DE GRUYTER OPEN

JAISCR, 2015, Vol. 5, No. 2, pp. 141 – 153
10.1515/jaiscr-2015-0025

A NEW APPROACH FOR DISCOVERING TOP-*K* SEQUENTIAL PATTERNS BASED ON THE VARIETY OF ITEMS

Shigeaki Sakurai[1], Minoru Nishizawa[2]

[1]*Big Data Cloud Technology Center, Toshiba Corporation Cloud & Solutions Company, 72-34, Horikawa-cho, Saiwai-ku, Kawasaki 212-8585, Japan*

[2]*Advanced IT Research and Development Center, Toshiba Solutions Company, 3-22, Katamachi, Fuchu, Tokyo 183-8512, Japan*

Abstract

This paper proposes a method that discovers various sequential patterns from sequential data. The sequential data is a set of sequences. Each sequence is a row of item sets. Many previous methods discover frequent sequential patterns from the data. However, the patterns tend to be similar to each other because they are composed of limited items. The patterns do not always correspond to the interests of analysts. Therefore, this paper tackles on the issue discovering various sequential patterns. The proposed method decides redundant sequential patterns by evaluating the variety of items and deletes them based on three kinds of delete processes. It can discover various sequential patterns within the upper bound for the number of sequential patterns given by the analysts. This paper applies the method to the synthetic sequential data which is characterized by number of items, their kind, and length of sequence. The effect of the method is verified through numerical experiments.

1 Introduction

Owing to the progress of computer environments and network environments, we can easily collect and store large amount of data which is recently called Big Data. We highly anticipate the progress of the analysis techniques for the data because new and unknown knowledge is buried in the data. Recently, the trend for the data analysis is accelerating more and more. Many companies and organizations aggressively strive for both use and application of the data. Many promising analysis techniques and examples are generated.

Big Data are composed of various types of data. Especially, it is predicted that the data generated form M2M (Machine to Machine) field and life log field explosively increases. That is, the rapid expansion of sequential data can be predicted. In the data mining research field, the analysis techniques for the sequential data has evolved from the discovery task of sequential patterns. The patterns are discovered from the sequential data where the data is composed of rows of discrete item sets. In the initial research, the discovery task aims at dealing with the data in the retail field. The item represents each sales item bought by customers. Recently, the application fields for the task are widely expanding. For example, an application field regards a word extracted from a text document as an item and a row of the words in the document as a sequence. Also, another application filed deals with discretized medical examination values collected from periodical medical examination. In addition, the other appli-

cation field deals with log data automatically sent from machines to machines. Various sequential data are generated from various application fields.

On the other hand, the initial pattern discovery methods try to discover all frequent sequential patterns by comparing their supports with the minimum support. Here, the support is an evaluation criterion for sequential patterns and the minimum support is a threshold given by analysts. However, the analysts cannot always give an appropriate minimum support because the number of frequent sequential patterns depends on the sequential data and the support cannot easily control the number. Thus, [17] proposes a method that gives the maximum number of sequential patterns as a threshold. The method discovers only frequent sequential patterns which are not included in other sequential patterns. They are called closed sequential patterns. The method can avoid the discovery of redundant sequential patterns to some extent. Also, it can avoid the explosion of the number of discovered sequential patterns.

Even if the method proposed by [17] can discover the patterns within appropriate number, it tends to discover many similar sequential patterns. That is, it discovers patterns composed of different combinations of the same items or different orders of appearance for the same items. This is because the method refers to only the frequencies of the sequential patterns. In the case of the sequential data collected from real world, the data usually includes noise. Therefore, the slight difference among sequential patterns is not important, because the difference reflects on the noise and the patterns can be interpreted as the sequential pattern with the same meaning. The analysts are not always interested in many similar sequential patterns. It is necessary for the discovery method to extract sequential patterns composed of various items.

Thus, this paper proposes a new method that aims at discovering various sequential patterns within top-k in the ranking of sequential patterns. The ranking can be decided by referring to both the minimum support and the item distribution. In the following sections, Section 2 defines the sequential data and the sequential patterns for this research. Section 3 shortly introduces the discovery method of all frequent sequential patterns [1]. Section 4 proposes upper pattern constraints and three delete processes for redundant sequential patterns. The constraints realize to discover various sequential patterns within top-k. Section 5 applies the proposed method into synthetic sequential data and verifies the effect of the proposed method. Section 6 discusses some researches related to this research. Lastly, Section 6 summarizes this paper and discuses future works.

2 Sequential data and sequential pattern

This paper deals with sequential data composed of sequentially arranged item sets. In the case of the retail field, each item represents a sales item described in a receipt. The set of the sales items in the receipt corresponds to the item set. Receipts are gathered for each customer. They are sequentially arranged according to their date. They are regarded as a sequence in the sequential data. The modeling method assumes that some same items are not included in an item set. It does not take consideration into the number of the sales. Also, it does not use additional information such as the price and the discount rate.

On the other hand, a sequential pattern is a characteristic row of item sets extracted from the sequential data. The discovery method needs evaluation criteria for characteristic rows in order to identify them. Some representative evaluation criteria are defined. In advance of the definition for the criteria, this section explains the concept of the inclusion.

