Journal of Computer Science and Cybernetics, V.36, N.1 (2020), 1–15
DOI 10.15625/1813-9663/36/1/14398

HIGH UTILITY ITEM INTERVAL SEQUENTIAL PATTERN MINING ALGORITHM

TRAN HUY DUONG[1,*], NGUYEN TRUONG THANG[1], VU DUC THI[2], TRAN THE ANH[1]

[1]*Institute of Information Technology, Vietnam Academy of Science and Technology*
[2]*Information Technology Institute, Vietnam National University (VNU)*
[*]*HuyDuong@ioit.ac.vn*

Abstract. High utility sequential pattern mining is a popular topic in data mining with the main purpose is to extract sequential patterns with high utility in the sequence database. Many recent works have proposed methods to solve this problem. However, most of them does not consider item intervals of sequential patterns which can lead to the extraction of sequential patterns with too long item interval, thus making little sense. In this paper, we propose a High Utility Item Interval Sequential Pattern (HUISP) algorithm to solve this problem. Our algorithm uses pattern growth approach and some techniques to increase algorithm's performance.

Keywords. Sequential pattern; Item interval; High utility.

1. INTRODUCTION

In general sequential pattern mining [1-4], the frequency of items and the order between items in the pattern are considered. For example, a sequential pattern like ⟨Bread, butter⟩ means that most customers will buy "butter" after they buy "bread". In this example, items like "bread" or "butter" have the same significance. Extracted sequential patterns in sequential pattern mining do not reflect other factors such as cost, profit or item interval between items. However, each item has different utility.

To solve this problem, Ahmed et al. [5] proposed a new research issue, namely utility sequential pattern mining, which considers not only frequency and order occurrence of each item in a quantitative sequence database but also their utility. According to Admed et al.'s definition, there are two ways of calculating the utility of an α pattern in an input sequence S_i using the sum of the utility of all the distinct occurrences of α in S_i; and using the maximum utilization of all occurrences of α in S_i. To simplify the calculation of utility, most recent works used the second way of calculation. These works [6-9] tried to improve performance of high utility mining problem in term of runtime and memory usage. However, none of them considered item interval of sequence pattern.

In fact, item intervals between itemsets have an important role. For instance, suppose that we have 2 sequences. S_1: ⟨(Tivi)(1 month, Bluetooth speaker)⟩ and S_2: ⟨(Tivi)(6 month, Bluetooth speaker)⟩. If we do not consider item intervals, these sequences are the same. But we can say that the sequence S_1 is more important than the sequence S_2, since S_1 has smaller item interval than S_2. To solve this problem, some works on item interval were proposed [10-12]. However, these works did not consider the item's significance when mining item interval sequential patterns. We previously proposed WIPrefixSpan [13] for mining weighted frequent pattern with item interval. In that study, we considered not only item intervals between itemsets but also item's significance (weighted). But the study did not consider the utility value in the database.

In real life data, each item has different utility and item intervals between itemsets are different, too. To solve high utility pattern mining with item interval, we have proposed UIPrefixSpan [14] algorithm. The algorithm considered not only item intervals of patterns but also their utilities. UIPrefixSpan, like the algorithm of Ahmed, is a two-phase algorithm which uses pattern growth approach of PrefixSpan [2] algorithm. In the first phase, UIPrefixSpan generates all candidate patterns. Then, in the second phase, the database is scanned again to calculate the real utility of all candidate patterns. After that, high utility sequential patterns with item interval which satisfy the minimum threshold are found. However, by generating candidate patterns in the first phase and checking again in the second phase low down the algorithm's performance.

In this paper, we develop a new algorithm called HUISP which uses some efficient techniques to improve algorithm's performance. HUISP requires one phase instead of two phases as in UIPrefixSpan.

The remainder of this paper is organized as follows: Section 2 provides a study of related works. Section 3 describes the problems and proposes the mining method for high utility sequential pattern with item interval. Section 4 presents experimental results. Conclusion and comments are presented in the last section.

2. RELATED WORKS

2.1. Item interval sequential pattern mining

Unlike traditional sequential pattern mining, item interval sequential pattern mining takes into account the item interval between items. In 2003, Chen et al. [10] proposed two algorithms, I-Apriori and I-PrefixSpan, for the time interval mining problem based on Apriori [1] and PrefixSpan [2], respectively. In 2005, Chen et al. [11] extended a previous work [10] by applying fuzzy theory to partition the time intervals using FTI-Apriori, an Apriori based algorithm that employs a distinct fuzzy membership function. Both of their works used extended sequence approach to represent time interval.

In 2006, Yu et al. [12] proposed a framework to generalize sequential pattern mining with item intervals. This work used four item interval constraints and the extended sequence approach to handle with item interval. This framework can handle too kinds of item interval measurement including item gap and time interval.

In 2017, A. SiriSha et al. [15] presented a new approach for mining time-interval based weighted sequential patterns. They used time intervals to obtain the weight of sequences, then sequential patterns are mined by considering the time interval weights.

In 2018, Phuong et al. [16] introduced fuzzy sequential patterns with fuzzy time-intervals in the quantitative sequence database problem. An algorithm named FSPFTIM which based on the Apriori algorithm was also proposed. In their work, both quantitative attributes and time intervals are represented by linguistic terms.

