Detection of Linkage Patterns Repeating across Multiple Sequential Data

Takahiro Miura
Department of Information and Electronic Engineering, Muroran Institute of Technology
27-1 Mizumoto-cho, Muroran-shi, Hokkaido
Japan, 050-8585

Yoshifumi Okada
College of Information and Systems, Muroran Institute of Technology
27-1 Mizumoto-cho, Muroran-shi, Hokkaido
Japan, 050-8585

ABSTRACT
Sequential data mining is a technology for acquiring useful information and patterns from large quantities of sequential data. Research into industrial and commercial applications of sequential data mining is flourishing. The aim of this study is to propose a new method for detecting groups of patterns that appear in a linked manner across multiple sequential data and repeat along a time axis. Such a set of pattern groups is called a "linkage pattern." Linkage pattern is detected by using interval graphs that are generated from frequent patterns in multiple sequential data. The difference between this method and previous methods is that it does not assume similarity or correlation between patterns in different sequential data. If a pattern that frequently occurs in individual sequential data does not show similarity with patterns in different sequential data, these patterns will be detected as a linkage pattern as long as they are linked along a time axis. In this paper, this method is applied to artificial data with embedded linkage patterns and the detection accuracy is evaluated using three indexes (precision, recall, and F-measure). As a result, it is shown that embedded linkage patterns can be suitably detected and that detection accuracy increases as the window width for frequent pattern detection decreases.

General Terms
Data mining, Sequential Data

Keywords
Sequential pattern mining, Linkage pattern, Interval graph

1. INTRODUCTION
With the increase in computer performance and the ubiquitous spread of computer technology, the storage of time-sequential data, such as Web log data and sensory data, has rapidly increased. Sequential data mining is a technology for acquiring useful information and patterns from large quantities of sequential data. Research in industrial and commercial applications of sequential data mining is flourishing [1, 2].

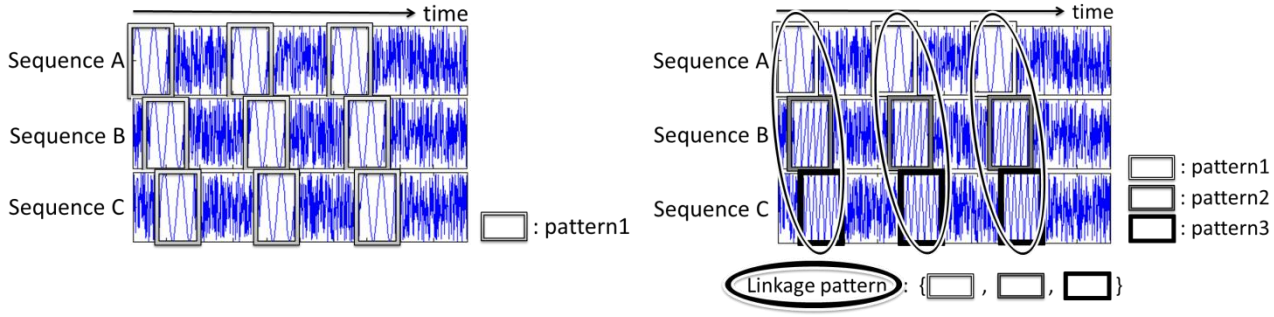
The majority of research related to sequential data mining has focused on the detection of patterns occurring in single sequence data [3, 4]. Recently, several methods aimed to propose multiple sequential data. These methods aimed to detect co-occurring or highly related patterns within a specific time interval from different sequential data [5 - 11].

This paper describes a new method to detect groups of linked patterns that exist across multiple sequential data and repeat along a time axis. This type of pattern group is called a "linkage pattern." The difference between this method and previous methods is that it does not suppose similarity or correlation between patterns in different sequential data. In other words, even if a pattern that frequently occurs in individual sequential data does not show similarity with frequent patterns in different sequential data, these patterns will be detected as a linkage pattern as long as they are linked along a time axis. Through the detection of linkage patterns, it will be possible to detect causal relationships across different sequential data, such as earthquake observation data and various other vital data from multiple sources. In this paper, we will investigate the effectiveness of this method by conducting performance evaluation experiments using artificially generated data.