Let two sequences s_1 ($= (l_{11}, l_{12}, \cdots, l_{1n_1})$) and s_2 ($= (l_{21}, l_{22}, \cdots, l_{2n_2})$) be given. If the condition as shown in Formula (1) is satisfied, s_1 includes s_2 and the relation is described by $s_2 \subseteq s_1$. Here, l_{ij} is the j-th item set composing the i-th sequence, n_k is the number of item sets included in s_k. The number is called the length.

$$\begin{aligned} &\exists y = \{y_1, y_2, \cdots, y_{n_2}\} \\ &\quad y_1 < y_2 < \cdots < y_{n_2} \\ &\quad l_{21} \subseteq l_{1y_1}, l_{22} \subseteq l_{1y_2}, \cdots, l_{2n_2} \subseteq l_{1y_{n_2}} \end{aligned} \tag{1}$$

For example, let a sequence s_A composed of three receipts for the customer "A" be given as shown in Example 1. In this example, items in-

cluded in the same item set are grouped by the symbols "{ "and "}". Different item sets are separated by the symbol "→". That is, this example represents that the customer "A" buys three sales items: "egg", "butter", and "bread" at the first day, he/she buys two sales items: "cereal" and "milk" at the second day, and he/she buys three sales items: "rice", "natto", and "egg" at the third day.

Example 1:
$\{\text{egg}, \text{butter}, \text{bread}\} \rightarrow$
$\{\text{cereal}, \text{milk}\} \rightarrow \{\text{rice}, \text{natto}, \text{egg}\}$

Next, let five sequences ($s_B \sim s_F$) for other customers "B" ∼ "F" be given as shown in Table 1. In the case of the customer "B", the first, the second, and the third item set comprising the sequence s_B are subsets of the first, the second, and the third item set composing the sequence s_A, respectively. Therefore, the sequence s_B is included in the sequence s_A. Similarly, the first item set for the sequence s_C is a subset of the first item set for the sequence s_A and the second item set for s_C is a subset of the third item set for s_A. Therefore, s_C is included in s_A. We note that the inclusion relation is satisfied even if the second item set for s_A does not correspond to any item sets in s_C.

In the case of the sequence s_D, the first item set for s_D is not a subset of all item sets for s_A. s_D is not included in s_A. Similarly the item "jam" which is a member of the first item set for the sequence s_E does not appear in s_A. s_E is not included in s_A.

On the other hand, in the case of the sequence s_F, the first item set for s_F is a subset of the third item set for s_A. The second item set for s_F is a subset of the second item set s_A. However, the order of appearance in each sequence is not equal to each other. s_F is not include in s_A.

Table 1. Sales item log

ID	Sequence
s_B	{egg, bread} → {cereal, milk} → {natto}
s_C	{butter, bread} → {rice, egg}
s_D	{egg, butter, bread, cereal} → {milk} → {egg}
s_E	{egg, jam, bread} → {cereal, milk}
s_F	{rice, natto, egg} → {cereal, milk}

Based on the inclusion relation between sequences, the discovery method calculates the frequency of each sequential patterns by counting up the number of sequences including it. The method calculates evaluation criteria of sequential patterns by referring to the frequency. Two representative criteria: support and confidence are defined by Formula (2) and Formula (3).

$$\text{supp}(s) = \frac{\text{the number of sequences including } s}{\text{the number of total sequences}} \tag{2}$$

$$\text{conf}(s|s_p) = \frac{\text{the number of sequences including } s}{\text{the number of sequences including } s_p} \tag{3}$$

Here, s is a sequential pattern and s_p is a sequential subpattern included in s. The support represents relative frequency of sequential patterns. The confidence represents conditional probability conditioned by the subpattern s_p. If we regard s_p as a premise part, the remaining part $s - s_p$ excluding s_p from s is regarded as a result part of an inference rule. We can use inference rules with high confidence in order to predict the result parts. For example, if many customers buy "bread" at one day and most of them do "milk" at next day, the confidence conf({bread}→{milk}|{bread}) is big. We can acquire such an inference rule that the premise part is {bread} and the result part is {milk}. By using the inference rule, we can predict that a customer buys "milk" at next day when he/she buys "bread".

3 The discovery method of sequential patterns

Many discovery method of sequential patterns have been proposed [1][3][14][19]. Most of them efficiently discover all sequential patterns whose supports are larger than or equal to the minimum support. The property shows that the evaluation criterion of sequential patterns monotonically decreases as the sequential patterns grow. The inference rules with high confidence are extracted from the discovered patterns. This is because the supports satisfies the Apriori property but the confidence does not satisfy it. The property is an important property for the efficient discovery of sequential patterns. This paper shortly explains the method based on the candidate [1] because the method can easily evaluate the variety of items.

Firstly, the discovery method based on the candidate calculates frequency of each item included in the sequential data. It calculates the support of each item by referring to the frequency. If the support is larger than or equal to the minimum support, the item is regarded as a frequent item. All items are evaluated their supports in order. The method can discover all frequent items. Each frequent item is called a first frequent item set. Next, the method picks up two first frequent item sets and generates a candidate item set whose number of items is 2. For example, let two first frequent item sets: {egg} and {butter} be given. The method generates a candidate as shown in Example 2.

Example 2: {egg, butter}

The method calculates the support for the candidate. If the support is larger than or equal to the minimum support, the candidate is a second frequent item set. All second frequent item sets are discovered by all the combinations of two first frequent item sets.

Next, the method combines two second frequent item sets satisfying such a condition that the top items of each second frequent item sets are equal to each other. It generates a candidate item set whose number of items is 3. The candidate is composed of the top item and two remaining items. For example, let two second frequent item sets: {egg, butter} and {egg, bread} be given. The top items of each second frequent item set are "egg" and are equal to each other. The method generates a candidate as shown in Example 3.

Example 3: {egg, butter, bread}

The method calculates the support for the candidate. If the support is larger than or equal to the minimum support, the candidate is a third frequent item set. All third frequent item sets are discovered by all the combinations of two second frequent item sets satisfying the condition. The Apriori property guarantees that the generation based on the condition leads to all frequent third item sets.