2.2. High utility sequential pattern mining

In many real-world datasets, not only occurrence frequency of patterns, but also their quantity and significance (like profit or price) have important roles. For example, the pattern ⟨iPhone X, MacBook Air ⟩may not be a high frequency pattern but it may contribute high profit to the shop due to its high profit. Thus, low frequency patterns may bring high profit but they may not be found in the sequence database by using traditional sequence pattern mining approach. To solve this problem, the high utility sequential pattern was proposed in 2010 with Admed [5] works. Admed proposed a new framework called high utility sequential pattern with two types of item utility: internal utility (representing item's quantity) and external utility (representing items importance like profit). Moreover, two algorithms were introduced using level-wise technique (the UL algorithm) and pattern-growth technique (the US algorithm). In Admed work, the utility of a pattern is calculated as the summation of utilities of all distinct occurrences in a sequence. This technique may find some personal repeatedly buying behaviors rather than common behaviors. To avoid such case and simplify the utility calculation, later works on HUSP mining used maximum utility measure.

In 2012, Yin et al. [6] proposed a general framework for mining HUSP and represents USpan algorithm which uses sequence weight utility (SWU) for pruning candidates and two data construction: LQS-tree and Utility Matrix for data representation. In 2014, Lan et al. [7] proposed PHUS, an algorithm based on a projection approach of PrefixSpan [2]. Their work used SWU as an upper bound for pruning candidates and a temporal sequence table for data representation. Alkan et al. [8] proposed a new upper bound called CRoM (Cummulated Rest of Match) which is used for pruning candidates before generation. They also represented the HuspExt algorithm with a Prefix tree structure for data representation. In 2019, Truong and Fournier ([17]) published a survey of high utility sequential pattern mining. This survey provided a concise overview of recent works in the HUSP mining field, presenting related problems and research opportunities. They also provided a formal theoretical framework for comparing upper-bounds used by HUSP mining algorithms.

3. PROBLEM STATEMENT AND DEFINITIONS

A quantitative sequence database with item interval $QiSDB$ is shown in Table 1. Each item in $QiSDB$ is assigned with a profit value as shown in Table 2.

Table 1. A QiSDB

$iSID$	Sequence
iS_1	$\langle 0, a[3]\rangle\ \langle 1, a[2]b[4]d[2]\rangle\ \langle 2, f[1]\rangle\ \langle 3, a[4]\rangle\ \langle 4, d[1]\rangle$
iS_2	$\langle 0, e[3]\rangle\ \langle 1, a[2]b[6]\rangle\ \langle 2, d[1]\rangle\ \langle 3, c[2]\rangle$
iS_3	$\langle 0, c[1]f[3]\rangle\ \langle 1, b[3]\rangle\ \langle 2, d[1]e[3]\rangle$
iS_4	$\langle 0, a[2]\rangle\ \langle 1, b[6]d[4]\rangle\ \langle 2, a[5]b[4]\rangle\ \langle 3, e[5]\rangle$
iS_5	$\langle 0, d[1]f[5]\rangle\ \langle 1, c[1]\rangle\ \langle 2, g[4]\rangle$
iS_6	$\langle 0, d[2]\rangle\ \langle 1, e[3]\rangle\ \langle 2, a[5]b[7]\rangle\ \langle 3, d[4]\rangle\ \langle 4, b[2]\rangle\ \langle 5, e[4]\rangle$
iS_7	$\langle 0, a[3]b[2]\rangle\ \langle 1, c[2]\rangle\ \langle 2, e[2]\rangle\ \langle 3, f[3]\rangle$
iS_8	$\langle 0, a[3]\rangle\ \langle 2, d[1]f[1]\rangle$
iS_9	$\langle 0, a[2]c[4]\rangle\ \langle 2, e[2]\rangle$

Table 2. Profit table

Items	Profit
a	3
b	2
c	1
d	6
e	5
f	2
g	8

3.1. Terms and definitions

Definition 1. An itemset $X \subseteq I$ is a set of items in the lexicographic order. If $|X| = r$ then itemset X is called r-itemset. $I = \{i_1, i_2, , i_n\}$ is a set of all items occurred in $QiSDB$.

Definition 2. Interval extended sequence.

$iS = \langle(t_{1,1}, X_1), (t_{1,2}, X_2), ..., (t_{1,m}, X_m)\rangle$ is a list of the itemsets ordered by their occurrence time. Here X_i $(1 \le i \le m)$ is an itemset and $t_{\alpha,\beta}$ is the item interval between itemsets X_α and X_β. If the datasets have occurrence time, $t_{\alpha,\beta}$ becomes the time interval and is defined by $t_{\alpha,\beta} = X_\beta.time - X_\alpha.time$.

Definition 3. Internal utility and external utility.

Internal utility of an item $i_j \in I$ in a sequence iS_a, denoted as $\text{iu}(i_j, iS_a)$, is quantity of item i_j in iS_a. External utility of item i_j is its significant value and denoted as $\text{eu}(i_j)$.

Table 1 is a $QiSDB$ with internal utility values and Table 2 is an external utility values table. The internal utility value represents items' quantities and external utility value represents profit per unit of that item. For example, for item a in iS_a, we have $\text{iu}(a, iS_9)$=2, and its external utility $\text{eu}(a)$=3. An item in a sequence may appear multiple times, in that case $\text{iu}(i_j, iS_k)$ is the maximum value among all the quantities of i_j in sequence iS_k. For example, $\text{iu}(a, iS_1)$= 4.

