Generalized Sequential Pattern Mining with Item Intervals

Yu Hirate
Media Network Center, Waseda University, Tokyo Japan
Email: hirate@acm.org

Hayato Yamana
Science and Engineering, Waseda University, Tokyo, Japan
Email: yamana@acm.org

***Abstract*— Sequential pattern mining is an important data mining method with broad applications that can extract frequent sequences while maintaining their order. However, it is important to identify item intervals of sequential patterns extracted by sequential pattern mining. For example, a sequence $< A, B >$ with a 1-day interval and a sequence $< A, B >$ with a 1-year interval are completely different; the former sequence may have some association, while the latter may not. To adopt item intervals, two approaches have been proposed for integration of item intervals with sequential pattern mining; (1) constraint-based mining and (2) extended sequence-based mining. However, although constraint-based mining approach avoids the extraction of sequences with non-interest time intervals such as too long intervals it has setbacks in that it is difficult to specify optimal constraints related to item interval, and users must re-execute constraint-based algorithms with changing constraint values. On the other hand, extended sequence-based mining approach does not need to specify constraints and re-execute. Since extended sequence-based mining approach cannot adopt any constraints based on time intervals, it may extract meaningless patterns, such as sequences with too long item intervals. This means these two approaches have not only advantages but also disadvantages. To solve this problem, in this paper, we generalize sequential pattern mining with item interval. The generalization includes three points; (a) a capability to handle two kinds of item interval measurement, item gap and time interval, (b) a capability to handle extended sequences which are defined by inserting pseudo items based on the interval itemization function, and (c) adopting four item interval constraints. Generalized sequential pattern mining is able to substitute all types of conventional sequential pattern mining algorithms with item intervals. Using Japanese earthquake data, we have confirmed that our proposed algorithm is able to extract sequential patterns with item interval, defined in a flexible manner by the interval itemization function.**

***Index Terms*— Data Mining, Sequential Pattern Mining, Item Intervals, Gap, Time-stamp**

I. INTRODUCTION

Due to recent developments in network infrastructure and both price reduction and increases in capacity of storage devices, it has become commonplace to archive large amounts of data. It is important to analyze such large data sets because they may contain new knowledge. The discovery of such knowledge requires data mining technology.

Sequential pattern mining, which is one of the most important of these technologies, extracts patterns that appear more frequently than a user-specified minimum support while maintaining their item occurrence order [1]. Many sequential pattern mining algorithms, such as GSP [2], PrefixSpan [3] [4],SPADE [5], and SPAM [6], have been proposed.

These sequential pattern mining algorithms [2]–[6], however, consider only the item occurrence order, but do not consider the item intervals between successive items. Thus, it is impossible to identify the item intervals between successive items extracted as frequent sequential patterns. For example, one customer may buy product-A, and then buy product-B 1 day later, while another customer may buy product-A, and then buy product-B 1 year later. Although these two customers' actions are completely different with regard to item interval, these sequential pattern mining algorithms would treat the two customers' actions as the same action. It is useful to be able to distinguish these customers' actions to understand not only what events will follow, but also when these events will occur.

To adopt item intervals, two approaches have been proposed for integration of item intervals with sequential pattern mining; (1) constraint-based mining [7] [8], and (2) extended sequence-based mining [9]–[12].

Constraint-based mining algorithms extract frequent sequences that satisfy both user-specified minimum support constraints and user-specified constraints [7] [8]. For example, when users set the maximal time interval to 1 day, they extract frequent sequences whose items occur within 1 day.

Although constraint-based mining algorithms escape extraction of sequences with non-interest time intervals such as too long intervals, however, it is difficult for users to specify an optimal constraint related to item

This paper is based on "Sequential Pattern Mining with Time Interval," by Y. Hirate, and H. Yamana, which appeared in the Proceedings of the 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2006), Singapore, April 9-12, 2006.

interval, and so the user must re-execute the algorithm after changing the constraint value. This results in a decrease in usability.

To extract frequent sequences consisting of the same items but satisfying different item interval constraints by one time execution, extended sequence-based mining algorithms have been proposed [9]–[12]. These algorithms extract frequent sequential patterns with item intervals by converting item intervals to pseudo items. However, as they cannot adopt any constraints related to item intervals, they may extract meaningless patterns, such as sequences with too long item intervals. For example, a sequence $< A, B >$ with a 1-year interval may not have any association in most cases. This means that extended sequence based mining should adopt some constraints which are related to time interval.

To solve problems of both constraint-based mining and extended sequence-based mining, in this paper, we generalize sequential pattern mining with item interval. The generalization includes three points; (a) a capability to handle two kinds of item-interval measurement, item gap and time interval, (b) a capability to handle extended sequences which are defined by inserting pseudo items based on the interval itemization function, and (c) adopting four item-interval constraints. Generalized sequential pattern mining is able to substitute all types of conventional sequential pattern mining algorithms with item intervals.

The remainder of this paper is organized as follows. In section II, we introduce sequential pattern mining and its algorithm PrefixSpan [3] [4]. Section III describes related work. In section IV, we propose "Generalized Sequential Pattern Mining with Item Intervals." Section V describes the proposed algorithm. Evaluation is presented in Section VI, and Section VII presents our conclusions.

II. Conventional Sequential Pattern Mining

In this section, we introduce sequential pattern mining [1] and the PrefixSpan [3] [4] algorithm, on which our proposed algorithm is based.

A. Sequential Pattern Mining

Let $I = \{i_1, i_2, \cdots, i_n\}$ be a set of all items. An itemset X is a subset of items sorted alphabetically. Sequence s is a list of itemsets sorted in ascending order according to their occurrence time. Sequence s is defined as $s =< X_1, X_2, \cdots, X_m >$. A sequential database SDB, which is a target of pattern extraction, is a set consisting of sequence s, and is defined as $SDB = \{s_1, s_2, \cdots, s_t\}$.