Generally, the method combines two i-th frequent item sets satisfying such a condition that the remaining items except the last item are equal to each other for the i-th frequent item sets. It generates a candidate item set whose number of items is i+1. The candidate is composed of i-1 common items and two last items. Figure 1 shows the outline of the generation. In this figure, each circle shows an item and its texture does a kind for the item. The method calculates the support for the candidate. If the support is larger than or equal to the minimum support, the candidate is a (i+1)-th frequent item set. All (i+1)-th frequent item sets are discovered by evaluating all the combinations of two i-th frequent item sets satisfying the condition. Each discovered frequent item set is one of first frequent sequential patterns.

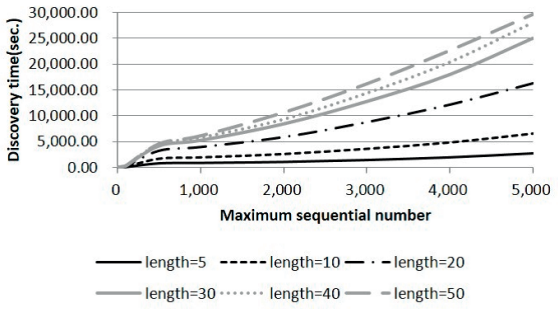


Figure 1. Generation of a candidate item set

After the discovery of all first frequent sequential patterns, the method expands patterns at the sequential direction. That is, the method picks up two first sequential patterns. It arranges two patterns in order and generates two candidates whose length is 2. For example, let two first frequent sequential patterns: {egg} and {cereal, milk} be given. The method can generate two candidates as shown in Example 4 and Example 5 by changing the order of appearance.

Example 4: {egg} → {cereal, milk}

Example 5: {cereal, milk} → {egg}

Each candidate is calculated its support. The method judges whether each candidate is frequent or not, respectively. Frequent candidates are second frequent sequential patterns.

The method combines two second frequent sequential patterns satisfying such a condition that the top item sets of each second frequent sequential pattern are equal to each other. It generates two can-

didates whose length is 3. Each candidate is composed of the top item set and two remaining item sets. The method can generate two candidates by changing the order of appearance for the remaining item sets. For example, let two second frequent sequential patterns $\{egg\} \rightarrow \{cereal, milk\}$ and $\{egg\} \rightarrow \{natto\}$ be given. The method can generate two candidates as shown in Example 6 and Example 7.

Example 6: $\{egg\} \rightarrow \{cereal, milk\} \rightarrow \{natto\}$

Example 7: $\{egg\} \rightarrow \{natto\} \rightarrow \{cereal, milk\}$

Generally, the method generates two candidates whose length is j+1 by combining two j-th frequent sequential patterns. The patterns satisfy such a condition that the subpatterns except the last item set are equal to each other. Each candidate is composed of the common subpattern and two last item sets. Figure 2 shows the outline of the generation. In this figure, each item set in a sequence is separated by arrow lines. The method can generate another candidate by changing the order of appearance for the last item sets. We note that the method can pick up two same j-th frequent sequential patterns. In this case, a candidate is generated because the change of the order leads to the same candidate. The frequent candidates are (j+1)-th frequent sequential patterns. The generation is repeated until all frequent sequential patterns are discovered.

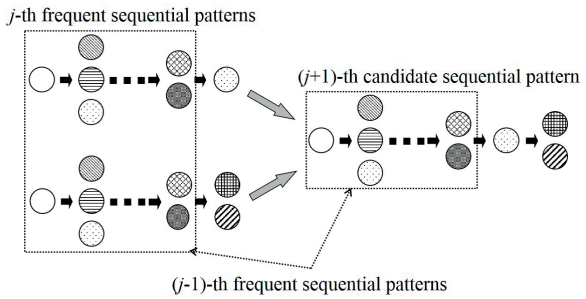


Figure 2. Generation of a candidate sequential pattern

4 Upper pattern constraint

This section proposes a upper pattern constraint and three delete processes. The upper pattern constraint is a constraint aiming at the discovery of sequential patterns within top-k. The value of k is the upper bound for the number of discovered sequential patterns and given by the analysts in advance of the analysis. It is called the maximum sequential number in the following. The upper pattern constraint selects sequential patterns based on a upper-lower relation between any two sequential patterns. The relation is decided by referring to both the frequency of sequential pattern and the item distribution composing it. Lower sequential patterns and redundant sequential patterns are deleted by using three delete processes. The processes can reduce the number of sequential patterns within the maximum sequential number. The concrete method is discussed hereafter.

The discovery method based on the upper pattern constraint refers to the minimum support. However, the role of the minimum support is different from the one of previous discovery methods. In the former role, the minimum support is an evaluation criterion excluding sequential patterns whose frequencies are too small and which are accidentally extracted by the influence of the noise. It is the minimum requirement which avoids discovering distinct uncharacteristic sequential patterns. We can set it to a smaller value than the latter role. This is because the previous discovery methods based on the latter role require high minimum support in order to avoid the explosion of number of discovered sequential patterns. In addition, in the latter role, the slight change of the minimum support can lead to the huge change of the number of discovered sequential patterns. The changes depend on the sequential data. It is difficult to decide appropriate minimum support. Usually, the decision is selected through trial and error. On the other hand, in the former role, the minimum support does not give a big impact to the number. This is because the number is controlled by the maximum sequential number. We can easily set the minimum support in the case of the former role.

The discovery method based on the upper pattern constraint uses the framework for the discovery method based on the candidate. It evaluates whether the number of stored sequential patterns is over the maximum sequential number or not, whenever a candidate sequential pattern is frequent. In the case that the number is over the maximum se-

quential number, the discovery method selects a sequential pattern keeping the variety of items even if the pattern is excluded from the stored sequential patterns. The selected pattern is deleted. We can anticipate that the variety of items is preserved in remaining stored sequential patterns. That is, non-similar sequential patterns are remained in it.