Definition 4. The utility of an item i_j in a sequence iS_a denoted as $\text{su}(i_j, iS_a)$ is defined by

$$\text{su}(i_j, iS_a) = \text{iu}(i_j, iS_a) \times \text{eu}(i_j).$$

For example, $\text{su}(a, iS_1)$= $\text{iu}(a, iS_1) \times \text{eu}(a) = 4 \times 3 = 12$.

Definition 5. The utility of a pattern $\alpha = \langle(t_{1,1}, X_1),\ (t_{1,2}, X_2), ...,\ (t_{1,n}, X_n)\rangle$ is a pattern with length n and $\alpha \subseteq iS_a$, sequence utility of the pattern α in iS_a denoted as $\text{su}(\alpha, iS_a)$ is defined by

$$\text{su}(\alpha, iS_a) = \max\{\sum_{i_j \in \alpha} su(i_j, iS_a), \forall \alpha \in iS_a\}$$

.

Definition 6. The sequence utility of an input sequence iS_a is the sum of utilities of all items in iS_a, which means

$$\text{su}(iS_a) = \sum_{i_j \in iS_a} \text{su}(i_j, iS_a).$$

Definition 7. The utility of a pattern α in a $QiSDB$ denoted as $\mathrm{su}(\alpha, QiSDB)$, is defined by

$$\mathrm{su}(\alpha, QiSDB) = \sum_{iS_a \in QiSDB} su(\alpha, iS_a).$$

Definition 8. The utility of a $QiSDB$ is defined by

$$\mathrm{su}(QiSDB) = \sum_{iS_a \in QiSDB} \mathrm{su}(iS_a).$$

Definition 9. Item interval constraints [12]

Given an interval extended sequence $\alpha = \langle (t_{1,1}, X_1),\ (t_{1,2}, X_2),\ (t_{1,3}, X_3), ...,\ (t_{1,n}, X_n)\rangle$, the item interval constraints are given as follows:

- C_1= $min\_item\_interval$ is a minimum item interval between any two adjacent itemsets, which mean $t_{i,i+1} \geq min\_item\_interval$ for all $\{i|1 \leq i \leq n-1\}$.
- C_2= $max\_item\_interval$ is a maximal item interval between any two adjacent itemsets, which mean $t_{i,i+1} \leq max\_item\_interval$ for all $\{i|1 \leq i \leq n-1\}$.
- C_3= $min\_whole\_interval$ is a minimum item interval between the first and the last itemset of the sequence, which mean $t_{1,n} \geq min\_whole\_interval$.
- C_4= $max\_whole\_interval$ is the maximal item interval between the first and the last itemset of the sequence, which mean $t_{1,n} \leq max\_whole\_interval$.

Definition 10. The high utility sequential pattern with item interval: Given a quantitative sequence database with item interval $QiSDB$, each item $i_j \in I$ in the input sequences iS_a is assigned with an internal utility $\mathrm{iu}(i_j, iS_a)$ and an external utility $\mathrm{eu}(i_j)$. Given a minimum utility threshold $minSeqUtil$ and four item interval constraints C_1, C_2, C_3, C_4, a sequential pattern $\alpha = \langle (t_{1,1}, X_1),\ (t_{1,2}, X_2),\ (t_{1,3}, X_3), ...,\ (t_{1,n}, X_n)\rangle$ is a high utility sequential pattern with item interval if it satisfies

$$\mathrm{su}(\alpha, QiSDB) \geq minSeqUtil$$

and $t_{\alpha,\beta}$ satisfies item interval constraints C_1, C_2, C_3, C_4.

Then the problem of mining high utility sequential pattern with item interval is defined as follows:

- Given a quantitative sequence database with item interval $QiSDB$, each item $i_j \in I$ in the input sequences iS_a is assigned with an internal utility $\mathrm{iu}(i_j, iS_a)$ and an external utility $\mathrm{eu}(i_j)$. Given a minimum utility threshold $minSeqUtil$ and four item interval constraints C_1, C_2, C_3, C_4, finding all high utility sequential patterns with item interval in $QiSDB$ which means finding the set L as

  $$L = \{\alpha \subseteq QiSDB | \mathrm{su}(\alpha, QiSDB) \geq minSeqUtil\}$$

  and $t_{\alpha,\beta}$ satisfies item interval constraints C_1, C_2, C_3, C_4.
- The high utility sequential pattern with item interval does not satisfy the downward closure property, which means a subsequence of a high utility sequential pattern with item interval may not be a high utility sequential pattern with item interval.

3.2. Maintaining downward closure property

In utility base framework, the downward closure property (DCP) of the sequence utility does not maintain. That means a subset of a high utility sequence does not necessarily be a high utility sequence. Thus, we can not use sequence utility but another value which ensures DCP for pruning the search space. The following definition of sequence weight utility is based on Ahmed [5] work.