Here, two sequences, $\alpha =< X_1, X_2, \cdots, X_p >$ and $\beta =< X'_1, X'_2, \cdots, X'_q > (p \leq q)$, are given. We say that β includes α iff $X_i \subset X'_i$ for all $\{i|1 \leq i \leq p\}$. The support of sequence α in SDB, denoted as $sup_{SDB}(\alpha)$, is the percentage of the sequences that include α in SDB. A frequent sequence is defined as a sequence whose support is higher than the user-specified minimum support, denoted as $min\_sup (0 \leq min\_sup \leq 1)$. Given SDB and min_sup, sequential pattern mining extracts all the frequent sequences.

TABLE I.
Example of SDB

SID	s(=sequence)
10	$< a, (abc), (ac) >$
20	$< (ad), c >$
30	$< (ef), (ab) >$

For example, given the SDB shown in TABLE I. ,and set the min_sup equal to 0.5, five frequent sequences $< a >$, $< b >$, $< c >$, $< a, c >$, and $< (ab) >$, are extracted.

B. PrefixSpan Algorithm

Pei *et al.* proposed PrefixSpan in 2000 as a fast sequential pattern mining algorithm [3] [4]. PrefixSpan extracts frequent sequences with depth-first search by executing SDB projection operations recursively. PrefixSpan consists of SDB projection operation, which is the most important process in the algorithm. The SDB projection operation can be explained as follows.

Let $\alpha =< X_1, X_2, \cdots, X_p >$. When there exists an integer j $(1 \leq j \leq p)$ that satisfies $X_n \subseteq X_j$ and $X_n \nsubseteq X_k$ for all k $(1 \leq k \leq j-1)$, we define sequence $< X_1, X_2, \cdots, X_{j-1}, X_n >$ as a prefix of α with regard to X_n, denoted as $Prefix(\alpha, X_n)$. The remaining sequence $< X'_j, X_{j+1}, \cdots, X_{p-1}, X_p >$ is defined as a postfix of α with regard to X_n, denoted as $Postfix(\alpha, X_n)$, where X'_j is the subset of X_j from which all items also including in X_n are excluded. On the other hand, when there exists no integer j, both $Prefix(\alpha, X_n)$ and $Postfix(\alpha, X_n)$ are defined as ϕ.

Then, let $\beta =< X_1, X_2, \cdots, X_{p-1}, X_p >$, $(p \geq 1)$ and t be any sequence in SDB. The β-projected database, denoted as $SDB|\beta$, is a collection of sequences s and defined as follows.

$$SDB|\beta = \{s|s = Postfix(t, \beta) \wedge s \neq \phi\} \quad (1)$$

The β-projected database is a collection of projected sequences, which are postfixes of the sequence included in SDB, with regard to β, and β is called a projection sequence of $SDB|\beta$.

As an example, we describe how PrefixSpan extracts frequent sequences from SDB shown in TABLE I, with $min\_sup = 0.5$. Figure 1 shows the projection operation process in this example.

1) Scan SDB once to find all frequent items; they are $< a >$, $< b >$, and $< c >$, where $sup_{SDB}(< a >) = 3$, $sup_{SDB}(< b >) = sup_{SDB}(< c >) = 2$.
2) Generate $SDB| < a >$, then scan $SDB| < a >$ to find all the frequent sequences with a single item. Frequent sequences with a single item in $SDB| < a >$ are $< c >$ and $< (\_b) >$. This means that $< a, c >$ and $< (ab) >$ are frequent sequences.

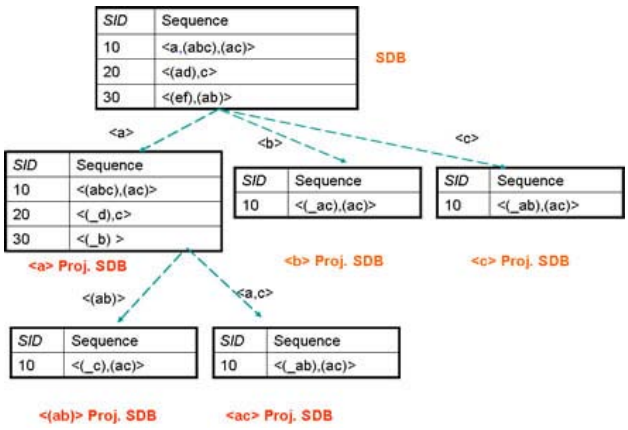

Figure 1. Running Example of PrefixSpan Algorithm

3) Generate $SDB| < a, c >$; however, there is no frequent sequence with a single item in $SDB| < a, c >$.
4) Generate $SDB| < (ab) >$; however, there is no frequent sequence with a single item in $SDB| < (ab) >$.
5) Terminate extraction of frequent sequences containing $< a >$ as a prefix. Then, initiate extraction of frequent sequences that include $< b >$ as a prefix.
6) However, there are no frequent sequences with a single item in $SDB| < b >$. Terminate extraction of frequent sequences containing $< b >$ as a prefix. Then, initiate extraction of frequent sequences that include $< c >$ as a prefix.
7) As there are no frequent sequences with a single item in $SDB| < c >$, terminate extraction of frequent sequences that include $< c >$ as a prefix.
8) Terminate frequent sequence generation.

III. Related Work

In this section, we introduce conventional algorithms for mining sequential patterns with item intervals.

A. Two measurement methods to represent item intervals

Item intervals are represented in two ways; item gap and time interval. Item gap is defined as the number of items between successive items, and time interval is defined as the length of time between the occurrence times of successive items.

Generally, item gap measurement has the advantage of being applicable to datasets the items of which have no occurrence time information. Thus, sequential pattern mining with item gap works well when applied to datasets the item intervals of which can be defined as the number of items, such as DNA sequences, and datasets the items of which occur at fixed time intervals. However, it does not work well when applied to datasets in which the items occur at irregular time intervals.

On the other hand, time interval measurement has the advantage of allowing the extraction of frequent sequential patterns with time interval information. When applied to datasets for the items of which occurrence time information is available, sequential pattern mining with time intervals can extract more precise patterns than sequential pattern mining with item gaps.