We can design various evaluation criteria in order to evaluate the variety of items. This paper focuses on the variance of item distribution in the stored sequential patterns. That is, the evaluation criterion is defined by Formula (4). In the following, it is called item variance ratio.

$$varRatio(s) = 1 - \frac{var_{it}(S_s)}{var_{it}(S_o)} \tag{4}$$

In this formula, s is a target sequential pattern, S_o is the whole set of stored sequential patterns, and S_s is such a remaining set that s is excluded from S_o. $var_{it}(S)$ is a function calculating a variance of items for a given sequential pattern set S. The variance of items is calculated by Formula (5).

$$var_{it}(S) = \frac{1}{N_{it}} \sum_{i \in s_p \in S} (freq_{s_p}(i) - \frac{\sum_{i \in s_p \in S} freq_{s_p}(i)}{N_{it}})^2 \tag{5}$$

In this formula, S is a target set of sequential patterns, s_p is one of sequential patterns included in S, and i is one of items included in s_p. N_{it} is kinds for items included in the sequential data. $freq_{s_p}(i)$ is a function calculating a number of an item i included in a given sequence s_p.

The item variance ratio shows that the variance of items is to be small when a sequential pattern with large item variance ratio is deleted from the stored sequential patterns. Therefore, the delete can lead to a uniform item distribution and can preserve various items. We can anticipate that various sequential patterns is preserved.

Next, this section discusses three delete processes for the discovery method based on the upper pattern constraint. First one is the delete process of same length patterns, second one is the delete process of included patterns, and third one is the delete process of different length patterns. Figure 3 shows the outline of these processes. In this figure, the number of patterns in dashed rectangles is controlled within the maximum sequential number. Each delete process is incrementally performed. After the delete process of different length patterns, the growth of sequential patterns is tried. In the following, this paper explains these processes in detail.

The processes can be buried in the discovery method based on the candidates. The discovery method based on the upper pattern constraint requires j-th frequent sequential patterns in order to generate (j+1)-th candidate sequential patterns. It is indispensable for the discovery method to preserve j-th frequent sequential patterns during the generation. After the generation, the discovery method can delete j-th frequent sequential patterns. Therefore, the first delete process is performed during the generation of (j+1)-th candidate sequential patterns. If the number of (j+1)-th frequent sequential patterns is over the maximum sequential number, the first delete process selects the lowest (j+1)-th frequent sequential pattern. It excludes the lowest one from stored (j+1)-th frequent sequential patterns. In Figure 3, the sequential pattern in the left bottom of the figure is a (j+1)-th frequent sequential pattern which is discovered in (k+1)-th order. The second sequential pattern in the left dashed rectangle is the lowest (j+1)-th frequent sequential pattern. It is excluded from the rectangle and the pattern in (k+1)-th order is inserted in the rectangle when the item variance ratio for the second sequential pattern is lower than the one for the discovered sequential pattern. If all (j+1)-th candidate sequential patterns are evaluated, the stored (j+1)-th frequent sequential patterns are merged with the stored first $\sim$ j-th frequent sequential patterns. The discovery method may temporally preserve sequential patterns whose number is two times as large as the maximum sequential number because the number of first $\sim$ j-th frequent sequential patterns can be equal to the maximum sequential number.

Next, the second delete process evaluates the inclusion relation between (j+1)-th frequent sequential patterns and remaining stored frequent sequential patterns. All included patterns are excluded from the stored frequent sequential patterns. In the center rectangle of Figure 3, two first $\sim$ j-th frequent sequential patterns are excluded by their corresponding (j+1)-th frequent sequential patterns.

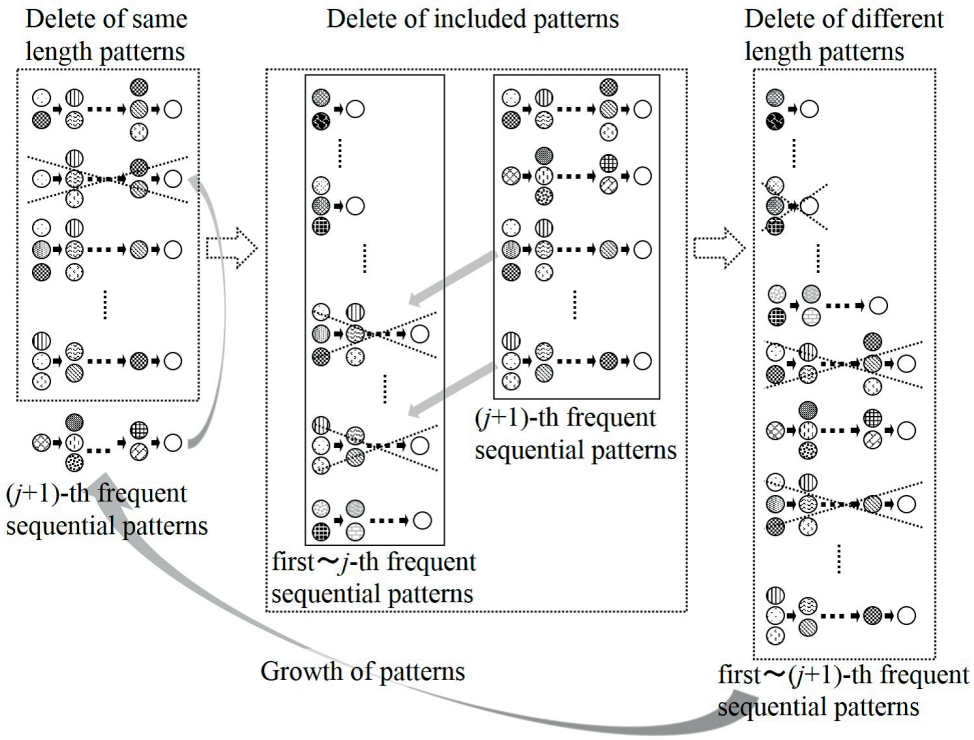


Figure 3. Three delete processes in the discovery method based on upper pattern constraints