Definition 11. Utility upper bound of sequence α. Given a sequence α, the utility upper bound of sequence α is denoted and defined as follows

$$\mathrm{ub}(\alpha) = \sum_{\alpha \subseteq iS_a \wedge iS_a \in QiSDB} \mathrm{su}(iS_a).$$

Definition 12. High utility upper bound sequential pattern. Given a minimum threshold $minSeqUtil$, a sequential pattern α is called a high utility upper bound sequential pattern if it satisfies

$$\mathrm{ub}(\alpha) \geq minSeqUtil$$

and α satisfies item interval constraints C_1, C_2, C_3, C_4.

High utility upper bound sequential patterns are used for pruning search space while maintaining downward closure property in mining high utility sequential patterns with item interval.

Lemma 1. *The utility upper bound maintains the downward closure property (DCP).*

Proof. Let α be a candidate pattern and d_α be a set of input sequences that contain α in $QiSDB$. Let β be a super-sequence of α then β cannot be presented in any sequence where α is absent. Therefore, the maximum utility upper bound of β is $\mathrm{ub}(\alpha)$. Then, if $\mathrm{ub}(\alpha)$ is less than minimum utility threshold $minSeqUtil$ then β is not a candidate pattern. ■

Lemma 2. *Given a QiSDB and a minimum utility threshold minSeqUtil, the high utility sequential patterns with item interval is a subset of high utility upper bound sequential patterns.*

Proof. Let α be a high utility sequential pattern with item interval. According to the Definition 5 and Definition 11, $\mathrm{su}(\alpha, QiSDB)$ must be less than or equal to $\mathrm{ub}(\alpha)$. So, if α is a high utility sequential pattern, it must be a high utility upper bound sequential pattern. ■

Lemma 3. *The downward closure property of the utility upper bound of sequential patterns still be kept while removing unpromising items.*

Proof. Given 2 items a and b that are 2 high utility upper bound sequences and item c is a low utility upper bound sequence. According to Lemma 2, because c is lower utility upper bound, then all patterns containing c cannot be high utility upper bound patterns. Assume that we have a pattern (a, b, c), item c can be eliminated from this pattern, then the new pattern will be (a, b). The utility of the new pattern (a, b) after removing item c can still be used as upper bound values of any subsequences in the new pattern like (a, b). So, downward closure property still be kept while removing unpromising items. ■

4. HIGH UTILITY SEQUENTIAL PATTERN MINING WITH ITEM INTERVAL (HUISP) ALGORITHM

In this section, we propose high utility sequential pattern mining with item interval (HUISP) algorithm. We use some techniques to improve algorithm's performance.

4.1. Utility table

Utility table is used to save utility and upper bound value of patterns in the mining process. Each row in the table includes three fields sequential pattern, upper bound and utility of that pattern. With a utility table, HUISP algorithm needs only one phase instead of two phases like UIPrefixSpan [14] and execution time of HUISP is also less than that of UIPrefixSpan.

We take item a in Table 1 as an example. Item a appears in these input sequences iS_1, iS_2, iS_4, iS_6, iS_7, iS_8, iS_9 and utilities of them are 55, 41, 90, 104, 31, 17, 20. Therefore $\text{ub}(a) = 358$. Item a appears three times in input sequence iS_1, according to Definitions 4, the utility of item a in iS_1 is 12. Similarly, the utilities of item a in input sequences iS_2, iS_4, iS_6, iS_7, iS_8, iS_9 are 6, 15, 15, 9, 9, 6, respectively. Due to Definition 7, utility of pattern $\langle a\rangle$ in QiSDB is 72. We do the same process to the rest of items, then we have the utility table as shown in Table 3.

Table 3. Utility table

Sequential pattern	ub	su
$\langle a\rangle$	358	72
$\langle b\rangle$	355	56
$\langle c\rangle$	390	10
$\langle d\rangle$	186	84
$\langle e\rangle$	320	95
$\langle f\rangle$	175	26
$\langle g\rangle$	49	32

4.2. Index table

We design an indexing structure to improve algorithm's performance. This structure is an index table with two fields candidate pattern and its index in the input sequence. The index of a pattern has two values the identifier of the input sequence which contains that pattern and the time value of the first appearance of the candidate pattern in the input sequence.

Table 4 is an index table of length-1 sequences. For example, sequence $\langle c\rangle$ has an index (2,3), (3,0), (5,1), (7,1), (9,0). Tuple (2,3) means item c happens in the second input sequence, the first appearance of item c in the second input sequence is at the position which has time value 3. Others tuple can be illustrated in the same way. The index table is useful when building a project database of length-1 pattern. Without index table, each time we build a project database of a length-1 pattern, we need a database scan (like in the PrefixSpan algorithm). With the index table, we can build all project databases of length-1 patterns without having to scan database again.

Table 4. Index table

Sequential pattern	Index
$\langle a\rangle$	(1,0), (2,1), (4,0), (6,2), (7,0), (8,0), (9,0)
$\langle b\rangle$	(1,1), (2,1), (3,1), (4,1), (6,1), (7,0)
$\langle c\rangle$	(2,3), (3,0), (5,1), (7,1), (9,0)
$\langle d\rangle$	(1,1), (2,2), (3,2), (4,1), (5,0), (6,0), (8,2)
$\langle e\rangle$	(2,0), (3,2), (4,3), (6,1), (7,2), (9,2)
$\langle f\rangle$	(1,2), (3,0), (5,0), (7,3), (8,2)
$\langle g\rangle$	(5,2)

4.3. The proposed algorithm

HUISP algorithm for mining high utility sequential patterns with item interval has some differences with UIPrefixSpan [14]. UIPrefixSpan algorithm has two phases: at the first phase, the algorithm finds all high utility upper bound sequential patterns with item interval; then in the second phase, real utilities of patterns are calculated to find all high utility sequential patterns with item interval. In HUISP algorithm, we use a utility table to calculate real utilities of patterns during the mining process. Moreover, we use a strategy to lower upper-bound of the patterns during the project database building process, that help to reduce un-potential patterns significantly. Beside that, by using index table, the time of finding subsequences is also reduced.