B. Two approaches to count support value

There exist two approaches to count support value; the item constraint approach and extended sequence approach. The item constraint approach involves extraction of sequences satisfying not only a user-specified minimum support constraint, but also user-specified constraints, such as maximum/minimum item intervals. The extended sequence approach extends sequences by inserting pseudo items which represent item intervals. After extending the original sequences, it extracts frequent sequential patterns from them.

C. Four categories in conventional sequential pattern mining with item intervals

Here, we have four categories–two approaches using two measurement methods: the item constraint approach using item gap, item constraint approach using time interval, extended sequence approach using item gap, and extended sequence approach using time interval.

1) Item constraint approach using item gap: Sequential pattern mining with the item constraint approach using item gap extracts sequences satisfying not only user-specified minimum support constraints, but also user-specified gap constraints. The minimum and maximum gap values should be defined as the gap constraint by the user. When counting the support of a sequence, only those sequences with item gaps larger than the minimum gap constraint and also smaller than the maximum gap constraint are counted. For example, let x be any item and the sequence $< A, B >$ exist. Then, the algorithm that belongs to this category counts $< A, B >$ and $< A, x, B >$ as the sequence $< A, B >$. However, it does not count $< A, x, x, B >$ as the sequence $< A, B >$ when the minimum gap is 0 and maximum gap is 1.

Zaki proposed the cSPADE algorithm [7], which belongs to this category, based on the SPADE algorithm [5].

2) Item constraint approach using time interval: Similar to the item constraint approach using item gap, sequential pattern mining with item constraint approach using time interval extracts sequences satisfying not only the user-specified minimum support constraint, but also user-specified time interval constraints. The minimum and maximum time interval values should be defined as the time interval constraints by the users. When counting the support of a sequence, only those sequences the items for which the time intervals are larger than the minimum time interval constraint and smaller than the maximum time interval constraint are counted.

Pei *et al.* proposed a constraint-based sequential pattern mining algorithm [8] based on PrefixSpan [3] [4]. In [8],

many types of constraints are discussed, including item gap constraints and time interval constraints.

3) Extended sequence approach using item gap: In this category, item interval is represented by the number of items between successive items. When counting the support of a sequence, only the sequence, whose item gap is same, is counted. For example, let x be any item. Then, the algorithm belonging to this category treats $< A, B >$, $< A, x, B >$ and $< A, x, x, B >$ as different sequences and counts their frequencies individually.

Kitakami et al. proposed Modified PrefixSpan [9], as an algorithm belonging to this category, based on PrefixSpan [3] [4].

4) Extended sequence approach using time interval: In this category, item interval is represented by pseudo items. Pseudo items are converted by user-defined time interval segmentation. After extending the original sequences by inserting pseudo items, it extracts frequent sequential patterns from them. When counting the support of a sequence, only the sequence, in which the pseudo item between successive items is same, is counted. For example, let x and y be a pseudo item that represents a user-specified time unit. Then, the algorithm belonging to this category treats $< A, B >$, $< A, x, B >$, and $< A, y, B >$ as different sequences and counts their frequencies individually.

Chen *et al.* proposed two algorithms, I-Apriori, I-PrefixSpan belonging to this category [10] [11]. I-Apriori is based on Apriori Algorithm [13], and I-PrefixSpan is based on PrefixSpan [3] [4]. We proposed an algorithm [12] belonging to this category, based on PrefixSpan [3] [4]. The difference between [10] [11] and [12] is how to define time interval segmentation.

D. Toward generalization

As described above, there are two approaches in sequential pattern mining with item intervals. By using constraint-based mining approach, it is able to avoid the extraction of sequences with users' non-interest item intervals. However, as it is difficult for users to specify optimal constraints using the item constraint approach, they must re-execute algorithms with changes in constraint values. This results in a decrease in usability. At this point, the extended sequence approach has an advantage compared to the item constraint approach because users do not have to specify the maximum or minimum intervals.

Even with extended sequence-based mining approach, however, conventional sequence-based mining methods cannot handle any constraints related to item intervals. Thus, users may receive meaningless patterns, such as sequences with too long item intervals.

As we described above, every four categories have advantages and disadvantages. Our proposed scheme, called generalized sequential pattern mining, is able to substitute all The categories of conventional sequential pattern mining algorithms with item intervals.

IV. GENERALIZED SEQUENTIAL PATTERN MINING WITH ITEM INTERVALS

In this section, we propose a new sequential pattern mining algorithm called "generalized sequential pattern mining with item intervals". The generalization includes three points; (1) a capability to handle two kinds of item interval measurement, item gap and time interval, (2) a capability to handle extended sequences which are defined by inserting items based on the interval itemization function, and (3) adopting four item interval constraints. Generalized sequential pattern mining is able to substitute all types of conventional sequential pattern mining algorithms with item intervals. As a result, our scheme generalizes the four categories of sequential pattern mining algorithms described in Section III into one.

A. Interval Extended Sequence

To add item intervals to sequential patterns, we define the interval extended sequence, denoted as is, by extending the original sequence s. The interval extended sequence is a list of items with item intervals, and is defined as follows:

$$\begin{aligned} is \;\; = \;\; & < (t_{1,1}, X_1), (t_{1,2}, X_2), (t_{1,3}, X_3), \\ & \cdots, (t_{1,m}, X_m) > \end{aligned} \tag{2}$$

Here, $X_i (1 \leq i \leq m)$ are items, and $t_{\alpha,\beta}$ is the item interval between X_α and X_β. When the datasets have item occurrence time information, such as time-stamp, $t_{\alpha,\beta}$ becomes the time interval and is defined by the following equation:

$$t_{\alpha,\beta} = X_\beta.time - X_\alpha.time \tag{3}$$

Here, $X_\alpha.time$ and $X_\beta.time$ are transaction occurrence times of X_α and X_β, respectively. On the other hand, when the datasets do not have item occurrence time information, $t_{\alpha,\beta}$ may become an item gap and is defined by the following equation:

$$t_{\alpha,\beta} = \beta - \alpha \tag{4}$$

B. Frequent Interval Extended Sequences

The item interval extended sequential database, $ISDB$, which is the target of pattern extraction, is a set of interval extended sequences is and is defined as $ISDB = \{is_1, is_2, \cdots, is_t\}$.

