, some incremental models have been developed for mining HUIs with record insertion, such as IHUP [31], FUP-HUI-INS [102], PRE-HUI-INS [103], HUI-list-INS [104], and EIHI [105]. Among these, the early algorithms, e.g., FUP-HUI-INS and PRE-HUI-INS, utilize the utility-oriented dynamic maintain strategies that are extended by the original FUP [100] and pre-large [101] concepts. Since FUP-HUI-INS and PRE-HUI-INS algorithms are processed by a Two-Phase model, an additional database rescan is still necessary to find the actual HUIs. Furthermore, computations are required to find the HTWUIs based on the pattern-growth approach. Both HUI-list-INS and EIHI utilize the utility-list [86] and utility property to significantly reduce runtime and memory usage. More complete reviews can be referred to [47].

• *Case 2: HUIM with Record Deletion*. In practical situations, record deletion is also an important issue in databases. Cheung et al. designed the FUP2 concept [106] to discover frequently updated itemsets for record deletion. Hong et al. developed the pre-large concept [101] for handling record deletion to avoid a multiple database scan each time. Two support thresholds are separately set in pre-large [101], and thus the original database is not required to be scanned until the number of accumulative deleted transactions achieves the designed safety bound. Since the FUP2 concept [106] cannot be directly applied to the HUIM, Lin et al. separately designed the FUP-HUI-DEL [107] and PRE-HUI-DEL [108] algorithms for handling record deletion to maintain and update the new HUIs based on the Two-Phase model. Recently, an efficient dynamic algorithm named HUI-list-DEL [109] was developed to discover HUIs by maintaining the built utility-list [86] structure for record deletion in dynamic databases. The new HUIs can be directly produced without candidate generation or numerous database scans.

• *Case 3: HUIM with Record Modification*. As one of the three common operations (record insertion, deletion, and modification) in databases, record modification is also commonly seen in real-life situations. For example, some typos or errors may occur when the collected data from periodic transactions is input into a computer using a keyboard. Thus, some information may become invalid or new information may arise. Lin et al. first proposed the FUP-HUP-tree-MOD algorithm [110] to address this issue. It is based on the FUP concept [100] and shows better performance compared to Two-Phase and some tree-based algorithms in batch mode. In addition, a faster PRE-HUI-MOD algorithm [111] extends the pre-large concept [101] to set the effective upper bound for discovering HTWUIs and HUIs from the dynamic databases.

4.3 Concise Representations of Utility Patterns

In the field of FPM, many techniques have been devised to derive compact representations of frequent patterns that eliminate redundancy but have rich information, such as free sets [112], non-derivable sets [113], maximal itemsets [114], and closed itemsets [115]. These representations significantly reduce the number of extracted frequent patterns, but some lead to loss of information (e.g., maximal itemsets [114]). Although the above UPM methods perform well in some cases, their performance may degrade when the minimum utility threshold is low. A large number of HUIs and candidates lead to long execution times and huge memory consumption. When computing resources are limited, this is a serious problem for the mining task. However, a large

amount of HUIs is difficult to comprehend and be analyzed by users. Thus, it is often impractical to generate and return the entire set of HUIs.

• *Maximal High-Utility Pattern*. To return representative HUIs to users, some concise representations of HUIs were proposed. Chan et al. introduced the concept of a utility frequent closed pattern [23], the definition of which is different from high-utility itemset [48], [52]. Shie et al. then proposed a new representation called maximal high-utility itemset in which a HUI is not a subset of any other HUI [69]. Although maximal HUI reduces the number of extracted HUIs, it is not lossless because the utilities of the subsets of a maximal HUI cannot be known without rescanning the database. Moreover, recovering all HUIs from the set of maximal HUIs is very inefficient since many subsets of a maximal HUI may have low utility.

• *Closed High-Utility Pattern*. To provide not only compact but also complete information about high-utility itemsets to users, Tseng et al. first addressed the problem of redundancy in high-utility itemset mining [116]. A lossless and compact representation named closed high-utility itemset [116] was introduced. To mine this representation, they proposed three algorithms named AprioriHC (Apriori-based approach for mining High-utility Closed itemsets), AprioriHC-D (AprioriHC algorithm with Discarding unpromising and isolated items), and Closed High-Utility Itemset Discovery (CHUID) [116]. Fournier-Viger et al. then proposed a fast and memory efficient algorithm named EFIM-Closed [117] to discover closed HUIs by extending the EFIM model [90]. It proposes three strategies to mine CHUIs efficiently: closure jumping, forward closure checking, and backward closure checking. EFIM-Closed relies on two new upper bounds, named local utility and sub-tree utility, to prune the search space, and it can calculate these upper bounds efficiently. Inspired by utility-list [86], some more efficient one-phase algorithms have been proposed to address this interesting issue, such as CHUI-Miner [118] and CHUM [119].

4.4 Mining High-Utility Quantitative Itemsets/Rules

Although extensive studies have been proposed for high-utility itemset mining, a critical limitation of these studies is that they ignore the quantity attribute of items in discovered HUIs. However, such information can be very useful and valuable in many applications. In view of this, the concept of High-Utility Quantitative Itemset mining (abbreviated as HUQI) [38], [120] has emerged. In the framework of HUQI mining, an item may have different quantities in the database and each item carrying a different quantity is regarded as a quantitative item. HUQI [38] and more efficient vertical utility-list-based VHUQI [120] were thus developed. An example of such a rule is $(bread, 3, 4) \Rightarrow (milk, 2, 3)$, which means that most customers who purchased three or four breads also purchased two or three milks. We can use this information to package products with quantities that have high utility and estimate the number of items that need to be reserved according to the number of other items.

4.5 High-Utility Sequential Pattern Mining

By integrating the utility factor and sequence data, the problem of high-utility sequential pattern mining (HUSPM) was introduced. For handling the utility of web log sequences, two tree structures, called utility-based WAS tree (UWAS-tree) [63] and incremental UWAS-tree (IUWAS-tree) [63], were developed to mine web access sequences. However, a sequence element with multiple items, such as $[(a, 3)(c, 4)]$, cannot be supported in these two models. The considered scenarios are rather simple, which limits their applicability for handling complex sequences. To this end, some algorithms were proposed to address the HUSPM problem.

• *UL and US* [41]. Since both UWAS-tree and IUWAS-tree algorithms cannot deal with sequences containing multiple items in each sequence element (transaction), Ahmed et al. designed two algorithms (level-wise Utility-Level (UL) [41] and pattern-growth Utility-Span (US) [41]) to mine HUSPs. UL and US extend traditional sequential pattern mining (SPM). The utility of a sequential pattern is calculated in two ways. The utilities of sequences having only distinct occurrences are added together, while the highest occurrences are selected from sequences with multiple occurrences and used to calculate the utilities. However, the problem definition in UL and US [41] is rather specific. No generic framework for transferring from SPM to high-utility sequence analysis has been proposed.

• *USpan* [42]. Yin et al. then formalized the problem of HUSPM, and proposed a generic framework and the USpan algorithm to mine high-utility sequences. [42]. A lexicographic quantitative sequence tree (LQS-tree) is constructed as the search space. Two concatenation mechanisms, *I-Concatenation* and *S-Concatenation*, are used to generate newly concatenated utility-based sequences. Based on the LQS-tree structure, USpan [42] adopts the sequence-weighted utilization (SWU) measure and the Sequence Weighted Downward Closure (SWDC) property to prune unpromising sequences and to improve the mining performance. However, a shortcoming of USpan is that the data representation w.r.t. the utility matrix is quite complex and memory-costly.

• *PHUS* [121]. Lan et al. then proposed the projection-based high-utility sequential pattern mining (PHUS) algorithm for mining HUSPs with the maximum utility measure and a *sequence-utility upper-bound* (SUUB) model [121]. The algorithm extends PrefixSpan [16] and uses a projection-based pruning strategy to obtain tight upper bounds on sequence utilities. Thus, it can avoid considering too many candidates, and improves the performance of mining HUSPs using the SUUB model.