Below are the details of proposed algorithm HUISP for mining high utility sequential patterns with item interval.

Procedure HUISP($QiSDB, minSeqUtil, C_1, C_2, C_3, C_4$)

```
Input : – Item interval extended quantitative sequence database $QiSDB$
        – Minimum threshold: $minSeqUtil$
        – Item interval constraint $C_1, C_2, C_3, C_4$

Output : The set of high utility sequential patterns with item interval
1: Start
2:     $\alpha = \varnothing$;
3:     $R = \varnothing$;L= $\varnothing$; //$R$ is candidate set, L is high utility set
4:     Scan $QiSDB$, with each input sequence $iS_a$:
         – Calculate the utilities of all items in each input sequence su($i, iS_a$)
         – Calculate the utilities of each input sequence su($iS_a$)
5:     Build utility table for all item $i$ in $QiSDB$.
```

```
6:      Scan utility table, with each item $i$ in the table:
          – Let $\alpha = \langle(0, i)\rangle$
          – If $\text{ub}(\alpha) \geq minSeqUtil$ then $R = \{R, \alpha\}$
          – Eliminate all item which not belong to $R$ from $QiSDB$
          – If $\text{su}(\alpha, QiSDB) \geq minSeqUtil$ then $L = \{L, \alpha\}$
7:      Build the index table for each item in candidate set $R$
8:      With each sequential pattern $\alpha$ in $R$, build $\alpha$-project database $QiSDB|_\alpha$ base on
    index table. $QiSDB|_\alpha$ include all input sequence $iS_a$ of $QiSDB$ which contains $\alpha$.
          – Recalculate the utilities of each input sequence in $QiSDB|_\alpha$
          – $R = \text{subHUISP}(\alpha, QiSDB|_\alpha, R, minSeqUtil, C_1, C_2, C_3, C_4)$
9:      Output $L$.
10: End
```

Procedure subHUISP finds all high utility sequential patterns with item interval in project database $QiSDB|_\alpha$ with prefix α. This procedure is as follows.

Procedure subHUISP($\alpha, QiSDB|_\alpha, R, minSeqUtil, C_1, C_2, C_3, C_4$**)**

```
    Input : – $QiSDB|_\alpha$ - Project database with prefix $\alpha$
            – Minimum threshold: $minSeqUtil$
            – Item interval constraint $C_1, C_2, C_3, C_4$
    Output : The set of high utility sequential patterns with item interval with prefix $\alpha$.
1: Start
2:      Scan $QiSDB|_\alpha$, calculate $\text{ub}(i)$ of each item and find all pairs of item $(\triangle\ t; i)$ that
    satisfy $\text{ub}(i) \geq minSeqUtil$, $C_1$ and $C_2$, with $i$ is an item data and $\triangle\ t$ is item interval
    between $\alpha$ and $i$
3:      Eliminate from $QiSDB|_\alpha$ all item $i$ that do not satisfy the condition $\text{ub}(i) \geq minSeqUtil$.
    Recalculate utilities of input sequences $\text{su}(iS_a)$ of $QiSDB|_\alpha$
4:      Let $\alpha = \langle\alpha; (\triangle\ t; i)\rangle$
5:      Check if $\alpha$ satisfies the $C_4$ condition
6:      Only if it satisfies $C_4$:
          – $R = \text{subHUISP}(\alpha, QiSDB|_\alpha, R, minSeqUtil, C_1, C_2, C_3, C_4)$
          – If $\alpha$ satisfies $C_3$ then $R = \{R, \alpha\}$
          – If $\text{su}(\alpha, QiSDB) \geq minSeqUtil$ then $L = L, \alpha$
7:      Output $L$
8: End
```

5. EXPERIMENTAL RESULTS AND EVALUATION

In this section, we report our experimental results on the performance of HUISP in comparison with UIPrefixSpan.

In the general case, the complexity of the algorithm HUISP is exponential O(n^L), where n is the number of items in the dataset and L is the maximum length of the sequence in the whole database.

Experiments are performed on a computer with a 7th generation Core $i7$ processor running Windows 10 and 8 GB RAM. Two algorithms are implemented in Java. All memory measurements are done by using the Java API.

5.1. Experimental datasets

We use synthetic datasets generated using an IBM data generator introduced in [1]. The parameters are set as follows:

$|D|$: Number of customers;

$|C|$: Average number of transactions per customer;

$|T|$: Average number of items per transaction;

$|S|$: Average length of maximal sequences;

$|I|$: Average length of itemsets of maximal sequences;

$|N|$: Number of distinct items.