When two interval extended sequences, $\alpha =< (t_{1,1}, X_1), (t_{1,2}, X_2), \cdots, (t_{1,m}, X_m) >$, $\beta =< (t'_{1,1}, X'_1), (t'_{1,2}, X'_2), \cdots, (t'_{1,m}, X'_m), \cdots, (t'_{1,n}, X'_n) >$ and an interval itemization function $I(t)$ are given, we say that β includes α iff $X_i \subset X'_i$ for all $\{i | 1 \leq i \leq m\}$ and $I(t_{1,i}) = I(t'_{1,i})$ for all $\{i | 1 \leq i \leq m\}$. The support of interval extended sequence α in $ISDB$, denoted as $sup_{ISDB}(\alpha)$, is the percentage of interval extended sequence that includes α. A frequent interval extended sequence is defined as the interval extended sequence whose support is higher than the user-specified minimum support. Given $ISDB$ and min_sup, generalized sequential pattern mining with item interval extracts all frequent interval extended sequences.

C. Item Interval Constraints

Without any item interval constraints, a set of frequent interval extended sequences may include meaningless frequent interval extracted sequences with regard to item intervals, such as sequences that contain item intervals that are either too long or too short. To exclude such interval extended sequences, item interval constraints should be defined by the users.

We adopt four types of item interval constraint. Let $< (t_{1,1}, X_1), (t_{1,2}, X_2), \cdots, (t_{1,m}, X_m) >$ be an extracted interval extended sequence. The four kinds of constraint are as follows:

- C_1 Let $min\_interval$ be a minimum item interval between any two adjacent items, C_1 is defined as $t_{i,i+1} \geq min\_interval$ for all $\{i | 1 \leq i \leq m-1\}$.
- C_2 Let $max\_interval$ be a maximal item interval between any two adjacent items, C_2 is defined as $t_{i,i+1} \leq max\_interval$ for all $\{i | 1 \leq i \leq m-1\}$.
- C_3 Let $min\_whole\_interval$ be a minimum item interval between the head and tail of the sequence, C_3 is defined as $t_{1,m} \geq min\_whole\_interval$.
- C_4 Let $max\_whole\_interval$ be the maximal item interval between the head and tail of the sequence, C_4 is defined as $t_{1,m} \leq max\_whole\_interval$.

Note that when we specify four types of interval constraint at the same time, the following four expressions must be fulfilled or no frequent interval extended sequences are extracted:

- $min\_interval \leq max\_interval$
- $min\_whole\_interval \leq max\_whole\_interval$
- $min\_interval \leq max\_whole\_interval$
- $max\_interval \leq max\_whole\_interval$

Here, C_1, C_2, and C_4 are categorized anti-monotone constraints, and C_3 is categorized monotone constraint [14]. An anti-monotone constraint satisfies "when a sequence A does not satisfy the constraint, any superset of A also does not satisfy the constraint." A monotone constraint satisfies "when a sequence A satisfies the constraint, any superset of A also satisfies the constraint."

D. Example

Given the interval extended sequential dataset shown in TABLE II., and set an interval itemize function as

$$I(t) = \lfloor \frac{t}{60 \times 60 \times 24} \rfloor \tag{5}$$

and a minimum support as $min\_sup = 0.5$, five interval extended sequences $< (0,a) >, < (0,b) >, < (0,c) >, < (0,a), (3,c) >, < (0,a), (0,b) >$ and $< (0,a), (2,a) >$ are extracted as frequent interval extended sequences. $< (0,a) >$, $< (0,b) >$, $< (0,c) >$ represent item a, b, c occur, respectively. $< (0,a), (0,b) >$ represents item a, b occur at the same time. $< (0,a), (3,c) >$ represents once item a occurs, item c will occur with item interval $(172800, 259200]$. In addition, $< (0,a), (2,a) >$ represents once item a occurs, item a will occur again with item interval $(86400, 172800]$. When $max\_interval$ set to 172800 (C_2), frequent interval extended sequence $< (0,a), (3,c) >$ is not extracted.

TABLE II.
EXAMPLE OF $ISDB$

$iSID$	is(=interval extended sequence)
10	$< (0,a), (86400, abc), (259200, ac) >$
20	$< (0,ad), (259200, c) >$
30	$< (0,aef), (172800, ab) >$

V. ALGORITHM FOR GENERALIZED SEQUENTIAL PATTERN MINING WITH ITEM INTERVALS

To extract frequent interval extended sequences, we propose a novel algorithm based on PrefixSpan [3] [4]. Similar to PrefixSpan, the proposed algorithm extracts frequent interval extended sequences with depth-first search by executing the projection operation recursively. We extend the sequential database projection operation in PrefixSpan to handle both interval extended sequence and item interval constraints. In this section, we describe (1) term definitions, (2) extended projection operation, and (3) the method of applying item interval constraints to the extended projection operation.

A. Term Definition

1) Definition 1: Prefix and Postfix of interval extended sequences:

Let $\alpha = < (t_{1,1}, X_1), (t_{1,2}, X_2), \cdots, (t_{1,m}, X_m) >$ be an interval extended sequence, and X_β be any itemset. When there exists an integer j $(1 \leq j \leq m)$ satisfying $X_\beta \subseteq X_j$ and $I(t_{1,\beta}) = I(t_{1,j})$, we define a prefix of is with regard to X_β,$I(t_{1,\beta})$ as follows:

$$\begin{aligned} &Prefix(\alpha, X_\beta, I(t_{1,\beta})) = \\ &\quad < (t_{1,1}, X_1), (t_{1,2}, X_2), \cdots, (t_{1,j}, X_j) > \end{aligned} \tag{6}$$