• *HuspExt* [122] and *HUS-Span* [125]. Alkan et al. [122] designed the high-utility sequential pattern extraction (HuspExt) algorithm with an upper-bound called Cumulate Rest of Match (CRoM). It uses a pruning before candidate generation (PBCG) strategy to prune unpromising sequences for mining HUSPs. However, HuspExt cannot discover the complete HUSPs due to the incorrect upper bound. In view of the previous upper bounds on sequence utilities not being tight enough, HUS-Span [125] utilizes two tight utility upper bounds, called prefix extension utility (PEU) and reduced sequence utility (RSU), as well as two companion pruning strategies, to identify high-utility sequential patterns.

• *ProUM* [123]. Gan et al. [123] proposed an efficient projection-based utility mining approach named ProUM to discover high-utility sequences by utilizing the upper bound named sequence extension utility (SEU) and the utility-array structure [123]. Different from the upper bound

TABLE 8
Algorithms for High-Utility Sequential Pattern Mining

Name	Description	Pros.	Cons.	Year
UWAS-tree & IUWAS-tree [63]	The specific algorithms designed for mining utility web log sequences.	The first paper that integrates utility measure into sequential pattern mining.	The considered scenarios are rather simple, which limits their applicability for complex sequences.	2010
UL & US [41]	A level-wise Utility-Level (UL) algorithm and a pattern-growth Utility-Span (US) algorithm.	They first extend SPM to HUSPM, and introduce the calculation of utility in a sequence.	The problem definition is rather specific. No generic framework is proposed.	2010
USpan [42]	First formalizes the problem of HUSPM, and proposes a generic framework.	The width and depth pruning methods substantially reduce the search space in the LQS-tree.	Data representation w.r.t. utility matrix in USpan is quite complex and memory-costly.	2012
PHUS [121]	A projection-based algorithm for mining HUSPs with the sequence-utility upper-bound (SUUB).	The projection mechanism and upper-bound SUUB model can avoid considering too many candidates.	The upper bound on sequence utility is not tight enough, and consumes longer runtime.	2014
HuspExt [122]	Based on the upper-bound CRoM, it utilizes a Pruning Before Candidate Generation (PBCG) strategy.	PBCG strategy prunes some unpromising sequences for mining HUSPs.	The CRoM upper-bound on sequence utility produces the incomplete set of HUSPs.	2015
HUS-Span [122]	Two upper-bounds (PEU and RSU) and utility-chain are proposed in this method.	PEU is more tighter than previous upper-bounds, and utility-chain is easily implemented.	It may consume longer runtime than USpan on some datasets.	2016
ProUM [123]	Using a new upper-bound named SEU and the array-based structure named utility-array.	SEU is a correct upper-bound and the projection-based pruning strategies are powerful to prune unpromising sequences.	The SEU upper bound is still an over-estimated bound; the search space can be further reduced.	2019
HUSP-ULL [124]	Based on the PEU upper-bound, it utilizes the utility-linked (UL)-list structure and two pruning strategies to prune candidates.	It outperforms the state-of-the-art algorithms for mining HUSPs, in terms of execution time and memory consumption.	The proposed UL-list structure is a complicated structure for implementation.	2019

used in USpan, SEU can guarantee the correctness and completeness of discovered results on sequence data. Besides, ProUM has better performance up to two orders of magnitude in terms of execution time on most sequence datasets than USpan and HUS-Span.

• *HUSP-ULL* [124]. The state-of-the-art HUSP-ULL [124] algorithm utilizes a new data structure namely utility-linked (UL)-list and two pruning strategies (called look ahead strategy and irrelevant item pruning strategy) to fast discover HUSPs. According to the extensive experiments [124], it shows that HUSP-ULL is the fastest when comparing to the current HUSPM algorithms. Some competitive results of these recent state-of-the-art HUSPM methods have been compared and summarized in the studies of ProUM [123] and HUSP-ULL [124].

Main characteristics of these HUSPM algorithm are summarized in Table 8. In addition, Shie et al. explored a new problem of mining high-utility mobile sequential patterns (HUMSPs) by integrating mobile data mining with utility mining [35], [36]. This is the first work that combines mobility patterns with utility factor to find high-utility mobile sequential patterns.

4.6 High-Utility Episode Mining

When the sequential data becomes an event sequence, the task of frequent episode mining (FEM) [11] is introduced. FEM reveals a significant amount of useful information hidden in the event sequence with a wide range of applications [11], [12], [13], [14]. However, the discovered frequent episode is still too simple and primitive. In some cases, FEM may lose some rich information, such as utility, important, risk, etc. Wu et al. [43] presented the first attempt to solve the problem of high-utility episode mining (HUEM) in a complex event sequence. However, the proposed UP-Span algorithm suffers from low efficiency in both runtime and memory consumption. Furthermore, the proposed upper-bound named Episode Weighted Utility (EWU) is a loose and basic utility bound for episodes. Guo et al. then proposed the TSpan algorithm with several improvements for UP-Span in a much more efficient manner [126], which can save considerable search space and runtime. Then, Lin et al. separately introduced some models to process complex event sequences and stock investment using high-utility episode mining and a genetic algorithm [44], [127]. In addition, the top-k issue of HUEM has been studied recently [128].

4.7 UPM in Big Data

In the big data era, it requires more efficient frameworks of UPM to handle the big data issue. Several models are presented to address UPM in big data [129], [130], [131]. Details are described below.

• *UPM in Big Itemset Data*. PHUI-Growth (Parallel mining High-Utility Itemsets by pattern-Growth) [129] is first proposed for parallel mining HUIs on Hadoop platform. It adopts the MapReduce [132] architecture to partition the whole mining tasks. As a distributed parallel algorithm, PHUI-Miner with a sampling strategy is introduced by Chen et al. [130]. It extracts the approximate HUIs from big data. Recently, the study of parallel mining of top-k HUIs in Spark in-memory computing architecture is further proposed. It inherits several advantages of Spark [133].

• *UPM in Big Sequence Data*. The BigHUSP model is the first work to discover distributed and parallel high-utility sequential patterns [131]. BigHUSP uses multiple steps of MapReduce [132] to process big data in parallel. In contrast to the traditional HUSPM approaches, it can deal with large-scale sequential data. MAHUSP [134] is a memory-adaptive approximation algorithm to efficiently discover high-utility sequential patterns over data streams. It employs a memory-adaptive mechanism using a bounded portion of memory, and guarantees that all HUSPs are discovered under certain circumstances. Experimental study shows that MAHUSP can not only discover HUSPs over data streams efficiently, but also adapt to memory allocation without sacrificing much of the quality of discovered HUSPs.

4.8 UPM in Stream Data

A data stream is an infinite sequence of data elements continuously arriving at a rapid rate [66], [67]. Mining useful patterns from data streams has become one of interesting problems of data mining [67], [135], [136]. However, few works on mining data streams consider the utility factor embedded in data streams. Tseng et al. first proposed the THUI-Mine (Temporal High-Utility Itemsets) model to mine temporal HUIs from data streams [137]. THUI-Mine can effectively identify the temporal HUIs by generating fewer temporal 2-itemsets of HTWUIs. Thus, the execution time can be reduced significantly in mining all HUIs from data streams. In this way, the discovery process under all time windows of data streams can be achieved with limited memory space and less candidates. Then, researchers for HUIM proposed several stream mining models, such as Mining High-Utility Itemsets based on BITvector (MHUI-BIT) [30], Mining High-Utility Itemsets based on TIDlist (MHUI-TID) [30], and Generation of maximal high-Utility Itemsets from Data strEams (GUIDE) [32], [69]. GUIDE is a framework that mines the compact maximal HUIs from data streams with different models (i.e., the landmark, sliding, and time fading window models) [32], [69]. In [70], the high-utility stream tree (HUS-tree) and HUPMS algorithm (high-utility pattern mining over stream data) are proposed for incremental and interactive UPM over data streams with a sliding window.

4.9 UPM with Various Interesting Constraints

Up to now, most of the algorithms for UPM have been developed to improve the efficiency of the mining process, while effectiveness of the algorithms for UPM is also very important, because it is related to its usefulness for various data, constraints, and applications. Researchers in the field of utility-oriented pattern mining have proposed many algorithms and models to extend effectiveness. Many constraint-based UPM algorithms have been extensively developed for various problems, targeting a wide range of applications. For example, mining high-utility patterns with products' on-shelf time period [85], [138], mining the up-to-date HUIs that reflect recent trends [139], [140], mining discriminative high-utility patterns [75], [141], mining top-k high-utility patterns without setting the minimum utility threshold [68], [142], UPM with multiple minimum utility thresholds [143], utility-based association rule mining [39], [40], UPM with consideration of various discount strategies [61], UPM by considering negative utility values [144], [145], UPM from uncertain data [73], [146], and extracting non-redundant correlated HUIs [62], [147]. Obviously, UPM with various interesting constraints is an active research topic.