Lastly, the third delete process is repeated until the number of remaining stored frequent sequential patterns arrives at the maximum sequential number. That is, if the number is over the maximum sequential number, the third delete process picks up the lowest stored frequent sequential patterns from the remaining ones and excludes it. The patterns in the right rectangle of Figure 3 are excluded according to their orders. Here, we note that the second delete process may lead to the smaller number than the maximum sequential number. In this case, the third delete process cannot be performed. The number of remaining stored frequent sequential pattern may be smaller than the maximum sequential number. Therefore, the maximum sequential number is the upper bound for the number of discovered frequent sequential patterns.

5 Numerical experiment

5.1 Experimental data

This paper uses synthetic sequential data generated by a data generator. The generator deals with tabular structured sequential data. Each item is composed of an attribute and its value. The generator can generate the sequential data based on given conditions and random numbers. That is, it generates sequences whose number is equal to given number of sequences. Length of each sequence corresponds to given length. Number of items composing an item set in the sequence is controlled within given maximum number of items. The number is decided by a uniform random number. Each item is composed of an attribute and an attribute value, and their selection is based on uniform random numbers, respectively. Table 2 shows the conditions for this experiment. Only the length is changed from 5 to 50. The remaining conditions are fixed. We can generate 6 sequential data sets corresponding to the length.

Table 2. Conditions of data generation

Condition	Value
Attribute	30
Attribute value	10
Item	300(=30×10)
Max. number of items	50
Length	5, 10, 20, 30, 40, 50
Number of sequences	10,000

Table 3 shows the memory size of the generated sequential data, average number of items in each sequence, and average kinds for items in each sequence for each data set. According to the increase of the length, the size and the average number of items linearly increase. On the other hand, each sequence includes most of kinds for items in the case that the length is 50.

Table 3. Features of sequential data

Length	Size(KB)	Item	Kind
5	5,340	127.7	107.7
10	10,522	254.9	176.6
20	21,192	510.7	249.4
30	31,788	766.0	279.2
40	42,359	1,020.6	291.4
50	52,911	1,274.8	296.4

5.2 Experimental method

This paper compares the discovery method based on the upper pattern constraint with the discovery method without using the constraint [1]. In the following, the former one is called the proposed method and the latter one is called the previous method. In the case of the proposed method, the maximum sequential numbers are set to 10, 50, 100, 500, 1,000, 2,000, 3,000, 4,000 and 5,000. Also, the minimum support is set to 0.005. In the case of the previous method, the minimum support is set to 0.0655. It is decided due to the preliminary experiment. It can discover second frequent sequential patterns from each data set. The experiment uses the synthetic sequential data based on the random numbers. Sequential patterns whose numbers of items are equal to each other can be anticipated to have similar supports. Large number of sequential patterns can be discovered if the minimum support is small. Therefore, the minimum support in the case of the previous method is set to a higher value than the one of the proposed method. In two methods, the maximum length is 5 and the maximum number of items is 1. They try to discover first $\sim$ 5th frequent sequential patterns whose number of items in each item set is 1. Additionally, they are measured their discovery time. The experiment measures the discovery time by using the "time" command offered by OS. The command can measure the time with second unit. We describe a sample program for the discovery methods with C language. The program is compiled by gcc. It is run in the computer environment as shown in Table 4.

Table 4. Computer environment

OS	CentOS
CPU	1GHz $\times$ 6 core
RAM	24GB

5.3 Experimental result

Figure 4 shows discovery time of the proposed method in the case that the maximum sequential numbers are changed. In this figure, x axis shows the maximum sequential number and y axis shows the discovery time with second unit. 6 lines in this figure represent results corresponding to each sequential data set. "*" in "length=*" represents length of sequences in the data set.

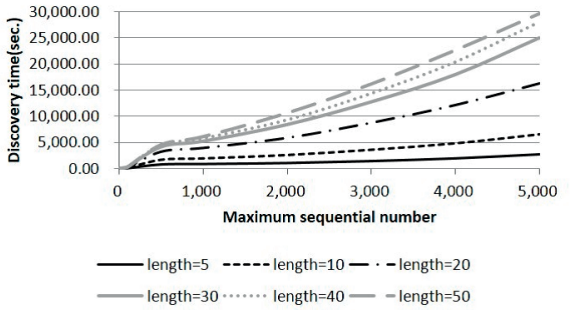


Figure 4. Discovery time

Table 5 shows number of discovered frequent sequential patterns in the case of the previous method. In this table, the column for "Length" shows the length of sequential data sets. Each column for "First", "Second", and "Third" represents the number of sequential patterns whose length is 1, 2, and 3, respectively. "-" shows that the previous method fails to discover all third sequential patterns. This table shows that all items (first sequential patterns) are discovered for all data sets. Also, it shows that all second sequential patterns are discovered as frequent ones for data sets except "length=5".

Table 5. Number of patterns in the case of the previous method

Length	First	Second	Third
5	300	2,436	0
10	300	90,000	0
20	300	90,000	-
30	300	90,000	-
40	300	90,000	-
50	300	90,000	-

Figure 5 shows ratios of the discovery time of the proposed method for the one of the previous method. In this figure, x axis shows the maximum sequential number and y axis shows the ratios. This figure includes 5 lines representing each sequential data set. In this figure, "length=5" is omitted because the previous method discovers second frequent sequential patterns whose number is small due to the big minimum support. Therefore, it does not sufficiently investigate candidate patterns whose length is 3 or more. The discovery time of the previous method is underestimated. On the other hand, in the case that the length is 20, 30, 40, and 50, the previous method cannot discover all third frequent sequential patterns because large amount of sequential patterns are discovered. Therefore, the discovery time in their cases is estimated by referring to ratio of investigated patterns. The ratio is from 0.303 to 0.378.