Postfix of α with regard to X_β,$I(t_{1,\beta})$ is also defined as follows:

$$\begin{aligned} &Postfix(\alpha, X_\beta, I(t_{1,\beta})) = \\ &\quad < (t_{j,j}, X'_j), (t_{j,j+1}, X_{j+1}), \cdots, (t_{j,m}, X_m) > \end{aligned} \tag{7}$$

where X'_j is the subset of X_j from which all the items also included in X_β are excluded. When $X'_j = \phi$, postfix of α with regard to X_β,$I(t_{1,\beta})$ becomes the following:

$$\begin{aligned} &Postfix(\alpha, X_\beta, I(t_{1,\beta})) = \\ &\quad < (t_{j,j+1}, X_{j+1}), (t_{j,j+2}, X_{j+2}), \cdots, (t_{j,m}, X_m) > \end{aligned} \tag{8}$$

On the other hand, when there exists no integer j, postfix of α with regard to X_β,$I(t_{1,\beta})$ becomes the following:

$$Prefix(\alpha, X_\beta, I(t_{1,\beta})) = \phi \tag{9}$$

$$Postfix(\alpha, X_\beta, I(t_{1,\beta})) = \phi \tag{10}$$

2) Definition 2: Projected interval extended sequence database:

The projected interval extended sequence database is a subset of $ISDB$. Let α be an interval extended sequence that occurs in $ISDB$ at least once. The α-Projected interval extended sequence database is a the collection of postfixes of $ISDB$ with regard to α. In this case, α is called the projection sequence.

3) Definition 3: Projection Level:

Projection level is the number of items included in the projection sequence. For example, let α be an interval extended sequence with l items. Generating $ISDB|\alpha$ is level l projection.

B. Interval extended projection

To extract frequent interval extended sequence, we extended the projection operation of PrefixSpan [3] [4] algorithm.

1) Level 1 Projection:

In the case of level 1 projection, as it is impossible to define item intervals with a single item, our algorithm scans $ISDB$ to extract all the frequent items, similar to PrefixSpan. For every frequent item i, the initial interval extended sequence α is created as $< (0, i) >$. Then, $ISDB|\alpha$ is created. After that, level 2 projection operation is executed. Note that when interval extended sequence α appears more than once in the same is, our algorithm generates multiple prefixes and postfixes at each α, and then treats them as different interval extended sequences. For example, for a sequence $< (0, a), (86400, abc), (172800, ac) >$, the projection result with projection sequence $< (0, a) >$ becomes three interval extended sequences, $< (86400, abc), (172800, ac) >$, $< (0, bc), (86400, ac) >$, and $< (0, c) >$.

2) Level 2 or later Projection:

In the case of level 2 or later projection, our algorithm scans projected $ISDB$ and counts the support of interval extended sequences. Let α be an interval extended sequence. Our algorithm counts the number of all possible pairs $(I(t_{1,i}), a)$ included in $ISDB|\alpha$ where $a \in X_i$. Then, for every pair satisfying the minimum support constraint, β is defined as $Prefix(\alpha, a, I(t_{1,i}))$, then $ISDB|\beta$ is generated. Formally, $ISDB|\beta$ is defined by the following equation:

$$\begin{aligned} ISDB|\beta \quad = \quad & \{is \mid is \neq \phi \wedge is = Postfix(\gamma, a, I(t_{1,i})) \\ & \wedge sup_{ISDB|\alpha}(I(t_{1,i}), a) \geq min\_sup\} \end{aligned} \quad (11)$$

where $\gamma \in ISDB|\alpha$, $a \in X_i$.

C. Applying item interval constrains

1) Adapting Anti-monotone Constraint:

To adapt anti-monotone constraints, C_1, C_2, and C_4, our algorithm checks these constraints at every projection. Formally, the projected interval extended sequence database is defined as follows:

$$\begin{aligned} ISDB|\beta \quad = \quad & \{is \mid is \neq \phi \wedge is = Postfix(\gamma, a, I(t_{1,i})) \\ & \wedge sup_{ISDB|\alpha}(I(t_{1,i}), a) \geq min\_sup \\ & \wedge \beta \ satisfies \ C_1, C_2, C_4\} \end{aligned} \quad (12)$$

where $\gamma \in ISDB|\alpha$, $a \in X_i$.

2) Adapting Monotone Constraint:

To adapt monotone constraint C_3, our algorithm checks whether extracted frequent interval extended sequences satisfy C_3 or not, after they have been extracted with satisfying minimum support constraint, C_1, C_2, and C_4. This means constraint C_3 is not taken care at $ISDB$ projection. This is because that we are not able to judge the satisfaction of constraint C_3 at $ISDB$ projection. Although an interval extended sequence δ does not satisfy the constraint C_3, some supersets ε, which include δ as a subset, may satisfy the constraint C_3. Note that generating $ISDB|\varepsilon$ is executed after generating $ISDB|\delta$.

When a candidate extracted sequence satisfies C_3, it is extracted as a result sequence. On the other hand, when a candidate extracted sequence does not satisfy C_3, it is not extracted as a result sequence.

D. Algorithm

Based on the above description, our proposed algorithm is shown below.

INPUT $ISDB$,$I(t)$,min_sup,C_1,C_2,C_3,C_4[1]
OUTPUT R(=frequent sequences satisfying constraints)
METHOD

1) Set is as ϕ.
2) Set R as ϕ.
3) Scan $ISDB$, and find frequent items with higher than min_sup. For all frequent items i,
   a) Define $is =< (0, i) >$, then $R = \{R, is\}$.
   b) Execute $R = projection(ISDB|is, R, I(t), min\_sup, C_1, C_2, C_3, C_4)$.
4) Output R.

E. Projection Algorithm

The projection routine is shown below. It computes level 2 or later projection.

INPUT $ISDB|is$,R,$I(t)$,min_sup,C_1,C_2,C_3,C_4
OUTPUT R
METHOD

1) Scan $ISDB|is$ to find all pairs of item and its itemized interval, denoted as $(\Delta t, i)$, that satisfy min_sup, C_1, and C_2.
2) Define $is =< is, (\Delta t, i) >$.
3) Check is whether satisfies C_4 or not.
4) Only when is satisfies C_4,
   a) Execute $R = projection(ISDB|is, R, I(t), min\_sup, C_1, C_2, C_3, C_4)$.
   b) When is satisfies C_3, $R = \{R, is\}$.
5) Return R.