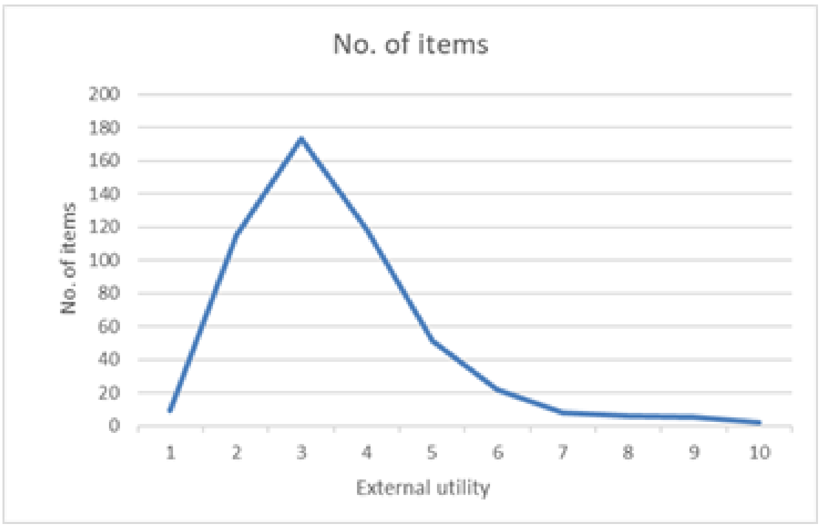
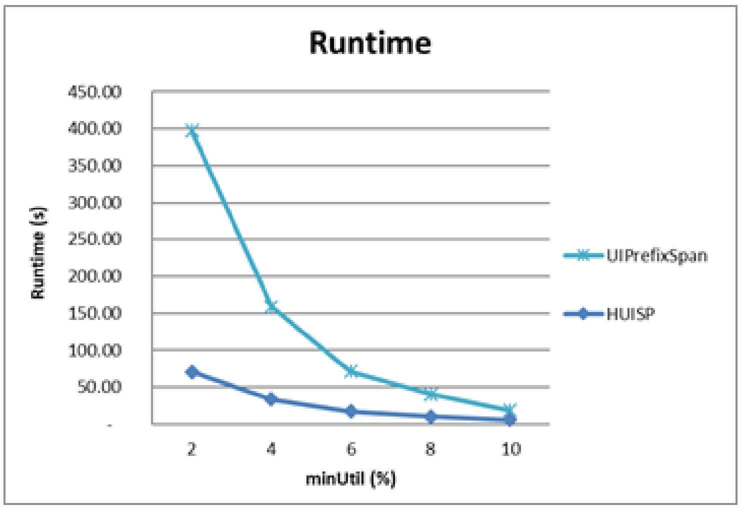
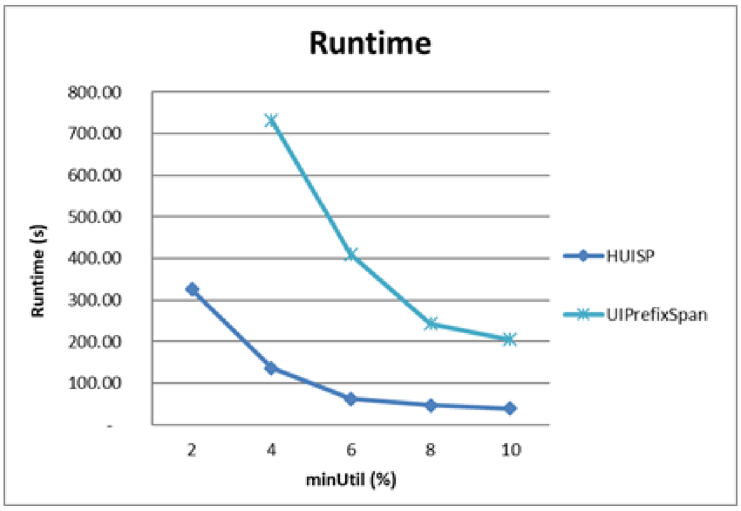


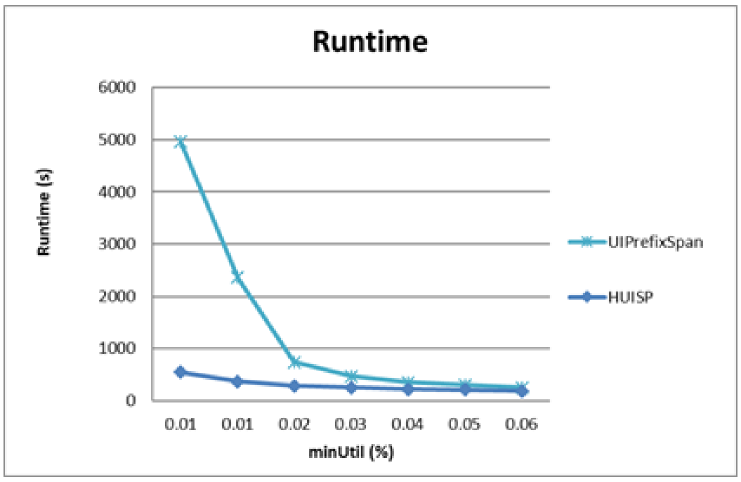
Figure 1. External utility distribution for 1000 items using log-normal distribution