4.10 Privacy Preserving for UPM

Since more useful information is in the expected utility-based patterns than in that of the frequent itemsets or sequences, privacy preserving for high-utility pattern mining (PPUM) is more realistic and critical than privacy-preserving data mining (PPDM) [148], [149], [150], [151]. Some preliminary studies have been done on this issue. Yeh et al. first designed two models, named Hiding High-Utility Itemset First (HHUIF) and Maximum Sensitive Itemsets Conflict First (MSICF), to hide sensitive HUIs in PPUM [152]. The main task of PPUM is to hide the sensitive high-utility itemsets (SHUIs). Lin et al. first developed a genetic-algorithm-based method to hide the user-specified SHUIs by inserting the dummy transactions into the original databases [153]. Yun et al. then developed a tree-based algorithm called the Fast Perturbation algorithm Using a Tree structure and Tables (FPUTT) for hiding SHUIs [154]. Then, other faster and more efficient algorithms were developed for PPUM, such as [155], [156]. A recent overview of PPUM has been reported by Gan et al. [157].

5 Open-Source Software and Datasets

5.1 Open-Source Software

Although the problem of UPM has been studied for more than 15 years, and the advanced topic of utility pattern mining also has been extended to many research fields, few implementations or source code of these algorithms have been released. This raises some barriers to other researchers in that they need to re-implement algorithms to use them or compare their performance with that of novel proposed algorithms. To make matters worse, this may introduce unfairness in running experimental comparisons, since the performance of pattern mining algorithms may commonly depend on the compiler and machine architecture used. We now list some open-source software specialized for UPM.

• *UP-Miner*. Tseng et al. proposed a first-of-its-kind utility mining toolbox named Utility Pattern Miner (UP-Miner) [158]. UP-Miner provides various models for utility-oriented pattern mining. The main merits of UP-Miner have three aspects. First, to the best of our knowledge, it is the first-of-its-kind cross-platform utility mining system. Second, it provides complete Java implementations of 13 algorithms for discovering different types of utility-oriented patterns, such as high-utility itemset (HUI), high-utility

sequential rule (HUSR), high-utility sequential pattern (HUSP), and high-utility episode (HUE), as well as the concise representations of utility patterns. In addition, it offers four functionalities for processing utility-based databases. Third, the toolbox and relevant materials, including source codes, demo paper, benchmark datasets, and data generators, have been made public on Website[3] for the benefit of the research community.

• *SPMF*. As a well-known open-source data mining library, SPMF [159] offers implementations of many algorithms and has been cited in more than 700 research papers since 2010. SPMF is written in Java, and provides implementations of 170 data mining algorithms, specializing in pattern mining. SPMF has the largest collection of implementations of various algorithms for pattern mining algorithms (i.e., FPM, ARM, SPM, etc.) and provides a user-friendly graphical interface.[4] In particular, it also provides the relevant materials, including source codes, documentation, user instruction, benchmark datasets, data generators, and academic papers. SPMF offers up to 30 algorithms for utility-oriented pattern mining, such as Two-Phase, UP-Growth, UP-Growth+, HUI-Miner, EFIM, USpan, HUSP-ULL, and many other state-of-the-art algorithms. More specifically, SPMF is distributed under the GPL v3 license and is suitable for both academic and industrial purposes.

5.2 Datasets for UPM

Several datasets are commonly used in the studies of UPM. All of them have been released at websites, such as SPMF [159], UP-Miner [158].

Real Datasets. *foodmart*: it is provided by Microsoft containing 21,556 customer transactions and 1,559 distinct items from an anonymous chain store. It contains the quantity and a unit profit of each item. *yoochoose-buys* commercial dataset was constructed in the RecSys Challenge 2015.[5] It contains a collection of 1,150,753 sessions from a retailer, where each session is encapsulating the click events. The total number of item IDs and category IDs is 54,287 and 347 correspondingly, with an interval of 6 months. *UK-online*[6]: it contains 541,909 transactions, which occurs between 01/12/2010 and 09/12/2011 for a UK-based and registered non-store online retail. The original data contains the real timestamp and many noise values. It has the attributes as InvoiceNo, StockCode, Quantity, InvoiceDate, UnitPrice, CustomerID, etc.

Semi-Authentic Datasets. They are the real datasets[7] (e.g., *chess*, *retail*, *kosarak*, *mushroom*, *accidents*, *BMSPOS2*) with synthetic utility values. The internal utility values are generated using a uniform distribution in [1, 10]. The external utility values are generated using a Gaussian (normal) distribution. Detailed description and characteristics of these real datasets can be referred to SPMF [159], UP-Miner [158], or existing UPM literature.

Synthetic Datasets. There are some synthetic itemset-based or sequence-based datasets generated by IBM Quest Dataset Generator [160], which have been commonly used in UPM. For example, the itemset-based synthetic *T10I4D100K*, *T40I10D100K*; and the sequence synthetic *C8S6T4I3D|X|K* are described in [124].

6 Open Challenges and Opportunities

Here, we discuss important open problems that have the potential to become future research areas in utility-oriented pattern mining. Owing to the rapid growth of the volume of data stored in databases, we have entered the era of Big Data. While analyzing utility-oriented patterns, we have identified numerous technical challenges and opportunities for UPM. We next highlight some important research opportunities, which are common to many, and sometimes all, UPM algorithms.

• *Application-Driven Algorithms*. Up to now, most of the algorithms for UPM have been developed to improve the efficiency of mining process. The effectiveness of the algorithms for UPM is also very important, because it is related to the usefulness on various data, constraints, and applications. In general, as described in Section 2.3, the application-driven algorithms with many particular features of utility patterns reflect real-life problems of different applications in various fields. How to propose a specialized UPM model for different applications (e.g., business, web intelligent, risk perdition, smart city, financial analysis, Internet of Things, Biomedicine, smart transportation) and experimentally show its effectiveness is necessary and challenging. Moreover, the incorporation of domain knowledge [161] has a higher influence on performance for some data mining methods. Utility mining guided by domain knowledge thus provides many opportunities.

• *Developing More Efficient Algorithms*. Traditionally, most pattern mining algorithms, especially UPM algorithms, are computationally expensive in terms of execution time and memory cost. This may be a serious problem for dense databases or databases containing numerous items/sequences or long transactions, depending on the minimum utility threshold chosen by the user. Although current UPM algorithms (e.g., HUI-Miner [86], EFIM [117], and mHUIMiner [92]) are much efficient than previous Apriori-based and tree-based algorithms, there is still room for improvement. 1) It is important to reduce the search space, and this requires to design novel pruning strategies that rely on upper-bounds on the utility measure that are tighter than current measures. 2) Moreover, we can design novel data structures to more quickly calculate the utility and upper-bounds, and integrate constraints in the mining process to reduce the search space. 3) Fast approximate algorithms [130] that guarantee a maximum error can also be developed.

• *Unified Framework for UPM*. Many variations of utility mining have been proposed to deal with various types of data and to solve different problems. The current paradigm used to solve utility-oriented pattern mining problem is to first define the definition of utility-based patterns with interest and their properties, and then develop an algorithm that can exploit the properties of the utility (e.g., upper bound) to efficiently mine them. Hence, this laborious process can be avoided if the following problem is solved: "Is there a paradigm such that existing and new definitions of

3. http://bigdatalab.cs.nctu.edu.tw/software.php
4. http://www.philippe-fournier-viger.com/spmf/index.php
5. https://recsys.acm.org/recsys15/challenge/
6. http://archive.ics.uci.edu/ml/datasets/Online+Retail/
7. http://fimi.ua.ac.be/data/

utility-based pattern (HUI [77], HUSP [42], [123], HUE [43]) can be solved by a unifying algorithm?" Owing to these challenges, the utility-oriented pattern mining problem, in its most general form, is not easy to solve. In fact, most of the existing utility mining techniques (e.g., HUIM [77], HUSPM [42], [123], HUEM [43], etc.) solve a specific formulation of a specific problem. Therefore, how to formalize utility mining tasks in a generic framework is crucial and challenging. Focus on general principles and modeling of UPM rather than specific implementations is more important and challenging.