This paper is structured as follows. In Section 2, the basic concepts of the method are stated. Section 3 explains the procedure. In Sections 4 and 5, the performance evaluation experiments are conducted and the results of these experiments are presented, respectively. Section 6 provides a summary of this paper.

2. BASIC CONCEPTS
Previous methods for multiple sequential data detect similar or correlated subsequences that frequently occur among different sequential data, as shown in Figure 1(a). The goal is to enumeration all subsequences occurring above a specified threshold (known as minimum support).

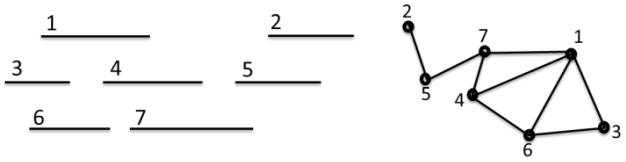
This study, on the other hand, aims to detect a set of pattern groups (a linkage pattern) that occurs in the same period of time across multiple sequential data, as shown in Figure 1(b). A linkage pattern is allowed to include different patterns in different sequential data as long as they are linked along a time axis. This study uses the concept of an interval graph, which is a subgraph of a chordal graph [12 - 14], to detect these types of linkage patterns. An interval graph is a graph that represents intervals, as shown in Figure 2(a). In an interval graph, each interval is a node and overlapping intervals are represented as the edges between nodes, as shown in Figure 2(b). If we regard each frequent pattern as nodes and those overlaps along time axis as edges, it can be treated as an interval graph mining problem. With interval graph mining, we can detect connected graphs in which a path exists between any two nodes, as shown in Figure 2(b). Hence, in addition to mutually overlapping frequent patterns, it is



(a) Sequential pattern detected by previous methods

(b) Linkage pattern detected by the proposed method

Fig 1: Previous methods and the proposed method



(a) Interval representation **(b) Interval graph**

Fig 2: Interval representation and Interval graph

to successively detect linked patterns along the time axis that are not mutually overlapping.

3. METHOD

Figure 3 shows the procedure for this method. In this study, we provide a method for detecting one kind of linkage pattern that occurs frequently across multiple sequential data. The linkage pattern is detected according to the following steps.

3.1 Preprocessing

First, normalization and discretization are performed on each sequential data. Normalization is initially performed on data x_t at time point t, using the following formula, so that the minimum value is 0 and the maximum value is 1:

$$normalize(x_t) = \frac{x_t - \min}{\max - \min}.$$

Here *max* and *min* represent the maximum value and minimum value, respectively, within the sequential data.

Next, the normalized data range (0–1) is divided into D stages and a discretized value from 0 to D-1 is assigned to each data.

3.2 Frequent pattern detection and their labeling

Figure 3(a) provides an outline of the detection of frequent patterns in each sequential data and their labeling. First, frequent patterns are detected for each sequence data. In this study, frequent patterns are detected using an algorithm proposed by *Mannila* et al. [4]. This algorithm uses the maximum window width w and minimum number of occurrences θ of the frequent pattern as input parameters. Labels are then applied to each of the frequent patterns. At this time, frequent patterns with a length of $w/2$ or less are excluded without being labeled. When multiple frequent patterns occur at the same time within the same sequential data, the frequent pattern with the maximum length is selected and a label is applied.

3.3 Linkage pattern detection

With the above process, an interval representation for the frequent patterns detected from each sequential data is generated, as shown in Figure 3(b). In the interval representation, interval graphs are extracted by finding intervals that overlap along the time axes of the sequential data. Finally, the interval graph with the highest number of occurrences is output as a linkage pattern.

4. EVALUATION EXPERIMENT

In this study, artificially generated sequential data is used to evaluate the accuracy of detecting linkage patterns. The artificial data was composed of three sets of sequential data in which each sequential data includes one kind of linkage pattern embedded into uniformly distributed random sequential data. The occurrence frequency of the linkage pattern was set to 10. Hereinafter, each of the 10 patterns is called an embedded linkage pattern.