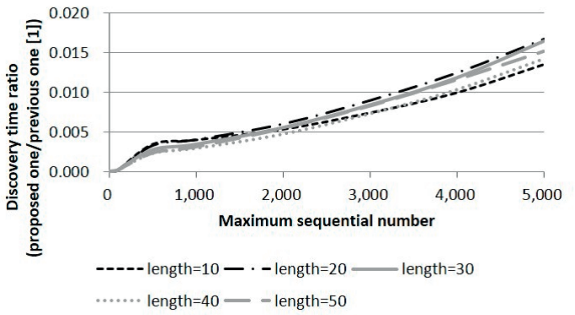


Figure 5. Relative ratio of discovery time

Figure 6 shows component ratios of frequent sequential patterns whose length is equal to each other. 6 sub-figures in this figure correspond to results for each data set. Each sub-figure has 9 parts such as 10, 50, 100, 500, 1,000, 2,000, 3,000, 4,000, and 5,000. Each number corresponds to the maximum sequential numbers. Also, each part has 5 bars even if some bars are missing in some parts. These bars correspond to the component ratios in the case that the length of sequential patterns is 1, 2, 3, 4, and 5. In these sub-graphs, y axis shows the component ratio.

Figure 7 shows kinds for items included in discovered sequential patterns. In this figure, x axis shows the maximum sequential number and y axis shows kinds for items. This figure has 6 lines. Each line shows results corresponding to each data set.

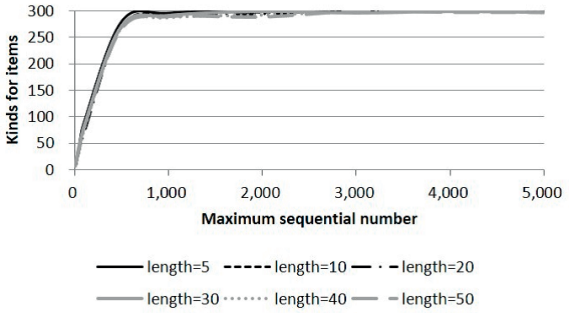


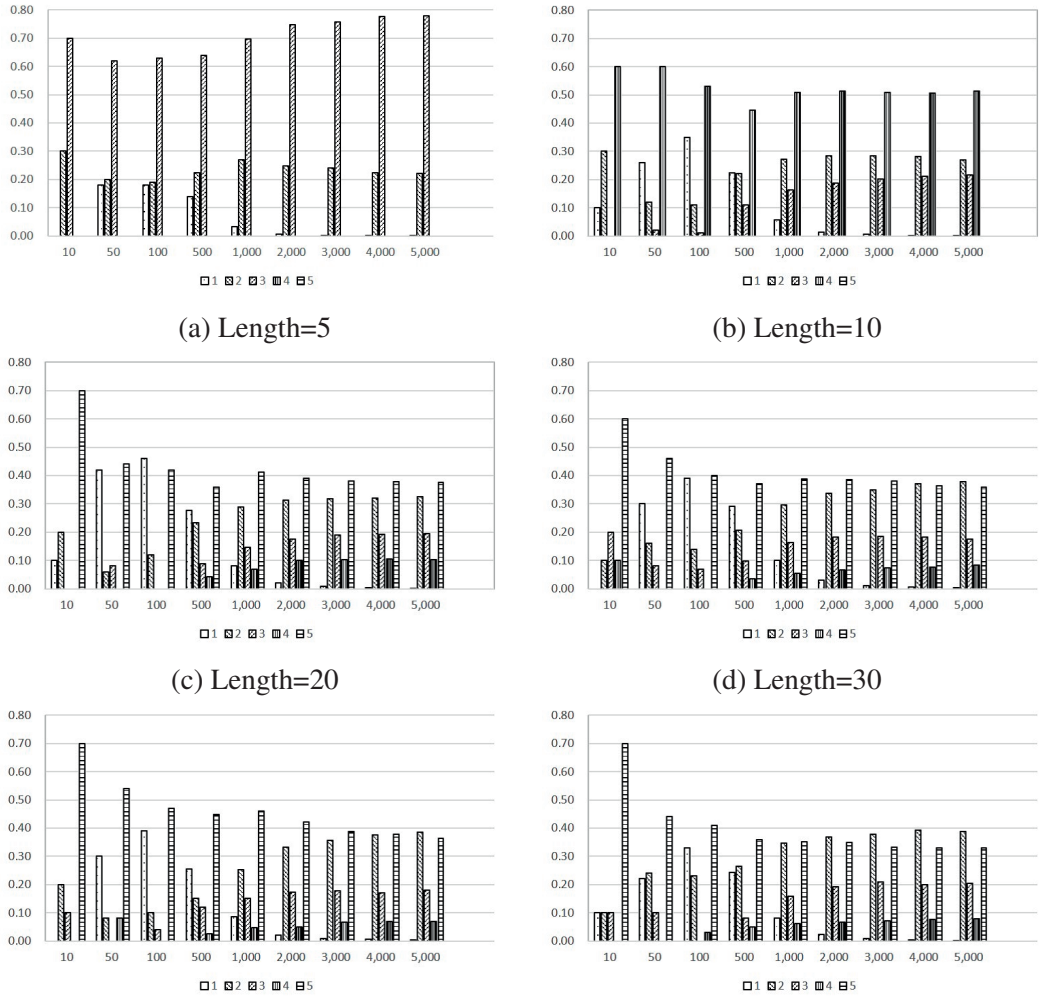
Figure 7. Kinds for items

5.4 Discussion

This section discusses the effect of the proposed method with three viewpoints.