• *Deal with Complex Data*. The amount of complex data has been explored during the past two decades, while most of the data mining and analysis approaches are not utility oriented. Many current techniques of UPM are not suited to dealing with various types of complex data, such as "structured data"[8] (i.e., pattern mining), "unstructured data"[9] (including documents, health records, audio, video, images, etc.), and "semi-structured data"[10] (i.e., XML, JSON), and most of these are the heterogeneous data. More specifically, the dynamic data [47], the uncertain data [8], [73], the high-dimensional datasets of moderate size, or the very large datasets of moderate complexity in real-life applications are commonly seen in different domains and applications. Bridging this gap requires the solution of fundamentally new research problems, which can be grouped into the following challenges: 1) how to define the utility function integrating with various rich features on complex data; 2) how to achieve utility maximization for the goal and mining task; and 3) how to develop new frameworks and algorithms to deal with new types of data. A need therefore arises for a better framework that extends the existing data mining methodologies, techniques, and tools, guided by utility and knowledge.

• *Large-Scale Data*. Efficiently mining large-scale databases may result in a high computational cost and memory consumption. Under the batch model, traditional UPM algorithms must be repeatedly applied to obtain updated results when new data are inserted [47]. However, in the Big Data era, incrementally or dynamically processing data [47] and taking into account the results of prior analysis is crucial. There are some challenging research opportunities of UPM for handling large-scale data (as described in Section 4.7): how to design the parallelized UPM algorithms and how to develop the UPM algorithms based on the existing technologies of Big Data (i.e., MapReduce [132] and Spark [133]). Some other promising areas of research are the design of distributed, parallel, multi-core, or graphical-processing-unit (GPU)-based algorithms [45], [162] for UPM. There are some open challenges and opportunities to improve the scalability of utility mining tasks from resource-constraint devices to collaborative and hybrid execution models.

• *Scalable Real-Time Pattern Mining*. There exists many interactive approaches for interactive data mining, but few have been extended to address the challenge of utility. And it is not trivial to adapt them. One of the most important future challenges is to develop scalable high-utility pattern online mining approaches for streaming data from electronic commerce. Specifically, research should focus on algorithms that are sub-linear to the input or, at the very least, linear. Other computational challenges, such as the demands of the results being returned in real- or near-real-time, are the open issues in the data mining community. For example, real-time mining with optimization requires a new formalism and solving techniques. As mentioned before, increasing quantity and complexity of data demands scalable solutions. Using the existing computational infrastructures for real-time utility-oriented mining massive datasets may be a feasible way.

8. https://en.wikipedia.org/wiki/Structure_mining
9. https://en.wikipedia.org/wiki/Unstructured_data
10. https://en.wikipedia.org/wiki/Semi-structured_data

7 Conclusions

The term utility is commonly used to mean "the quality of being useful," and utilities are widely used in data-mining and decision-making processes to extract different useful kinds of knowledge. Utilities are subjective and can be acquired from domain experts/users. Utility mining in data is a vital task, with numerous high-impact applications, including cross-marketing, e-commerce, finance, medical, and biomedical applications. Up to now, many techniques and approaches have been extensively proposed for the task of UPM. In this survey, we have provided a comprehensive review of utility-oriented pattern mining, both in terms of current status and future directions. This survey describes various problems associated with mining utility-based patterns and methods for addressing these problems, including 1) *high-utility itemset mining* (HUIM), 2) *high-utility association rule mining* (HUARM), 3) *high-utility sequential pattern mining* (HUSPM), 4) *high-utility sequential rule mining* (HUSRM), and 5) *high-utility episode mining* (HUEM). Overall, we have not only reviewed the most common, as well as the state-of-the-art, approaches for UPM but have also provided a comprehensive review of advanced UPM topics. Finally, we have identified several important issues and research opportunities for UPM.

Acknowledgments

This research was partially supported by the Shenzhen Technical Project under No. KQJSCX 20170726103424709 and No. JCYJ 20170307151733005, and by the China Scholarship Council Program. The authors would like to thank the editors and anonymous reviewers for their detailed comments and constructive suggestions which have improved the quality of this paper.

References

[1] M. S. Chen, J. Han, and P. S. Yu, "Data mining: An overview from a database perspective," *IEEE Trans. Knowl. Data Eng.*, vol. 8, no. 6, pp. 866–883, Dec. 1996.
[2] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*. Amsterdam, The Netherlands: Elsevier, 2011.
[3] Y. S. Koh and S. D. Ravana, "Unsupervised rare pattern mining: A survey," *ACM Trans. Knowl. Discovery Data*, vol. 10, no. 4, 2016, Art. no. 45.
[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 Interdisciplinary Rev.: Data Mining Knowl. Discovery*, vol. 7, no. 4, 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. W. Tsai, C. F. Lai, M. C. Chiang, L. T. Yang, et al.,"Data mining for internet of things: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 77–97, Jan.–Mar. 2014.

[7] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach," *Data Mining Knowl. Discovery*, vol. 8, no. 1, pp. 53–87, 2004.

[8] C. C. Aggarwal, Y. Li, J. Wang, and J. Wang, "Frequent pattern mining with uncertain data," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 29–38.

[9] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *ACM SIGMOD Rec.*, vol. 22, no. 2, pp. 207–216, 1993.

[10] R. Agrawal, R. Srikant, et al., "Fast algorithms for mining association rules," in *Proc. 20th Int. Conf. Very Large Data Bases*, 1994, pp. 487–499.

[11] H. Mannila, H. Toivonen, and A. I. Verkamo, "Discovery of frequent episodes in event sequences," *Data Mining Knowl. Discovery*, vol. 1, no. 3, pp. 259–289, 1997.

[12] K. Y. Huang and C. H. Chang, "Efficient mining of frequent episodes from complex sequences," *Inf. Syst.*, vol. 33, no. 1, pp. 96–114, 2008.

[13] A. Achar, S. Laxman, and P. Sastry, "A unified view of the Apriori-based algorithms for frequent episode discovery," *Knowl. Inf. Syst.*, vol. 31, no. 2, pp. 223–250, 2012.

[14] A. Achar, A. Ibrahim, and P. Sastry, "Pattern-growth based frequent serial episode discovery," *Data Knowl. Eng.*, vol. 87, pp. 91–108, 2013.

[15] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proc. 11th Int. Conf. Data Eng.*, 1995, pp. 3–14.

[16] J. Han, J. Pei, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, "PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth," in *Proc. 17th Int. Conf. Data Eng.*, 2001, pp. 215–224.

[17] W. Gan, J. C. W. Lin, P. Fournier-Viger, H. C. Chao, and P. S. Yu, "A survey of parallel sequential pattern mining," *ACM Trans. Knowl. Discovery Data*, vol. 13, no. 3, 2019, Art. no. 25.

[18] L. Geng and H. J. Hamilton, "Interestingness measures for data mining: A survey," *ACM Comput. Surveys*, vol. 38, no. 3, 2006, Art. no. 9.

[19] J. Pei, J. Han, and L. V. Lakshmanan, "Mining frequent itemsets with convertible constraints," in *Proc. 17th Int. Conf. Data Eng.*, 2001, pp. 433–442.

[20] P. N. Tan, V. Kumar, and J. Srivastava, "Selecting the right objective measure for association analysis," *Inf. Syst.*, vol. 29, no. 4, pp. 293–313, 2004.

[21] K. McGarry, "A survey of interestingness measures for knowledge discovery," *Knowl. Eng. Rev.*, vol. 20, no. 1, pp. 39–61, 2005.

[22] Y. D. Shen, Z. Zhang, and Q. Yang, "Objective-oriented utility-based association mining," in *Proc. IEEE Int. Conf. Data Mining*, 2002, pp. 426–433.

[23] R. Chan, Q. Yang, and Y. D. Shen, "Mining high utility itemsets," in *Proc. 3rd IEEE Int. Conf. Data Mining*, 2003, pp. 19–26.

[24] H. Yao, H. J. Hamilton, and L. Geng, "A unified framework for utility-based measures for mining itemsets," in *Proc. ACM SIGKDD 2nd Workshop Utility-Based Data Mining*, 2006, pp. 28–37.