In this experiment, three artificial data samples (hereafter Sample 1, Sample 2, and Sample 3) were created. Sample1 was artificial data in which frequent patterns in the linkage pattern are identical length (11 data points) and appear at the same start time. Sample2 was artificial data in which frequent patterns in the linkage pattern are identical length (11 data points) and appear at different start times. Sample3 was artificial data in which frequent patterns in the linkage pattern are different lengths (5, 11, and 15 data points) and appear at the same start time. Figure 4 shows the artificial data for Sample 1.

The maximum width w was set to 5, 10, and 15, and frequent pattern frequency thresholds θ of 10, 30, and 50 were used for each maximum window width for Samples 1–3, respectively. Performance was evaluated using the detection accuracy of embedded linkage patterns across the three sequential data. The precision, recall, and F-measure, which considers both indexes, were used as evaluation indexes. These were calculated according to the following formula:

$$Precision = CDP / DDP,$$

$$Recall = CDP / EDP,$$

$$F\text{-}measure = 2 * Precision * Recall / (Precision + Recall).$$

Here *CDP* is the number of data points in the correctly detected areas of the embedded linkage patterns, *DDP* is the number of data points in the areas of the embedded linkage patterns detected by this method, and *EDP* is the number of data points in the embedded linkage patterns.

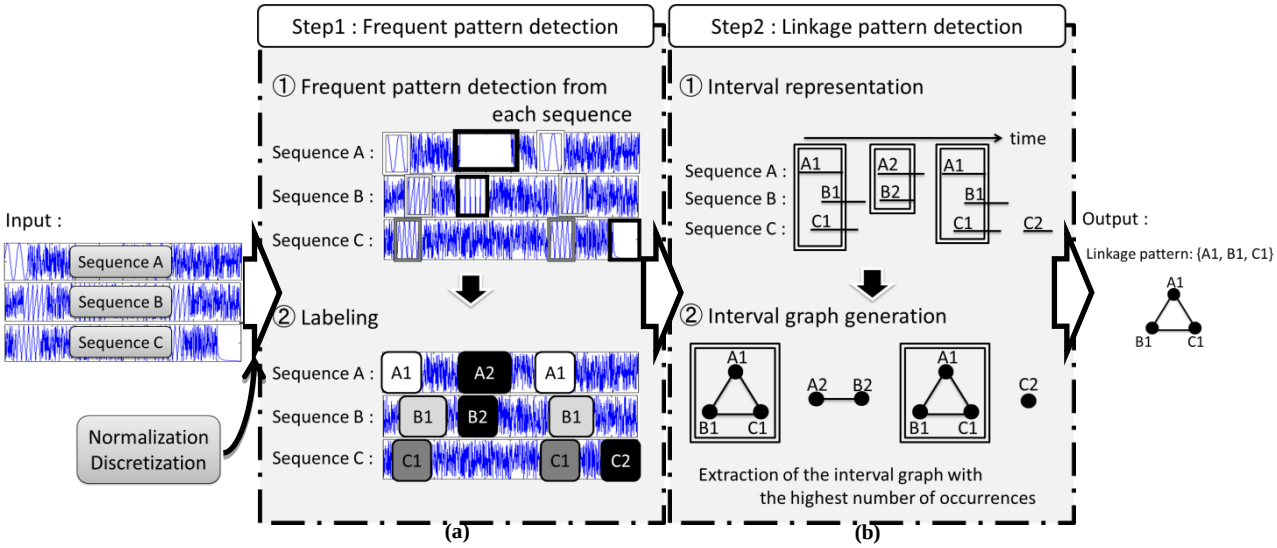
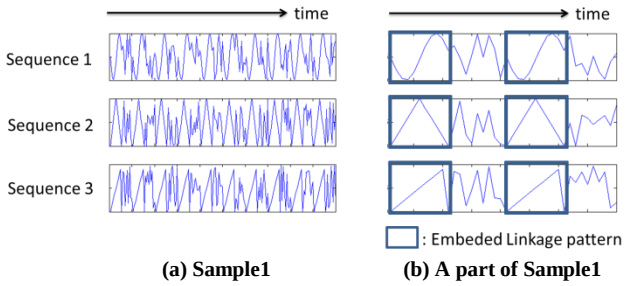