a) *D10K.C9.T8.S7.I8.N1K*

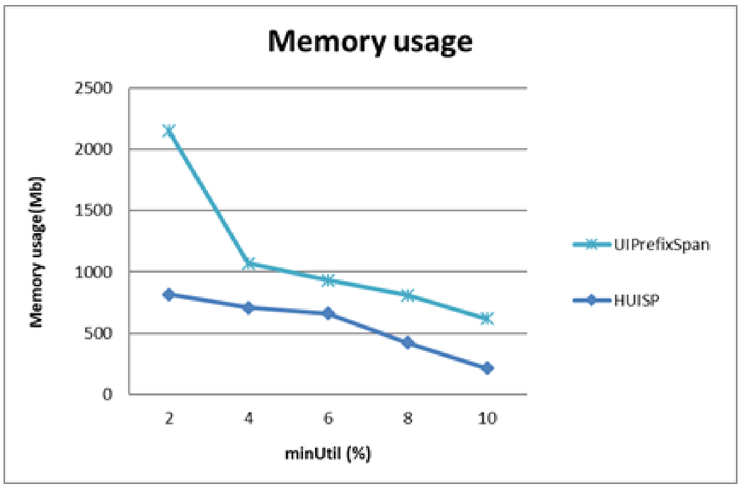


b) *D200K.C10.T9.S9.I7.N1K*

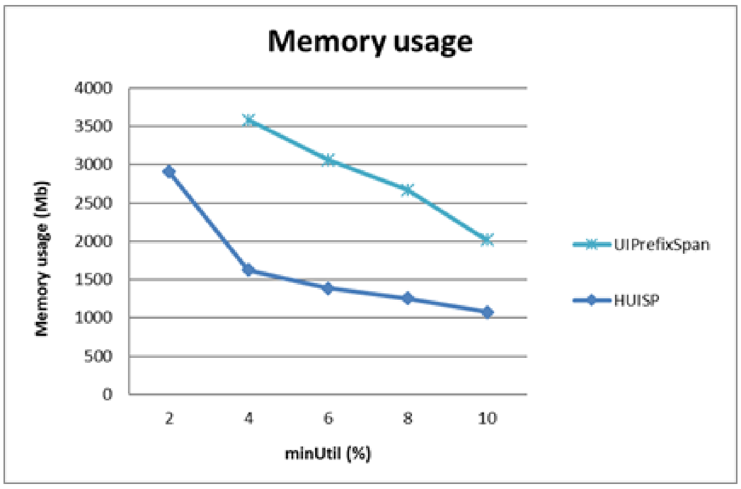


c) *Bible*

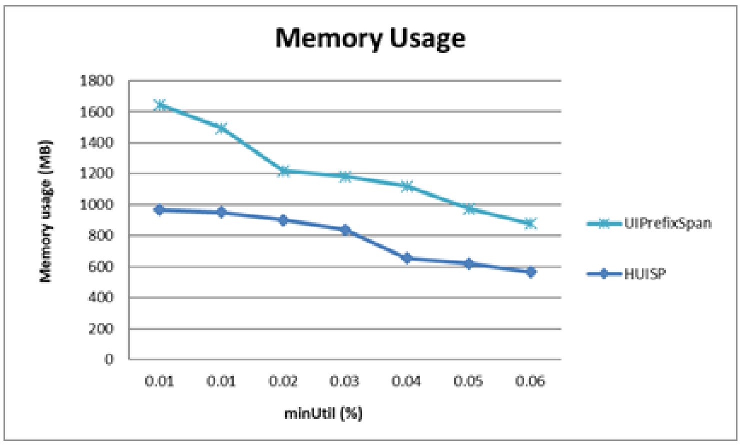
Figure 2. Runtime



a) *D10K.C9.T8.S7.I8.N1K*



b) *D200K.C10.T9.S9.I7.N1K*



c) *Bible*

Figure 3. Memory usage

We generate 2 synthetic datasets

D10K.C9.T8.S7.I8.N1K (DS1) and D200K.C10.T9.S9.I7.N1K (DS2).

We also use a real-life dataset, Bible (`http://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php`), which contains 36369 sequences and 13905 distinct items. The average length of a sequence is 21.6. The average number of distinct items per sequence is 17.84.

To fit the problem of high utility sequential pattern mining, we generate the quantities of the items in the database which are ranged from 1 to 5. With each dataset, we also generate a profit table with profit values ranged from 1 to 10 using a log-normal distribution. Figure 1 shows the profit distribution of 1000 items in DS1 and DS2:

We generate occurrence time to each dataset according to itemsets' order. It means in each sequence, the first itemset has occurrence time 0, the second itemset has occurrence time 1, the third itemset has occurrence time 2, and so on.

5.2. Performance evaluation

Figure 2 and Figure 3 show execution time and memory usage of the two algorithms UIPrefixSpan and HUISP, respectively. With DS1 and DS2, different *minSeqUtil* from 2% to 10 % are used and item interval constraints are set to $C_1 = 3,\ C_2 = 15,\ C_3 = 5,\ C_4 = 30$. With Bible dataset, different minSeqUtil from 0.08% to 0.02% are used and item interval constraints are set to $C_1 = 0,\ C_2 = 5,\ C_3 = 0,\ C_4 = 15$.

Figure 2 and Figure 3 show the two algorithm's performance in term of runtime and memory usages. As shown in the figures, HUISP has better performance compared with UIPrefixSpan. By using a utility table and index table, HUISP performs in one phase instead of two phases as in UIPrefixSpan. Moreover, HUISP removes un-potential items out of project database after each recursion, so upper bound values are reduced and that this makes search space reduced and thus improves the algorithms performance.