[25] A. Marshall, *Principles of Economics*, 8th ed. London, U.K.: Macmillan, 1926.

[26] R. J. Hilderman and H. J. Hamilton, "Measuring the interestingness of discovered knowledge: A principled approach," *Intell. Data Anal.*, vol. 7, no. 4, pp. 347–382, 2003.

[27] A. Silberschatz and A. Tuzhilin, "On subjective measures of interestingness in knowledge discovery," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 1995, pp. 275–281.

[28] T. De Bie, "Maximum entropy models and subjective interestingness: An application to tiles in binary databases," *Data Mining Knowl. Discovery*, vol. 23, no. 3, pp. 407–446, 2011.

[29] H. Yao and H. J. Hamilton, "Mining itemset utilities from transaction databases," *Data Knowl. Eng.*, vol. 59, no. 3, pp. 603–626, 2006.

[30] H. F. Li, H. Y. Huang, Y. C. Chen, Y. J. Liu, and S. Y. Lee, "Fast and memory efficient mining of high utility itemsets in data streams," in *Proc. 8th IEEE Int. Conf. Data Mining*, 2008, pp. 881–886.

[31] C. F. Ahmed, S. K. Tanbeer, B. S. Jeong, and Y. K. Lee, "Efficient tree structures for high utility pattern mining in incremental databases," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 12, pp. 1708–1721, Dec. 2009.

[32] B. E. Shie, V. S. Tseng, and P. S. Yu, "Online mining of temporal maximal utility itemsets from data streams," in *Proc. ACM Symp. Appl. Comput.*, 2010, pp. 1622–1626.

[33] A. Erwin, R. P. Gopalan, and N. Achuthan, "Efficient mining of high utility itemsets from large datasets," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2008, pp. 554–561.

[34] 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, 2008.

[35] B. E. Shie, H. F. Hsiao, V. S. Tseng, and P. S. Yu, "Mining high utility mobile sequential patterns in mobile commerce environments," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2011, pp. 224–238.

[36] B. E. Shie, H. F. Hsiao, and V. S. Tseng, "Efficient algorithms for discovering high utility user behavior patterns in mobile commerce environments," *Knowl. Inf. Syst.*, vol. 37, no. 2, pp. 363–387, 2013.

[37] M. Zihayat, H. Davoudi, and A. An, "Mining significant high utility gene regulation sequential patterns," *BMC Syst. Biol.*, vol. 11, no. 6, 2017, Art. no. 109.

[38] S. J. Yen and Y. S. Lee, "Mining high utility quantitative association rules," in *Proc. Int. Conf. Data Warehousing Knowl. Discovery*, 2007, pp. 283–292.

[39] D. Lee, S. H. Park, and S. Moon, "Utility-based association rule mining: A marketing solution for cross-selling," *Expert Syst. Appl.*, vol. 40, no. 7, pp. 2715–2725, 2013.

[40] J. Sahoo, A. K. Das, and A. Goswami, "An efficient approach for mining association rules from high utility itemsets," *Expert Syst. Appl.*, vol. 42, no. 13, pp. 5754–5778, 2015.

[41] C. F. Ahmed, S. K. Tanbeer, and B. S. Jeong, "A novel approach for mining high-utility sequential patterns in sequence databases," *ETRI J.*, vol. 32, no. 5, pp. 676–686, 2010.

[42] J. Yin, Z. Zheng, and L. Cao, "USpan: An efficient algorithm for mining high utility sequential patterns," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 660–668.

[43] C. W. Wu, Y. F. Lin, P. S. Yu, and V. S. Tseng, "Mining high utility episodes in complex event sequences," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 536–544.

[44] Y. F. Lin, C. W. Wu, C. F. Huang, and V. S. Tseng, "Discovering utility-based episode rules in complex event sequences," *Expert Syst. Appl.*, vol. 42, no. 12, pp. 5303–5314, 2015.

[45] W. Gan, J. C. W. Lin, H. C. Chao, and J. Zhan, "Data mining in distributed environment: A survey," *Wiley Interdisciplinary Rev.: Data Mining Knowl. Discovery*, vol. 7, no. 6, 2017, Art. no. e1216.

[46] B. Nath, D. Bhattacharyya, and A. Ghosh, "Incremental association rule mining: A survey," *Wiley Interdisciplinary Rev.: Data Mining Knowl. Discovery*, vol. 3, no. 3, pp. 157–169, 2013.

[47] W. Gan, J. C. W. Lin, P. Fournier-Viger, H. C. Chao, T. P. Hong, and H. Fujita, "A survey of incremental high-utility itemset mining," *Wiley Interdisciplinary Rev.: Data Mining Knowl. Discovery*, vol. 8, no. 2, 2018, Art. no. e1242.

[48] 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.

[49] S. Zida, P. Fournier-Viger, C. W. Wu, J. C. W. Lin, and V. S. Tseng, "Efficient mining of high-utility sequential rules," in *Proc. Int. Workshop Mach. Learn. Data Mining Pattern Recognit.*, 2015, pp. 157–171.

[50] P. Fournier-Viger, U. Faghihi, R. Nkambou, and E. M. Nguifo, "CMRules: Mining sequential rules common to several sequences," *Knowl.-Based Syst.*, vol. 25, no. 1, pp. 63–76, 2012.

[51] C. Jiang, F. Coenen, and M. Zito, "A survey of frequent subgraph mining algorithms," *Knowl. Eng. Rev.*, vol. 28, no. 1, pp. 75–105, 2013.

[52] H. Yao, H. J. Hamilton, and C. J. Butz, "A foundational approach to mining itemset utilities from databases," in *Proc. SIAM Int. Conf. Data Mining*, 2004, pp. 482–486.

[53] C. H. Cai, A. W. C. Fu, C. Cheng, and W. Kwong, "Mining association rules with weighted items," in *Proc. Int. Database Eng. Appl. Symp.*, 1998, pp. 68–77.

[54] W. Wang, J. Yang, and P. S. Yu, "Efficient mining of weighted association rules (WAR)," in *Proc. 6th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2000, pp. 270–274.

[55] F. Tao, F. Murtagh, and M. Farid, "Weighted association rule mining using weighted support and significance framework," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2003, pp. 661–666.

[56] K. Sun and F. Bai, "Mining weighted association rules without preassigned weights," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 4, pp. 489–495, Apr. 2008.

[57] J. C. W. Lin, W. Gan, P. Fournier-Viger, and T. P. Hong, "RWFIM: Recent weighted-frequent itemsets mining," *Eng. Appl. Artif. Intell.*, vol. 45, pp. 18–32, 2015.

[58] J. C. W. Lin, W. Gan, P. Fournier-Viger, T. P. Hong, and V. S. Tseng, "Weighted frequent itemset mining over uncertain databases," *Appl. Intell.*, vol. 44, no. 1, pp. 232–250, 2016.

[59] W. Gan, J. C. W. Lin, P. Fournier-Viger, H. C. Chao, J. M. T. Wu, and J. Zhan, "Extracting recent weighted-based patterns from uncertain temporal databases," *Eng. Appl. Artif. Intell.*, vol. 61, pp. 161–172, 2017.

[60] Y. C. Li, J. S. Yeh, and C. C. Chang, "Direct candidates generation: A novel algorithm for discovering complete share-frequent itemsets," in *Proc. Int. Conf. Fuzzy Syst. Knowl. Discovery*, 2005, pp. 551–560.

[61] 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. Inform.*, vol. 30, no. 2, pp. 109–126, 2016.

[62] 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, 2018.

[63] C. F. Ahmed, S. K. Tanbeer, and B. S. Jeong, "A framework for mining high utility web access sequences," *IETE Tech. Rev.*, vol. 28, no. 1, pp. 3–16, 2011.

[64] C. F. Ahmed, S. K. Tanbeer, B. S. Jeong, and Y. K. Lee, "Efficient mining of utility-based web path traversal patterns," in *Proc. 11th Int. Conf. Adv. Commun. Technol.*, 2009, pp. 2215–2218.

[65] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, 2010.

[66] L. Golab and M. T. Özsu, "Issues in data stream management," *ACM SIGMOD Rec.*, vol. 32, no. 2, pp. 5–14, 2003.