Possibility of discovery for characteristic patterns: Table 5 shows that the previous method cannot discover sequential patterns whose length is longer than or equal to 3. In the case that the length for sequential data sets is 5 and 10, the supports of the patterns are smaller than the minimum support and the previous method cannot discover the patterns. In the case that the length is longer than or equal to 20, the numbers of the patterns are very huge and the previous method fails to store them in the memory. The previous method cannot sufficiently identify patterns, even if each data set tends to include many similar patterns. On the other hand, the proposed method succeeds to discover third $\sim$ 5th sequential patterns as shown in Figure 6. We can anticipate that characteristic sequential patterns are included in the discovered sequential patterns. We can confirm the possibility of discovery for characteristic patterns based on the proposed method.

Discovery time: Figure 4 shows that the discovery time increases in case that the size of sequential data sets increases or the number of discovered sequential patterns increases. However, the increase of the discovery time is not exponential and



(a) Length=5 (b) Length=10

(c) Length=20 (d) Length=30

(e) Length=40 (f) Length=50

Figure 6. Component ratios of frequent sequential patterns

is controlled within an range of linear increase. In the case that the sequential data set for "length=50" is analyzed and the maximum sequential number is set to 5,000, the proposed method can discover sequential patterns about 8.2 hours. If the analysis starts at the end of the business hour, we can get the result by the start of the next one. The discovery time is sufficiently practical.

On the other hand, Figure 5 shows that the proposed method can discover sequential patterns more speedily than the previous one can. In the case that the maximum sequential number is set to 5,000, the ratios of the proposed method for the previous one are from 0.0135 to 0.0167. In addition, the proposed method investigates 4th and 5th seqeutnail patterns. We can confirm that the discovery time is dramatically decreased.

In this experiment, the proposed method is serially performed. However, it can be in parallel processed by dividing a sequential data set or by dividing candidate sequential patterns. The parallel process can lead to decrease the discovery time.

Variety of discovered patterns: Figure 7 shows the increase of kinds for items according to the one of the maximum sequential numbers. In the case that the maximum sequential number is larger than or equal to 1,000, sequential patterns discovered by the proposed method include most of kinds for items. We can confirm that the proposed method preserves the variety of the sequential patterns.

Also, Figure 6 shows that the component ratios are close to each other. Especially, the trend is clearer in the case that the maximum sequential numbers are larger. The discovery of more sequential patterns can lessen the imbalance of items. We can anticipate to discover sequential patterns without the imbalance by using larger maximum sequential numbers.

According to these discussions, we believe that the proposed method can discover various sequential patterns without explosion of the number for sequential patterns.

6 Related works

This section discusses related works. This paper deals with discrete sequential data composed of items or item sets. However, it is necessary to deal with numerical sequential data. This is because there are large amount of numerical sequential data such as stock price sequences, electricity usage sequences, and temperature sequences for indoor/outdoor in real world applications. A discretization method of the numerical data is indispensable in order to apply the proposed method. SAX(Symbolic Aggregate approXimation) [11] is such an initial method that numerical sequences transform into symbol sequences. We can use the symbol sequences as the input of the proposed method. SAX divides numerical sequences into subsequences with the same length. It calculates an average value for each subsequence. Each average value is transformed into a symbol. Then, SAX assumes that the distribution of normalized average values corresponds to a Gaussian distribution. SAX decides such thresholds in the distribution that each symbol appears with the same frequency. An interval decided by the thresholds corresponds to a symbol. SAX are revised with many viewpoints. [12] proposes a discretization method that uses the average value, the maximum value, and the minimum value in each subsequence. Also, [13] and [20] propose discretization methods that use the average value and the direction of change in each subsequence. These revised method aim at more accurate discretization by the use of additional information. On the other hand, the concept of sequences or discovered patterns has been expanded in many researches. In this research direction, we firstly note to the usage of the time information. [7] and [18] deal with the time between items in sequences. [7] discovers sequential patterns that have the time between items and [18] does sequential patterns that have the time interval between them. Also, [8] deals with items with the time interval. It discovers relations between items that are defined by Allen's temporal interval logic [2]. The logic can represent the relations such as "before", "meet", "overlap", and so on. [10] uses an expression of calendar described by a fuzzy set as a constraint. It discovers asynchronous periodical association rules in the period satisfying the constraint. [9] generates a sequence described by fuzzy sets from a numerical sequence. It focuses on items in sequences breaking out at the different time. It discovers association rules representing the relation between the items. Next, we note researches for the other direction of

the expansion. [6] deals with numerical sequences. It regards the change between neighbor points in a sequence as a trend. Each sequence is transformed into trend sequences. Sequential patterns composed of trends are discovered. [5] deals with incomplete sequences whose parts of items are missing. It estimates such probability that a missing item breaks out. In the discovery of frequent sequential patterns, the probability is referred. [4] deals with literature in the biomedical field. Words in the literature are assigned labels such as part of speech, gene name, and relations among words. Fuzzy sequential patterns are discovered from sequences composed of words with labels. Each pattern represents interaction between genes. Even if some related works are discussed in this section, we can find many other related works. We can believe that the research area is attractive and that additional researches should be continued.

Conclusion

This paper proposed a new method discovering various sequential patterns within top-k. The method can avoid the discovery of many similar sequential patterns. Also, it can avoid the explosion of number for sequential patterns due to the set of inappropriate parameters. This paper verified the effect of the method through numerical experiments based on synthetic tabular structured data.