6. CONCLUSIONS

We propose HUISP, an algorithm to discover the high utility sequential patterns with item interval using pattern growth approach. The algorithm uses some efficient techniques to improve the algorithm's performance. First, we use a utility table which saves patterns' utilities during the mining process. This makes HUISP performing in one phase instead of two phases as in our pervious algorithm UIPrefixSpan. Second, index table is designed to quickly find the relevant quantitative sequences for prefixes to be processed in the recursive process. Finally, by using pruning un-potential items strategy, the upper bound of utilities is lower and lots of low profit candidate subsequence can then be avoided.

Our algorithm is one of the algorithms for mining the item interval patterns. With four item interval constraints, HUISP helps to reduce the candidate patterns when compared with other algorithms without item interval constraints. Thus, HUISP can find more meaningful patterns. With above comments, we can conclude that HUISP is an efficient algorithm for mining high utility sequential patterns with item interval.

ACKNOWLEDGMENT

This work is sponsored by a research grant from IOIT (CS19.08).

REFERENCES

[1] R. Agrawal, R. Srikant, "Mining sequential patterns," in *Proceedings of the Eleventh International Conference on Data Engineering Date of Conference.* Taipei, Taiwan: IEEE, March, 6–10, 1995. DOI: 10.1109/ICDE.1995.380415

[2] J. Pei, J. Han, B.M. Asi, H. Pino, "PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth," in *Proceedings 17th International Conference on Data Engineering.* Heidelberg, Germany: IEEE, April 2–6, 2001. DOI: 10.1109/ICDE.2001.914830

[3] J. Ayres, J. Gehrke, T. Yiu, and J. Flannick, "Sequential pattern mining using bitmap representation," in *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining,* July 2002 (pages 429–435) https://doi.org/10.1145/775047.775109

[4] M. Zaki, "SPADE: An efficient algorithm for mining frequent sequences," *Machine Learning*, vol. 40, pp.31–60, 2000. https://doi.org/10.1023/A:1007652502315

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

[6] Yin, J., Zheng, Z., Cao, L, "USpan: an efficient algorithm for mining high utility sequential patterns," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data MiningAugust,* 2012 (pages 660-668). https://doi.org/10.1145/2339530.2339636

[7] G.C. Lan, T.P. Hong, V.S. Tseng, S.L. Wang, "Applying the maximum utility measure in high utility sequential pattern mining," *Expert Systems with Applications,* vol. 41, no. 11, p. 5071-5081, 2014.

[8] Alkan, O. K. and Karagoz, P., "CRoM and HuspExt: Improving efficiency of high utility sequential pattern extraction," in *IEEE Transactions on Knowledge and Data Engineering,* vol. 27, no. 10, pp. 2645–2657, 2015. Doi: 10.1109/TKDE.2015.2420557

[9] J.Z. Wang, J.L. Huang, Y.C. Chen, "On efficiently mining high utility sequential patterns," in *Knowl. Inf. Syst,* vol. 49, no. 2, p. 597-627, 2016.

[10] Y.-L. Chen, T.C.-H. Huang, "Discovering time-interval sequential patterns in sequence databases," *Expert Systems with Applications*, vol. 25, no. 3, p. 343–354, 2003.

[11] Y.-L. Chen, M.-C. Chiang, and M.-T. Ko, "Discovering fuzzy time-interval sequential patterns in sequence databases," *IEEE Transactions on Systems Man and Cybernetics*, vol. 35, no. 5, pp. 959–972, 2005.

[12] Yu Hirate, Hayato Yamana, "Generalized sequential pattern mining with item," *Journal of Computers*, vol. 1, no. 3, pp.51–60, 2006.

[13] Tran Huy Duong, Vu Duc Thi, "Algorithm mining normalized weighted frequent sequential patterns with Time intervals," *Research, Development and Application on Information & Communication Technology*, vol. V-2, no. 34, pp. 72–81, 2015. https://ictmag.vn/cntt-tt/article/view/191/pdf

[14] Tran Huy Duong, Tran The Anh, Nguyen Tien Thuy, "An algorithm for mining high utility sequential patterns with time interval," in *Proceedings of 20th Vietnam National Conference,* Quy Nhon, November 23–24, 2017.

[15] A. Sirisha, Suresh Pabboju, G. Narsimha, "An approach to mine Time Interval based Weighted Sequential Patterns in Sequence Databases," *2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS),* Dec. 4–7, 2017. Doi: 10.1109/SITIS.2017.16

[16] Truong Duc Phuong, Do Van Thanh and Nguyen Duc Dung, 'Mining fuzzy sequential patterns with fuzzy time-intervals in quantitative sequence databases," *Cybernetics and Information Technologies*, vol. 18, no. 2, pp. 3-19, 2018. Doi: 10.2478/cait-2018-0024

[17] Truong-Chi T., Fournier-Viger P, "A Survey of High Utility Sequential Pattern Mining," in *High-Utility Pattern Mining. Studies in Big Data*, vol 51. Springer, Cham, January 19, 2019. https://doi.org/10.1007/978-3-030-04921-8_4

Received on September 06, 2019
Revised on October 18, 2019