[67] Y. Chi, H. Wang, P. S. Yu, and R. R. Muntz, "Moment: Maintaining closed frequent itemsets over a stream sliding window," in *Proc. 4th IEEE Int. Conf. Data Mining*, 2004, pp. 59–66.

[68] M. Zihayat and A. An, "Mining top-*k* high utility patterns over data streams," *Inf. Sci.*, vol. 285, pp. 138–161, 2014.

[69] 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. 12 947–12 960, 2012.

[70] C. F. Ahmed, S. K. Tanbeer, B. S. Jeong, and H. J. Choi, "Interactive mining of high utility patterns over data streams," *Expert Syst. Appl.*, vol. 39, no. 15, pp. 11 979–11 991, 2012.

[71] Y. C. Liu, C. P. Cheng, and V. S. Tseng, "Mining differential top-*k* co-expression patterns from time course comparative gene expression datasets," *BMC Bioinf.*, vol. 14, no. 1, 2013, Art. no. 230.

[72] T. P. Hong, C. H. Lee, and S. L. Wang, "Effective utility mining with the measure of average utility," *Expert Syst. Appl.*, vol. 38, no. 7, pp. 8259–8265, 2011.

[73] J. C. W. Lin, W. Gan, P. Fournier-Viger, T. P. Hong, and V. S. Tseng, "Efficient algorithms for mining high-utility itemsets in uncertain databases," *Knowl.-Based Syst.*, vol. 96, pp. 171–187, 2016.

[74] C. K. Chui, B. Kao, and E. Hung, "Mining frequent itemsets from uncertain data," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2007, pp. 47–58.

[75] 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, 2017.

[76] W. Gan, J. C. W. Lin, P. Fournier-Viger, H. C. Chao, and P. S. Yu, "HUOPM: High-utility occupancy pattern mining," *IEEE Trans. Cybern.*, early access, Feb. 20, 2019, doi: 10.1109/TCYB.2019.2896267.

[77] 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*, 2005, pp. 689–695.

[78] J. Hu and A. Mojsilovic, "High-utility pattern mining: A method for discovery of high-utility item sets," *Pattern Recognit.*, vol. 40, no. 11, pp. 3317–3324, 2007.

[79] C. F. Ahmed, S. K. Tanbeer, B. S. Jeong, and Y. K. Lee, "An efficient candidate pruning technique for high utility pattern mining," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2009, pp. 749–756.

[80] V. S. Tseng, C. W. Wu, B. E. Shie, and P. S. Yu, "UP-Growth: An efficient algorithm for high utility itemset mining," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 253–262.

[81] W. Song, Y. Liu, and J. Li, "Mining high utility itemsets by dynamically pruning the tree structure," *Appl. Intell.*, vol. 40, no. 1, pp. 29–43, 2014.

[82] H. Ryang, U. Yun, and K. H. Ryu, "Fast algorithm for high utility pattern mining with the sum of item quantities," *Intell. Data Anal.*, vol. 20, no. 2, pp. 395–415, 2016.

[83] A. Erwin, R. P. Gopalan, and N. Achuthan, "CTU-Mine: An efficient high utility itemset mining algorithm using the pattern growth approach," in *Proc. 7th IEEE Int. Conf. Comput. Inf. Technol.*, 2007, pp. 71–76.

[84] C. W. Lin, T. P. Hong, and W. H. Lu, "An effective tree structure for mining high utility itemsets," *Expert Syst. Appl.*, vol. 38, no. 6, pp. 7419–7424, 2011.

[85] G. C. Lan, "A study on efficient algorithms for on-shelf utility mining," PhD Thesis, National Cheng Kung Univ., pp. 1–154, 2012.

[86] 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.

[87] P. Fournier-Viger, C. W. Wu, S. Zida, and V. S. Tseng, "FHM: Faster high-utility itemset mining using estimated utility cooccurrence pruning," in *Proc. Int. Symp. Methodologies Intell. Syst.*, 2014, pp. 83–92.

[88] J. Liu, K. Wang, and B. C. Fung, "Direct discovery of high utility itemsets without candidate generation," in *Proc. IEEE 12th Int. Conf. Data Mining*, 2012, pp. 984–989.

[89] S. Krishnamoorthy, "Pruning strategies for mining high utility itemsets," *Expert Syst. Appl.*, vol. 42, no. 5, pp. 2371–2381, 2015.

[90] 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. Mexican Int. Conf. Artif. Intell.*, 2015, pp. 530–546.

[91] H. Ryang and U. Yun, "Indexed list-based high utility pattern mining with utility upper-bound reduction and pattern combination techniques," *Knowl. Inf. Syst.*, vol. 51, no. 2, pp. 627–659, 2017.

[92] A. Y. Peng, Y. S. Koh, and P. Riddle, "mHUIMiner: A fast high utility itemset mining algorithm for sparse datasets," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2017, pp. 196–207.

[93] R. Rymon, "Search through systematic set enumeration," in *Proc. 3rd Int. Conf. Principles Knowl. Represenation Reasoning*, 1992, pp. 539–550.

[94] W. Song, Z. Zhang, and J. Li, "A high utility itemset mining algorithm based on subsume index," *Knowl. Inf. Syst.*, vol. 49, no. 1, pp. 315–340, 2016.

[95] W. Song, B. Yang, and Z. Xu, "Index-BitTableFI: An improved algorithm for mining frequent itemsets," *Knowl.-Based Syst.*, vol. 21, no. 6, pp. 507–513, 2008.

[96] G. C. Lan, T. P. Hong, and V. S. Tseng, "Efficiently mining high average-utility itemsets with an improved upper-bound strategy," *Int. J. Inf. Technol. Decision Making*, vol. 11, no. 05, pp. 1009–1030, 2012.

[97] J. C. W. Lin, T. Li, P. Fournier-Viger, T. P. Hong, J. Zhan, and M. Voznak, "An efficient algorithm to mine high average-utility itemsets," *Adv. Eng. Inform.*, vol. 30, no. 2, pp. 233–243, 2016.

[98] J. C. W. Lin, S. Ren, P. Fournier-Viger, and T. P. Hong, "EHAUPM: Efficient high average-utility pattern mining with tighter upper bounds," *IEEE Access*, vol. 5, pp. 12 927–12 940, 2017.

[99] U. Yun, D. Kim, E. Yoon, and H. Fujita, "Damped window based high average utility pattern mining over data streams," *Knowl.-Based Syst.*, vol. 44, pp. 188–205, 2017.

[100] D. W. Cheung, J. Han, V. T. Ng, and C. Wong, "Maintenance of discovered association rules in large databases: An incremental updating technique," in *Proc. 12th Int. Conf. Data Eng.*, 1996, pp. 106–114.

[101] T. P. Hong, C. Y. Wang, and Y. H. Tao, "A new incremental data mining algorithm using pre-large itemsets," *Intell. Data Anal.*, vol. 5, no. 2, pp. 111–129, 2001.

[102] C. W. Lin, G. C. Lan, and T. P. Hong, "An incremental mining algorithm for high utility itemsets," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 7173–7180, 2012.

[103] C. W. Lin, T. P. Hong, G. C. Lan, J. W. Wong, and W. Y. Lin, "Incrementally mining high utility patterns based on pre-large concept," *Appl. Intell.*, vol. 40, no. 2, pp. 343–357, 2014.

[104] J. C. W. Lin, W. Gan, T. P. Hong, and B. Zhang, "An incremental high-utility mining algorithm with transaction insertion," *Sci. World J.*, vol. 2015, 2015, Art. no. 161564.

[105] P. Fournier-Viger, J. C. W. Lin, T. Gueniche, and P. Barhate, "Efficient incremental high utility itemset mining," in *Proc. ASE BigData Social Inform.*, 2015, Art. no. 53.

[106] D. W. Cheung, S. D. Lee, and B. Kao, "A general incremental technique for maintaining discovered association rules," in *Proc. 5th Int. Conf. Database Syst. Adv. Appl.*, 1997, pp. 185–194.

[107] C. W. Lin, G. C. Lan, and T. P. Hong, "Mining high utility itemsets for transaction deletion in a dynamic database," *Intell. Data Anal.*, vol. 19, no. 1, pp. 43–55, 2015.