In future works, we are planning to apply the software incorporating the proposed method to real application data such machine log data, medical examination data, and nursing service data. We will aim at discovering effective sequential patterns from the data. On the other hand, we are planning to expand the analysis techniques for the sequential patterns. For example, the discretization technique for numerical sequential data, the generation technique of the time relation among items, the division technique for very long sequences, and the real-time update technique of sequential patterns are promising analysis techniques. In addition, we will tackle on the research of the techniques dealing with more complicated data such as the combination of text data, voice data, and image data. We can believe that the development of these techniques leads to the realization of smarter society.

References

[1] R. Agrawal, R. Srikant, Mining Sequential Patterns, Proc. of the 11th International Conference on Data Engineering, 1995, pp. 3-14.

[2] J. F. Allen, Maintaining Knowledge about Temporal Intervals, Communications of the ACM, vol. 26, no. 11, 1983, pp. 832-843.

[3] J. Ayres, J. E. Gehrke, T. Yiu, J. Flannick, Sequential Pattern Mining using Bitmaps, Proc. of the 8th International Conference on Knowledge Discovery and Data Mining, 2002, pp. 429-435.

[4] J.-H. Chiang, Z.-X. Yin, C.-Y. Chen, Discovering Gene-gene Relations from Fuzzy Sequential Sentence Patterns in Biomedical Literature, Proc. of the 13th IEEE International Conference on Fuzzy Systems, vol. 2, 2004, pp. 1165-1168.

[5] C. Fiot, A. Laurent, M. Teisseire, Approximate Sequential Patterns for Incomplete Sequence Database Mining, Proc. of the 16th IEEE International Conference on Fuzzy Systems, 2007, pp. 1-6.

[6] C. Fiot, F. Masseglia, A. Laurent, M. Teisseire, TED and EVA: Expressing Temporal Tendencies among Quantitative Variables using Fuzzy Sequential Patterns, Proc. of the 17th IEEE International Conference on Fuzzy Systems, 2008, pp. 1861-1868.

[7] F. Giannotti, M. Nanni, D. Pedreschi, Efficient Mining of Temporally Annotated Sequences, Proc. of the 2006 SIAM International Conference on Data Mining, 2006, pp. 348-359.

[8] F. Höppner, Discovery of Temporal Patterns - Learning Rules about the Qualitative Behaviour of Time Series, Proc. of the 5th European Conference on Principles of Data Mining and Knowledge Discovery, 2001, pp. 192-203.

[9] Y.-P. Huang, L.-J. Kao, A Novel Approach to Mining Inter-transaction Fuzzy Association Rules from Stock Price Variation Data, Proc. of the 14th IEEE International Conference on Fuzzy Systems, 2005, pp. 791-796.

[10] J.-Y. Jiang, W.-J. Lee, S.-J. Lee, Mining Calendar-based Asynchronous Periodical Association Rules with Fuzzy Calendar Constraints, Proc. of the 14th IEEE International Conference on Fuzzy Systems, 2005, pp. 773-778.

[11] J. Lin, E. Keogh, S. Lonarrdi, B. Chiu, A Symbolic Representation of Time Series, with Implications for Streaming Algorithms, Proc. of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2003, pp. 2-11.

[12] B. Lkhagava, Y. Suzuki, K. Kawagoe, Extended SAX: Extension of Symbolic Aggregate Approximation for Financial Time Series Data Representation, Proc. of the Data Engineering Workshop 2006, 2006, 4A0-8.

[13] S. Malinowski, T. Guyet, R. Quiniou, R. Tavenard, 1d-SAX : A Novel Symbolic Representation for Time Series, Proc . of the 12th International Symposium on Intelligent Data Analysis, 2013, pp. 273-284.

[14] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, M. Hsu, PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth, Proc. of the 17th International Conference on Data Engineering, 2001, pp. 215-224.

[15] S. Sakurai, K. Ueno, R. Orihara, Discovery of Time Series Event Patterns based on Time Constraints from Textual Data, International Journal of Computational Intelligence, vol. 4, no. 2, 2008, pp. 144-151.

[16] R. Srikant, R. Agrawal, Mining Sequential Patterns: Generalizations and Performance Improvements, Proc. of the 5th International Conference on Extending Database Technology, 1996, pp. 3-17.

[17] P. Tzvetkov, X. Yan, J. Han, TSP: Mining Top-*k* Closed Sequential Patterns, Knowledge and Information Systems, vol. 7, issue 4, 2005, pp. 438-457.

[18] A. Vautier, M.-O. Cordier, R. Quiniou, An Inductive Database for Mining Temporal Patterns in Event Sequences, Proc. of the 2005 ECML/PKDD Workshop on Mining Spatial and Temporal Data, 2005, pp. 1640-1641.

[19] M. J. Zaki, SPADE: An Efficient Algorithm for Mining Frequent Sequences, Machine Learning, vol. 42, no. 1, 2001, pp. 31-60.

[20] W. Zalewski, F. Silva, H. D. Lee, A. G. Maletzke, F. C. Wu, Time Series Discretization based on the Approximation of the Local Slope Information, Proc. of the 13th Ibero-American Conference on AI, 2012, pp. 91-100.

Shigeaki Sakurai received an MS degree in mathematics and a Ph.D. degree in industrial administration from Tokyo University of Science, Japan, in 1991 and 2001, respectively. He has been the associate professor at the Tokyo Institute of Technology from 2009 to 2013. He is a senior specialist research at the IoT Technology Center, Toshiba Corporation. His research interests include data mining, soft computing, and web technology.

Minoru Nishizawa received an MS degree in engineering from Saga University, Japan, in 2000. He is a specialist research at the IoT Technology Center, Toshiba Corporation. His research interests include data mining, soft computing, and computer security.