Fig 3: Procedure of the proposed method



(a) Sample1 **(b) A part of Sample1**

Fig 4: An example of artificial data

5. RESULT AND OBSERVATIONS

Figure 5 is a graph of the precision, recall, and F-measure results of this method, as applied to Samples 1, 2, and 3. The graph shows the respective scores for each window width. These results show that, for all indexes, the score increases as the window width narrows. On the other hand, as the window width increases, the score rapidly decreases. In particular, when the maximum window width *w* is set to 15 in Sample 3, the respective indexes cannot be calculated because labels are attached for all time points on the sequential data and accordingly one interval graph is output for entire sequential data. This occurs because, while embedded patterns can be covered and detected, with increasing window size, the labeling is also applied to every data point unrelated to the embedded linkage patterns.

Figure 6 shows examples of correctly and incorrectly detected embedded linkage patterns. In these figures, a cross mark indicates data points in which labels are attached by the frequent pattern detection stage. Areas surrounded by black squares are the areas detected as interval graphs. Those surrounded by an outlined square indicate the areas judged as embedded linkage patterns. From Figure 6(a), it can be seen that when the frequent patterns in each sequential data are correctly detected, interval graph generation and linkage pattern detection are also correctly performed. On the other hand, from Figure 6(b), it can be confirmed that although embedded linkage patterns are broadly covered correctly at the interval graph generation stage, the linkage pattern detection process fails. This is because pseudo frequent patterns formed by random numbers are concatenated with the interval graphs of the embedded linkage patterns. Thus, it will be necessary to exclude the noise in appropriate ways and implement a mechanism that more accurately identifies embedded linkage patterns from interval graphs.

6. SUMMARY

In this paper, we have proposed a method for detecting a linkage pattern that repeats across multiple sequential data, using interval graph representation of frequent patterns. As an experiment, this method was applied to three sets of artificial data and the performance was evaluated using precision, recall, and F-measure. The results show the possibility of appropriately detecting linkage pattern. Further, we found that to improve detection accuracy, it is necessary to incorporate noise reduction process in the identification of a linkage pattern from interval graphs.

In future, a method for highly accurate detection of a linkage pattern from interval graphs will be devised. Further, the presented method will be enhanced to a robust method for sequential data with fluctuations and noises. Additionally, this method will be applied to actual data, such as earthquake wave data or other vital data, to investigate its practical usage in terms of both accuracy and calculation time.

7. ACKNOWLEDGMENTS

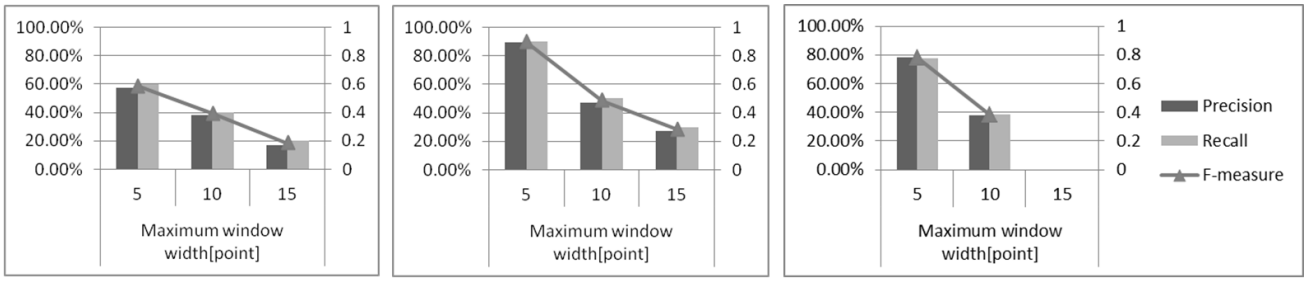
This work was partly supported by Grant-in-Aid for Young Scientists (B) No.24700204 from MEXT Japan.