[108] C. W. Lin, T. P. Hong, G. C. Lan, J. W. Wong, and W. Y. Lin, "Efficient updating of discovered high-utility itemsets for transaction deletion in dynamic databases," *Adv. Eng. Inform.*, vol. 29, no. 1, pp. 16–27, 2015.

[109] J. C. W. Lin, W. Gan, and T. P. Hong, "A fast maintenance algorithm of the discovered high-utility itemsets with transaction deletion," *Intell. Data Anal.*, vol. 20, no. 4, pp. 891–913, 2016.

[110] C. W. Lin, B. Zhang, W. Gan, B. W. Chen, S. Rho, and T. P. Hong, "Updating high-utility pattern trees with transaction modification," *Multimedia Tools Appl.*, vol. 75, no. 9, pp. 4887–4912, 2016.

[111] J. C. W. Lin, W. Gan, and T. P. Hong, "A fast updated algorithm to maintain the discovered high-utility itemsets for transaction modification," *Adv. Eng. Inform.*, vol. 29, no. 3, pp. 562–574, 2015.

[112] J. F. Boulicaut, A. Bykowski, and C. Rigotti, "Free-sets: A condensed representation of boolean data for the approximation of frequency queries," *Data Mining Knowl. Discovery*, vol. 7, no. 1, pp. 5–22, 2003.

[113] T. Calders and B. Goethals, "Mining all non-derivable frequent itemsets," in *Proc. Eur. Conf. Principles Data Mining Knowl. Discovery*, 2002, pp. 74–86.

[114] K. Gouda and M. J. Zaki, "Efficiently mining maximal frequent itemsets," in *Proc. IEEE Int. Conf. Data Mining*, 2001, pp. 163–170.

[115] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, "Efficient mining of association rules using closed itemset lattices," *Inf. Syst.*, vol. 24, no. 1, pp. 25–46, 1999.

[116] V. S. Tseng, C. W. Wu, P. Fournier-Viger, and P. S. Yu, "Efficient algorithms for mining the concise and lossless representation of high utility itemsets," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 3, pp. 726–739, Mar. 2015.

[117] P. Fournier-Viger, S. Zida, J. C. W. Lin, C. W. Wu, and V. S. Tseng, "EFIM-Closed: Fast and memory efficient discovery of closed high-utility itemsets," in *Proc. Int. Conf. Mach. Learn. Data Mining Pattern Recognit.*, 2016, pp. 199–213.

[118] 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.*, 2015, pp. 187–194.

[119] J. Sahoo, A. K. Das, and A. Goswami, "An efficient fast algorithm for discovering closed+ high utility itemsets," *Appl. Intell.*, vol. 45, no. 1, pp. 44–74, 2016.

[120] C. H. Li, C. W. Wu, and V. S. Tseng, "Efficient vertical mining of high utility quantitative itemsets," in *Proc. IEEE Int. Conf. Granular Comput.*, 2014, pp. 155–160.

[121] G. C. Lan, T. P. Hong, V. S. Tseng, and S. L. Wang, "Applying the maximum utility measure in high utility sequential pattern mining," *Expert Syst. Appl.*, vol. 41, no. 11, pp. 5071–5081, 2014.

[122] O. K. Alkan and P. Karagoz, "CRoM and HuspExt: Improving efficiency of high utility sequential pattern extraction," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 10, pp. 2645–2657, Oct. 2015.

[123] W. Gan, J. C. W. Lin, J. Zhang, H. C. Chao, H. Fujita, and P. S. Yu, "ProUM: Projection-based utility mining on sequence data," *arXiv:1904.07764*, 2019.

[124] W. Gan, J. C. W. Lin, J. Zhang, P. Fournier-Viger, H. C. Chao, and P. S. Yu, "Fast utility mining on sequence data," *arXiv:1904.12248*, 2019.

[125] J. Z. Wang, J. L. Huang, and Y. C. Chen, "On efficiently mining high utility sequential patterns," *Knowl. Inf. Syst.*, vol. 49, no. 2, pp. 597–627, 2016.

[126] G. Guo, L. Zhang, Q. Liu, E. Chen, F. Zhu, and C. Guan, "High utility episode mining made practical and fast," in *Proc. Int. Conf. Adv. Data Mining Appl.*, 2014, pp. 71–84.

[127] Y. F. Lin, C. F. Huang, and V. S. Tseng, "A novel methodology for stock investment using high utility episode mining and genetic algorithm," *Appl. Soft Comput.*, vol. 59, pp. 303–315, 2017.

[128] S. Rathore, S. Dawar, V. Goyal, and D. Patel, "Top-*k* high utility episode mining from a complex event sequence," in *Proc. 21st Int. Conf. Manage. Data Comput. Soc. India*, 2016, pp. 56–63.

[129] 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*, 2015, pp. 649–661.

[130] Y. Chen and A. An, "Approximate parallel high utility itemset mining," *Big Data Res.*, vol. 6, pp. 26–42, 2016.

[131] M. Zihayat, Z. Z. Hut, A. An, and Y. Hut, "Distributed and parallel high utility sequential pattern mining," in *Proc. IEEE Int. Conf. Big Data*, 2016, pp. 853–862.

[132] J. Dean and S. Ghemawat, "MapReduce: A flexible data processing tool," *Commun. ACM*, vol. 53, no. 1, pp. 72–77, 2010.

[133] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in *Proc. 9th USENIX Conf. Netw. Syst. Des. Implementation*, 2012, pp. 2–2.

[134] M. Zihayat, Y. Chen, and A. An, "Memory-adaptive high utility sequential pattern mining over data streams," *Mach. Learn.*, vol. 106, no. 6, pp. 799–836, 2017.

[135] G. S. Manku and R. Motwani, "Approximate frequency counts over data streams," in *Proc. 28th Int. Conf. Very Large Databases*, 2002, pp. 346–357.

[136] H. F. Li, M. K. Shan, and S. Y. Lee, "DSM-FI: An efficient algorithm for mining frequent itemsets in data streams," *Knowl. Inf. Syst.*, vol. 17, no. 1, pp. 79–97, 2008.

[137] C. J. Chu, V. S. Tseng, and T. Liang, "An efficient algorithm for mining temporal high utility itemsets from data streams," *J. Syst. Softw.*, vol. 81, no. 7, pp. 1105–1117, 2008.

[138] 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, 2011.

[139] J. C. W. Lin, W. Gan, T. P. Hong, and V. S. Tseng, "Efficient algorithms for mining up-to-date high-utility patterns," *Adv. Eng. Inform.*, vol. 29, no. 3, pp. 648–661, 2015.

[140] W. Gan, J. C. W. Lin, P. Fournier-Viger, and H. C. Chao, "Mining recent high-utility patterns from temporal databases with time-sensitive constraint," in *Proc. Int. Conf. Big Data Analytics Knowl. Discovery*, 2016, pp. 3–18.

[141] 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, 2011.

[142] C. W. Wu, B. E. Shie, V. S. Tseng, and P. S. Yu, "Mining top-*k* high utility itemsets," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 78–86.

[143] J. C. W. Lin, W. Gan, P. Fournier-Viger, T. P. Hong, and J. Zhan, "Efficient mining of high-utility itemsets using multiple minimum utility thresholds," *Knowl.-Based Syst.*, vol. 113, pp. 100–115, 2016.

[144] J. C. W. Lin, P. Fournier-Viger, and W. Gan, "FHN: An efficient algorithm for mining high-utility itemsets with negative unit profits," *Knowl.-Based Syst.*, vol. 111, pp. 283–298, 2016.

[145] W. Gan, J. C. W. Lin, P. Fournier-Viger, H. C. Chao, and V. S. Tseng, "Mining high-utility itemsets with both positive and negative unit profits from uncertain databases," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2017, pp. 434–446.

[146] J. C. W. Lin, W. Gan, P. Fournier-Viger, T. P. Hong, and V. S. Tseng, "Efficiently mining uncertain high-utility itemsets," *Soft Comput.*, vol. 21, no. 11, pp. 2801–2820, 2017.

[147] W. Gan, J. C. W. Lin, H. C. Chao, T. P. Hong, and S. Y. Philip, "CoUPM: Correlated utility-based pattern mining," in *Proc. IEEE Int. Conf. Big Data*, 2018, pp. 2607–2616.