[1]Constraints C_1,C_2,C_3,C_4 are optional input parameters. Users do not have to specify any of the four constraints.

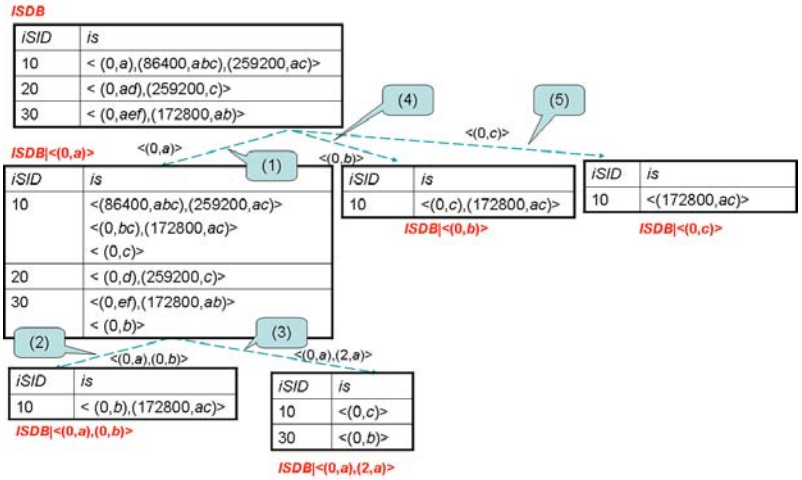

Figure 2. Running Example of Proposed Algorithm

F. Running Example of Proposed Algorithm

We describe a running example of our algorithm from $ISDB$ shown in Table II with $min\_sup = 0.5$, $I(t) = \lfloor \frac{t}{86400} \rfloor$, and $max\_interval = 172800(C_2)$. Figure 2 shows $ISDB$ projection operation process in this example.

1) Scan $ISDB$ once to find all the frequent items in $ISDB$; they are $< a >$,$< b >$, and $< c >$. Then, create initial interval extended sequences: $< (0, a) >$, $< (0, b) >$, $< (0, c) >$. These three is are appended to R.
2) Select the most frequent item a, execute projection operation with regard to $< (0, a) >$, then generate $ISDB| < (0, a) >$ (process (1) in Figure 2).
3) Scan $ISDB| < (0, a) >$ to find all the pairs that satisfy the minimum support constraint and C_2; they are $(0, b)$, $(2, a)$[2].
4) Append $< (0, a), (0, b) >$, $< (0, a), (2, a) >$ to R.
5) Generate $ISDB| < (0, a), (0, b) >$ and scan it to find all the item and itemized interval pairs that satisfy both the minimum support constraint and C_2. However, there are no frequent pairs in $ISDB| < (0, a), (0, b) >$, then terminate extracting frequent is that contain $< (0, a), (0, b) >$ as a prefix (process (2) in Figure 2).
6) Generate $ISDB| < (0, a), (2, a) >$ and scan it to find all the item and itemized interval pairs that satisfy both the minimum support constraint and C_2. However, there are no frequent pairs in $ISDB| < (0, a), (2, a) >$, then terminate extracting frequent is that contain $< (0, a), (2, a) >$ as a prefix (process (3) in Figure 2).
7) Terminate extraction frequent is that contain $< (0, a) >$ as a prefix.
8) Select second most frequent item b[3], execute projection operation with regard to $< (0, b) >$, then generate $ISDB| < (0, b) >$ (process (4) in Figure 2).
9) Scan $ISDB| < (0, b) >$ to find all the item and itemized interval pairs that satisfy the minimum support constraint and C_2. However, there are no frequent pairs in $ISDB| < (0, b) >$, then terminate extracting frequent is that contain $< (0, b) >$ as a prefix.
10) Select third most frequent item c, execute projection operation with regard to $< (0, c) >$, then generate $ISDB| < (0, c) >$ (process (5) in Figure 2).
11) Scan $ISDB| < (0, c) >$ to find all the item and itemized interval pairs that satisfy the minimum support constraint and C_2. However, there are no frequent pairs in $ISDB| < (0, c) >$, then terminate extracting frequent is that contain $< (0, c) >$ as a prefix.
12) Output frequent interval extended sequences, R, then terminate the algorithm.

VI. EVALUATION

In this section, we present evaluations of our proposed algorithm. As datasets, we used a Japanese earthquake dataset [15]. First, we introduce the properties of the Japanese earthquake dataset and our evaluation environment. Then, we report three evaluations based on the Japanese earthquake dataset.

The first evaluation is a comparison of the quality of extracted sequences between conventional sequential pattern mining algorithms and the proposed algorithm. The second evaluation is a comparison of the number of extracted sequences with/without adapting item interval constraints. The third evaluation is a comparison of computation time between conventional sequential pattern mining algorithms and the proposed algorithm.

A. Japanese earthquake dataset

The Japanese earthquake dataset is distributed via K-net [15] provided by the National Research Institute of

[2]The pair $(3, c)$ satisfies minimum support constraint, but dose not satisfy item constraint C_2

[3]Item b and c have the same support value. In this case, we select items according to lexical order.

TABLE III.
EXAMPLE OF EXTRACTED FREQUENT is

Extracted frequent is	Support(%)
$< (0, H) >$	69.231
$< (0, H), ((0, 1day], H) >$	15.385
$< (0, H), ((0, 1day], H), ((0, 1day], H) >$	6.154
$< (0, H), ((1day, 2days], H) >$	6.154

TABLE IV.
EXAMPLE OF EXTRACTED FREQUENT s

Extracted frequent s	Support(%)
$< H >$	69.231
$< H, H >$	52.301
$< H, D >$	21.539

Earth Science and Disaster Prevention, and includes data from 3,296 earthquakes that occurred from May 1995 to December 2003. The earthquake data have 5 fields: latitude of the epicenter, longitude of the epicenter, depth of the epicenter, magnitude on the Richter scale, and time-stamp. The original earthquake data were preprocessed as follows.