8. REFERENCES

[1] Tak-chungm F. 2011. A review on time series data mining. Engineering Applications of Artificial Intelligence. Volume 24, Issue 1. 164-181

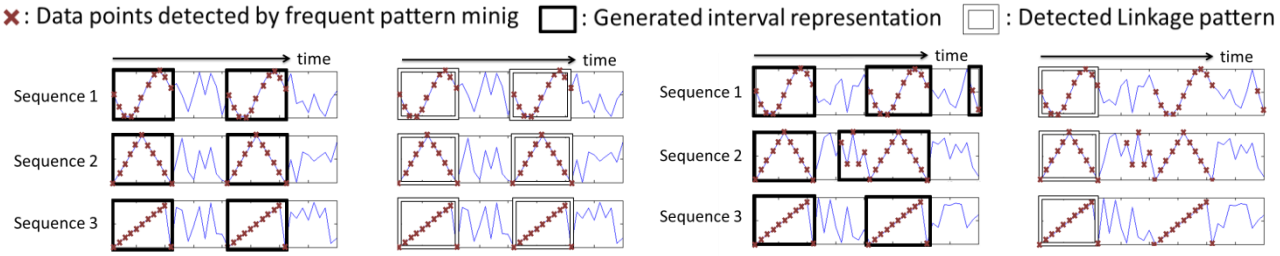
[2] Zhao, Q. and Bhowmick, S. S. 2003. Sequential Pattern Mining: A Survey. Technical Report. CAIS.Nanyang Technological University. Singapore. No. 2003118.

[3] Mannila, H., Toivonen, H. and Verkamo, A.I. 1997. Discovery of Frequent Episodes in Event Sequences. Data Mining and Knowledge Discovery 1. 259-289.



(a) Sample1 **(b) Sample2** **(c) Sample3**

Fig 5: Detection accuracies



(a) Correctly detected Linkage pattern **(b) Incorrectly detected Linkage pattern**

Fig 6: Linkage pattern detection in Sample1

[4] Ohtani, H. Kida, T. Uno, T and Arimura, H. Efficient Serial Episode Mining with Minimal Occurrences. 2009. The Third International Conference on Ubiquitous Information Management and Communication.

[5] Wen-Chi, P and Zhung-Xun, Liao. Mining sequential patterns across multiple sequence databases. 2009. Data & Knowledge Engineering Volume 68, Issue10. 1014-1033.

[6] Gong, C. Xindong, W. and Xingquan, Z. Mining Sequential Patterns across Time Sequences. 2008. New Generation Computing, 26. 75-96.

[7] Sakurai, Y. Faloutsos, C and Yamamuro, M. Stream monitoring under the time warping distance. 2007. In Proc. of ICDE. 1046-1055.

[8] Sakurai, Y., Papadimitriou, S. and Faloutsos, C.2005. BRAID: Stream Mining through Group Lag Correlations. In Proc. of ACM SIGMOD Conference. 599-610

[9] Zhu, Y. and Shasha, D. 2002. StatStream: Statistical Monitoring of Thousands of Data Streams in Real Time. In Proc. Of VLDB.358-369.

[10] Agrawal, R. and Srikant, R.1995. Mining Sequential Patterns. Proc. of The 11th Int'l Conf. on Data Engineering.3-14.

[11] Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U. and Hsu, M.-C. 2001. PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. Proc. of the 17th Int'l Conf. on Data Engineering. 215-224.

[12] Miyoshi, N. Shigezumi, T. Uehara, R and Watanabe, O. 2009. Scale free interval graphs. Theoretical Computer Science Volume 410, Issue 45. 4588-4600.

[13] Korte, N. and Mohring, R.H. 1979. An incremental linear-time algorithm for recognizing interval graphs. SIAM Journal on Computing, vol. 18. 68-81.

[14] Lueker, G.S. and Booth, K.S. 1979. A linear time algorithm for deciding interval graph isomorphism. Journal of the ACM, vol. 26. 183-195.