[148] R. Agrawal and R. Srikant, "Privacy-preserving data mining," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 439–450, 2000.

[149] Y. Lindell and B. Pinkas, "Privacy preserving data mining," in *Proc. Annu. Int. Cryptology Conf.*, 2000, pp. 36–54.

[150] C. C. Aggarwal and P. S. Yu, "A general survey of privacy-preserving data mining models and algorithms," in *Privacy-Preserving Data Mining*. Berlin, Germany: Springer, 2008, pp. 11–52.

[151] T. Zhu, G. Li, W. Zhou, and P. S. Yu, "Differentially private data publishing and analysis: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 8, pp. 1619–1638, Aug. 2017.

[152] J. S. Yeh and P. C. Hsu, "HHUIF and MSICF: Novel algorithms for privacy preserving utility mining," *Expert Syst. Appl.*, vol. 37, no. 7, pp. 4779–4786, 2010.

[153] C. W. Lin, T. P. Hong, J. W. Wong, G. C. Lan, and W. Y. Lin, "A GA-based approach to hide sensitive high utility itemsets," *Sci. World J.*, vol. 2014, 2014, Art. no. 804629.

[154] U. Yun and J. Kim, "A fast perturbation algorithm using tree structure for privacy preserving utility mining," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1149–1165, 2015.

[155] J. C. W. Lin, T. Y. Wu, P. Fournier-Viger, G. Lin, J. Zhan, and M. Voznak, "Fast algorithms for hiding sensitive high-utility itemsets in privacy-preserving utility mining," *Eng. Appl. Artif. Intell.*, vol. 55, pp. 269–284, 2016.

[156] J. C. W. Lin, T. P. Hong, P. Fournier-Viger, Q. Liu, J. W. Wong, and J. Zhan, "Efficient hiding of confidential high-utility itemsets with minimal side effects," *J. Exp. Theoretical Artif. Intell.*, vol. 29, no. 6, pp. 1225–1245, 2017.

[157] W. Gan, J. C. W. Lin, H. C. Chao, S. L. Wang, and P. S. Yu, "Privacy preserving utility mining: A survey," in *Proc. IEEE Int. Conf. Big Data*, 2018, pp. 2617–2626.
[158] V. S. Tseng, C. W. Wu, J. H. Lin, and P. Fournier-Viger, "UP-Miner: A utility pattern mining toolbox," in *Proc. IEEE Int. Conf. Data Mining Workshop*, 2015, pp. 1656–1659.
[159] P. Fournier-Viger, J. C. W. Lin, A. Gomariz, T. Gueniche, A. Soltani, Z. Deng, and H. T. Lam, "The SPMF open-source data mining library version 2," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2016, pp. 36–40.
[160] R. Agrawal and R. Srikant, "Quest synthetic data generator," 1994. [Online]. Available: http://www.Almaden.ibm.com/cs/quest/syndata.html
[161] L. Cao, "Domain-driven data mining: Challenges and prospects," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 6, pp. 755–769, Jun. 2010.
[162] S. Hong, T. Oguntebi, and K. Olukotun, "Efficient parallel graph exploration on multi-core CPU and GPU," in *Proc. Int. Conf. Parallel Archit. Compilation Techn.*, 2011, pp. 78–88.

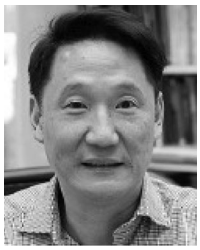
Wensheng Gan received the BS degree in computer science from South China Normal University, Guangdong, China, in 2013. He is working toward the PhD degree in computer science and technology at the Harbin Institute of Technology (Shenzhen), Guangdong, China. He was a joint PhD student with the University of Illinois at Chicago (UIC), from 2017 to 2019. His research interests include data mining, utility computing, and big data analytics. He has published more than 50 research papers in peer-reviewed journals (i.e., the *IEEE Transactions on Knowledge and Data Engineerning*, the *ACM Transactions on Knowledge Discovery from Data*, the *IEEE Transactions on Cybernetics*, *ACM Transactions on Data Science*, *Knowledge-Based Systems*) and conferences, which have received more than 600 citations.

Jerry Chun-Wei Lin (SM'19) received the PhD degree in computer science and information engineering from the National Cheng Kung University, Tainan, Taiwan, in 2010. He is an associate professor with the Western Norway University of Applied Sciences, Bergen, Norway. His research interests include data mining, big data analytics, machine learning, soft computing, and privacy-preserving and security. He has published more than 300 research papers in peer-reviewed international conferences (i.e., IEEE ICDE, IEEE ICDM, PKDD, and PAKDD) and journals (i.e., the *IEEE Transactions on Knowledge and Data Engineering*, the *IEEE Transactions on Cybernetics*, the *ACM Transactions on Knowledge Discovery from Data*, and the *ACM Transactions on Data Science*). He is the co-leader of the popular SPMF open-source data mining library, the project leader of PPSF open-source privacy and security library, the editor-in-chief (EiC) of the *Data Science and Pattern Recognition* (*DSPR*) journal, and associate editor of the *Journal of Internet Technology* and *IEEE Access*. He is the senior member of the IEEE and ACM.

Philippe Fournier-Viger received the PhD degree in computer science from the University of Quebec, Montreal, in 2010. He is full professor and Youth 1,000 scholar with the Harbin Institute of Technology (Shenzhen), Shenzhen, China. His research interests include pattern mining, sequence analysis and prediction, and social network mining. He has published more than 250 research papers in refereed international conferences and journals. He is the founder of the popular SPMF open-source data mining library, which has been cited in more than 800 research papers. He is editor-in-chief (EiC) of the *Data Science and Pattern Recognition* (*DSPR*) journal.

Han-Chieh Chao (SM'04) received the MS and PhD degrees in electrical engineering from Purdue University, in 1989 and 1993, respectively. He has been the president of the National Dong Hwa University since February 2016. His research interests include high-speed networks, wireless networks, IPv6-based networks, and artificial intelligence. He has published nearly 500 peer-reviewed research papers. He is the editor-in-chief (EiC) of the *IET Networks* and the *Journal of Internet Technology*. He has served as a guest editor of the *ACM Mobile Networks and Applications*, the *IEEE Journal on Selected Areas in Communications*, the *IEEE Communications Magazine*, the *IEEE Systems Journal*, *Computer Communications*, the *IEEE Proceedings Communications*, *Wireless Personal Communications*, and *Wireless Communications & Mobile Computing*. He is a senior member of the IEEE and a fellow of the IET.

Vincent S. Tseng (SM'16) received the PhD degree in computer science from National Chiao Tung University, Taiwan, in 1997. He is currently a distinguished professor with the Department of Computer Science, National Chiao Tung University, Taiwan. His research interests covering data mining, big data, biomedical informatics, mobile, and Web technologies. He has published more than 400 research papers in peer-reviewed journals and conferences and holds 15 patents. He has been on the editorial board of a number of journals, including the *IEEE Transactions on Knowledge and Data Engineering*, the *ACM Transactions on Knowledge Discovery from Data*, and the *IEEE Journal of Biomedical and Health Informatics*. He is a senior member of the IEEE.



Philip S. Yu (F'93) received the BS degree in electrical engineering from National Taiwan University, the MS and PhD degrees in electrical engineering from Stanford University, and the MBA degree from New York University. He is a distinguished professor of computer science with the University of Illinois at Chicago (UIC) and holds the Wexler Chair in Information Technology, UIC. Before joining UIC, he was with IBM, where he was manager of the Software Tools and Techniques Department, Thomas J. Watson Research Center. His research interests include databases, data mining, artificial intelligence, and privacy. He has published more than 1,300 papers in peer-reviewed journals (i.e., the *IEEE Transactions on Knowledge and Data*, the *IEEE Transactions on Parallel and Distributed*, the *ACM Transactions on Knowledge Discovery from Data*, the *VLDB Journal*) and conferences (i.e., SIGMOD, KDD, ICDE, WWW, AAAI, SIGIR, ICML, etc). He holds or has applied for more than 300 U.S. patents. He was the editor-in-chief of the *ACM Transactions on Knowledge Discovery from Data*. He received the ACM SIGKDD 2016 Innovation Award, and the IEEE Computer Society 2013 Technical Achievement Award. He is a fellow of the ACM and IEEE.