1) Only the earthquake data occurred in Japan were used.
2) Japan was divided into a 1-degree latitude and 1-degree longitude square grid.
3) A sequence was defined as a list of earthquakes occurring in the same square in the grid.
4) All earthquake data were itemized to the same item referring to depth of the epicenter and magnitude on the Richter scale. Note that magnitude m was divided by "$m < 2$", "$2 \leq m < 4$", "$4 \leq m < 6$", and "$m \geq 8$" and depth of the epicenter d was divided by "$d < 10km$", "$10km \leq d < 100km$", and "$d \geq 100km$."

B. Evaluation Environment

All the experiments were performed on a 3.2 GHz Pentium 4 PC with 1 GB of main memory, running Fedora Core 3 Linux kernel version 2.6.9. All the programs were written in C++ and compiled with gcc3.4.2.

C. Comparison of Extract Sequence Quality

Table III shows part of the extracted frequent interval extended sequences with interval itemization function $I(t) = \lfloor \frac{t}{86400} \rfloor$[4] based on our proposed algorithm, and Table IV shows apart of the frequent sequences extracted using the conventional sequential pattern mining algorithm, PrefixSpan [3] [4].

Note that in Tables III and IV, item-H represents earthquakes of magnitude between 4.0 and 6.0, with epicenter depths of 10–100 km. Item-D represents earthquakes of magnitude between 2.0 and 4.0, with epicenter depths of 0–10 km.

[4]This interval itemization function is driven by rounding off "t(sec) is divided by 1 day(=$60 \times 60 \times 24$sec)."

TABLE V.
EXAMPLE OF EXTRACTED FREQUENT is

Extracted frequent s	Support(%)
$< (0, H), ((0, 1hour], H) >$	10.769
$< (0, H), ((1hour, 2hours], H) >$	3.077
$< (0, H), ((8hours, 16hours], H) >$	6.154
$< (0, H), ((256hours, 512hours], H) >$	3.077

Calculating the confidence based on the support of extracted is shown in Table III yields the following knowledge. Once item-H occurs, item-H will occur again:

- within 1 day with probability of $\frac{15.385}{69.231} \times 100 = 22.2\%$.
- within 1 to 2 days with probability of $\frac{6.154}{69.231} \times 100 = 8.9\%$.

On the other hand, as shown in Table IV, there is no item interval information in extracted sequences using conventional sequential pattern mining. Thus, users are not able to predict when item-H will occur again. These results indicate that the patterns extracted by the proposed algorithm are more useful than those extracted by the conventional sequential pattern mining algorithm.

Next, to extract more detailed sequences with short time periods, such as within 1 day, we can use a new interval itemization function $I(t)$ to zoom in on the itemizing scale for short periods but zoom out for long periods. For example, when we define the interval itemization function as

$$I(t) = \lfloor \log_2 \frac{t}{3600} \rfloor$$

the interval extended sequences shown in Table V are extracted.

As shown in Table V, as users are able to specify the interval itemization function freely, they can receive frequent interval extended sequences with item intervals that they want to extract.

D. Comparison of the Number of Extracted Sequences

Table VI shows the number of extracted interval extended sequences with varying C_1, C_2, C_3, and C_4. In this evaluation, min_sup was set to 0.03, and the interval itemization function was set to $I(t) = \lfloor \frac{t}{86400} \rfloor$. Note that "–" in Table VI indicates no value was specified as the concerned constraint.

As shown in Table VI, the number of extracted interval extended sequence decreases with specification of the item interval constrains. This means that users are able to discard meaningless interval extended sequences from a set of result sequences.

E. Comparison of Execution Time

In this evaluation, we compared execution time and the number of extracted sequences with varying min_sup. We compared sequential pattern mining with item interval, our proposed algorithm, and the conventional sequential

TABLE VI.
THE NUMBER OF EXTRACTED WITH CHANGING ITEM INTERVAL CONSTRAINTS

Cond.	C_1(days)	C_2(days)	C_3(days)	C_4(days)	# of is
A	–	–	–	–	2300
B	–	10	–	–	118
C	–	100	–	–	329
D	–	1000	–	–	1726
E	–	10	–	100	118
F	–	10	–	1000	118
G	–	100	–	1000	329
H	10	–	100	–	1934
I	10	–	1000	–	649
J	100	–	1000	–	615

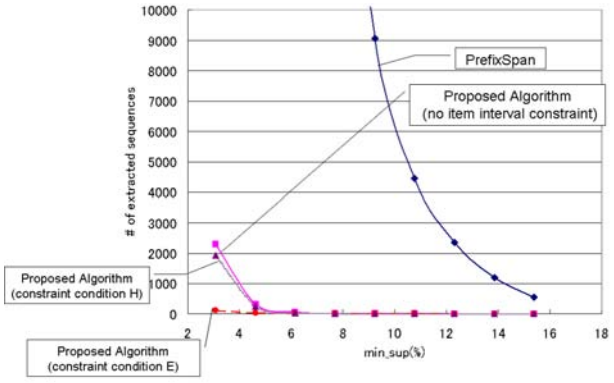

Figure 3. The Number of Patterns vs. min_sup

pattern mining algorithm, PrefixSpan. In the case of our proposed algorithm, we tested 3 situations: no item interval constraints, item interval constraint with condition E in Table VI, and item interval constraint with condition H in Table VI. Figure 3 shows the relation between the number of extracted frequent patterns and min_sup. Figure 4 shows the relation between execution time and min_sup.

As shown in Figure 3, using PrefixSpan, as the earthquake dataset is very dense, the number of extracted frequent sequences increases exponentially as min_sup decreases. On the other hand, using the proposed algorithm, the number of extracted patterns increases more slowly than with PrefixSpan because the proposed algorithm is able to distinguish the same sequences with different item intervals. Furthermore, as described above, because adapting item interval constraints reduces the number of extracted interval extended sequences, the number of extracted patterns increases much more slowly with our proposed algorithm than with PrefixSpan.

As the proposed algorithm prevents the exponential increase in number of extracted patterns, it can also prevent the huge increase in the execution time, as shown in Figure 4. These results indicate that users can analyze more sensitive patterns in a short time using our proposed method.

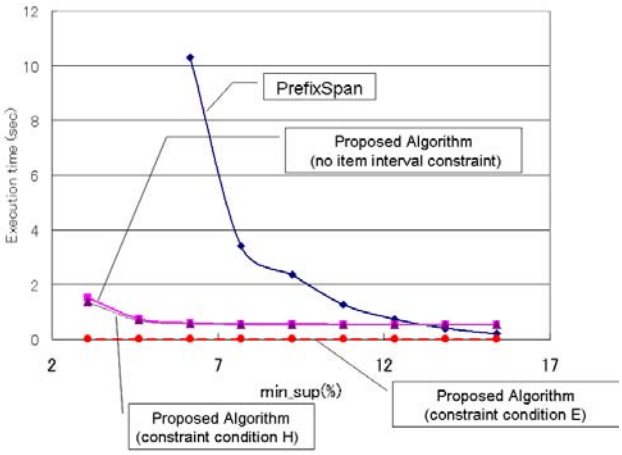

Figure 4. Execution Time vs. min_sup

VII. CONCLUSION

In this paper, we have proposed a novel algorithm for generalizing sequential pattern mining with time intervals based on our algorithm proposed previously [12]. The generalization includes three points; (a) a capability to handle two kinds of item interval measurement, item gap and time interval, (b) a capability to handle extended sequences which are defined by inserting pseudo items based on the interval itemization function, and (c) adopting four item interval constraints. Generalized sequential pattern mining is able to substitute all types of conventional sequential pattern mining algorithms with item intervals.

Evaluations using a Japanese earthquake dataset confirmed that generalized sequential pattern mining with time interval is able to extract interval extended sequences that include time interval with variable segmentation size. In addition, by adapting our types of constraint related to time intervals, it also excludes extraction of interval extended sequences with time intervals in which the user is not interested.

ACKNOWLEDGMENT

his research was funded in part by “e-Society: the Comprehensive Development Foundation Software” of MEXT (Ministry of Education, Culture, Sports, Science, and Technology) and “21-Century COE Programs: ICT Productive Academia” of MEXT.

REFERENCES

[1] Agrawal, R. and Srikant, R., “Mining Sequential Patterns,” in *Proc. of IEEE ICDE’95*, pp. 3–14, 1995.
[2] Srikant, R. and Agrawal, R., “Mining Sequential Patterns: Generalization and Performance Improvements,” in *Proc. of EDBT’96*, pp. 3–17, 1996.
[3] Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U. and Hsu, M.-C., “PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth,” in *Proc. of IEEE ICDE’01*, pp. 215–224, 2001.

[4] Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U. and Hsu, M.-C., "Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach," in *Trans. of IEEE Trans. on Knowledge and Data Engineering*, Vol. 16, No. 10, pp. 1–17, 2004.
[5] Zaki, M., "An Efficient Algorithm for Mining Frequent Sequences," *Machine Learning*, Vol. 40, pp. 31–60, 2000.
[6] Ayres, J., Gehrke, J., Yiu, T. and Flannick, J., "Sequential Pattern Mining using Bitmap Representation," in *Proc. of ACM SIGKDD'02*, pp. 429–435, 2002.
[7] Zaki, M., "Sequence Mining in Categorical Domains: Incorporating Constraints," in *Proc. of CIKM'00*, pp. 422–429, 2000.
[8] Pei, J., Han, J. and Wang, W., "Mining Sequential Pattern with Constraints in Large Databases," in *Proc. of CIKM'02*, pp. 18–25, 2002.
[9] Kitakami, H., Kanbara, T., Mori, Y., Kuroki, S. and Yamazaki, Y., "Modified PrefixSpan Method for Motif Discovery in Sequence Databases," in *Proc. of PRICAI2002*, pp. 482–491, 2002.
[10] Chen, Y.L., Chiang, M.C. and Ko, M.T. "Discovering time-interval sequential patterns in sequence databases," *Expert Syst. Appl.*, Vol. 25, No. 3, pp. 343–354, 2003.
[11] Chen, Y.L., Huang, T.C., "Discovering fuzzy time-interval sequential patterns in sequence databases," in *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 35, No. 5, pp. 959–972, 2005.
[12] Hirate, Y. and Yamana, H., "Sequential Pattern Mining with Time Intervals," in *Proc. of 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2006)*, pp. 775–779, 2006.
[13] Agrawal, R. and Srikant, R., "Fast Algorithms for Mining Association Rules," in *Proc. VLDB'94*, pp. 487–499, 1994.
[14] Pei, J. and Han, J., "Can We Push More Constraints into Frequent Pattern Mining?," in *Proc. of ACM SIGKDD'00*, pp. 350–354, 2000.
[15] K-NET Kyoshin Network, http://www.k-net.bosai.go.jp.

Yu Hirate was born in Japan in Oct. 1980. He received his B.IS. and M.E. degree from the Dept. of Computer Science, Waseda University, Tokyo, Japan, in 2004 and 2005, respectively.

Currently, he is both a Ph.D. candidate at the Dept. of Computer Science, Waseda University, and assistant professor of the Media Network Center, Waseda University.

His current research interests include sequential pattern mining application for a set of time series data. He received a Microsoft Fellowship award from Microsoft Research Asia in 2005.

Assist. Prof. Hirate is a member of ACM.

Hayato Yamana was born in Japan in 1964. He received his BS, MS, and Dr. Eng. from Waseda University, Tokyo, Japan in 1987, 1989, and 1993, respectively.

In 1993, he joined Electrotechnical Laboratory as a researcher. From 1998 to 1999, he joined the Ministry of Industry of International Trade and Industry as a technical official. From 1999 to 2000, he worked at Electrotechnical Laboratory as a senior researcher. From 2000 to 2004, he was an associate professor at Waseda University. Since April 2005, he has been professor of the Computer Science Div. of Waseda University. He worked as IEEE Computer Society Japan Chapter Chair from 2002 to 2004. His area of research is Data Mining and Computer Architecture.

Prof. Yamana is a member of IEEE and ACM.