



멀티플 이벤트를 갖는 멀티플 스트림에서 시간 인터벌 관계와 인과관계 마이닝

황부현¹ · 김재인² · 박재형^{1*}¹전남대학교 전자컴퓨터공학부 · ²한국전자통신연구원 호남연구센터

Causal Relations and Temporal Interval Relations from Multiple Streams with Multiple Events

Buhyun Hwang¹ · Jaemin Kim² · Jaehyung Park^{1*}¹School of Electronics and Computer Engineering, Chonnam National University, Gwangju, 61186, Republic of Korea²Electronics & Telecommunications Research Institute, 176-11 Cheomdan Gwagi-ro, Bukgu, 61012, Republic of Korea

[요 약]

시간 속성을 갖는 데이터집합에서 유용한 정보를 탐사하는 방법으로 빈발 이벤트시퀀스는 이벤트들 사이의 인과관계를 분명하게 표현하지 못한다. 동일한 타입의 이벤트시퀀스가 인터벌이벤트로 요약이 된다면 인터벌 이벤트들로부터 인터벌들의 시간관계를 탐사할 수 있다. 이 논문에서는 시간속성을 갖는 이벤트들로부터 인터벌 시간관계에 근거하여 이벤트들의 인과관계를 마이닝하는 새로운 방법을 제안한다. 동일한 타입의 이벤트에 대한 인터벌이벤트를 독립적인 다수의 인터벌이벤트들로 나누어 보다 많은 인터벌이벤트들의 시간관계를 찾을 수 있도록 한다. 빈발 인터벌 시간관계로부터 빈발 인과관계를 찾는 것($FCR-FIR$, $DFCR-FIRc$)보다 인터벌 시간관계에서 탐사한 인과관계로부터 빈발 인과관계를 찾는 것($DFCR-CRc$)이 더 많은 유용한 정보를 찾을 수 있다. 실험을 통하여 $DFCR-CRc$ 이 보다 많은 유용지식을 탐사한다는 것을 입증하였다.

[Abstract]

Most of these are used to discover frequent event sequences from time-point-based events. However, these frequent event sequences do not clearly represent the causal relationship among events. If a sequence of events of the same type is summarized into one interval-based event, we can discover temporal interval relations from interval-based events. From these temporal interval relations, causal relations among interval-based events can be easily found. In this paper, we proposed a new method that mines causal relations from time-point-based events. Experiments were performed in this study to evaluate three methods ($DFCR-FIR$, $DFCR-FIRc$, and $DFCR-CRc$) to discover the causal relations and temporal interval relations from time-point-based events. $DFCR-FIR$ and $DFCR-FIRc$ discover frequent causal relations from frequent temporal interval relations. $DFCR-CRc$ discovers frequent causal relations from the causal relations. We found that $DFCR-CRc$ provides more useful and effective knowledge than two methods $DFCR-FIR$ and $DFCR-FIRc$.

색인어 : 데이터마이닝, 인과관계, 시간관계, 시간속성이벤트

Key word : data mining, causal relation, temporal relation, time-point-based event

<http://dx.doi.org/10.9728/dcs.2018.19.12.2403>

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 01 December 2018; Revised 14 December 2018

Accepted 23 December 2018

*Corresponding Author; Jaehyung Park

Tel: +82-62-530-1796

E-mail: hyeoung@jnu.ac.kr

I . Introduction

Data mining is a technology used to analyze a large amount of accumulated data and to extract valuable knowledge from the data for decision-making. It can be widely used in areas such as market analysis, decision support, fraud detection, customer classification, medical care, and so on [1], [2], [3], [4], [25]. Recently, many works have focused on the temporal data mining for discovering temporal knowledge from a temporal data set. The knowledge consists of sequential patterns [5], periodic patterns [6], frequent episodes [3], frequent temporal interval relations [7], similarity [8], and causality of events [9], [10], [11], [12], [13], [14], [15].

Temporal data sets can be classified into either a time point-based data set or an interval-based data set. When a time-point-based event data from stream data sources is gathered, its interval or duration to represent its continuity is generally not known. Thus, an interval-based data set is typically obtained from a time-point-based data set. An interval-based event holds summarized information of time-point-based events. From the temporal relations of interval-based events, called “temporal interval relations”, more important knowledge can be discovered such as the causality of events.

Also, a temporal data set can be classified into four categories according to the number of streams and the number of concurrent events as follows:

1. Single Stream and Single Event (SSSM) [3]; SSSM is a set of events from single stream, where one time slot contains only one event.
2. Single Stream and Multiple Events (SSME) [4], [6], [8]; SSME is a set of events from a single stream, where one time slot contains multiple events.
3. Multiple Streams and Single Events (MSSE); MSSE is a set of events from multiple streams, where one time slot contains only one event.
4. Multiple Streams and Multiple Events (MSME) [7], [9], [16]; MSME is a set of events from multiple streams, where one time slot contains multiple events.

The sequential pattern mining problem was introduced by Agrawal and Srikant [17]. Sequential pattern mining is used to find all frequent subsequences from a set of sequences. From frequent patterns, we can find the order of events. This order can be used in a business strategy. Examples of interval-based event data include library lending, stock fluctuation, patient diseases, meteorology data, etc. Wu and Chen [7] defined a new type of non-ambiguous temporal pattern for interval-based event data. They found that patterns discovered by the Allen-based

approach have ambiguity problems. To solve the ambiguity problems, he proposed the TPrefixSpan algorithm.

In this paper, we propose a new temporal data mining method that discovers temporal interval relations between interval-based events which are computed from a time-point-based data set. Also, we will discuss the casual relations among interval-based events. The objective of our research is to develop an algorithm to process big-data from a selected hospital. Each department in the hospital treats many patients and produces medical records for the patients. Each medical record has several events (symptoms or the result of check-ups). Each patient is a stream source and one medical record contains many events. Therefore, we selected the MSME data set.

The basic concept of our method is summarized as follows, in which temporal interval relations and causal relations are discovered from a MSME data set:

1. Compute frequent event types from MSME data set
2. For each stream and each frequent event type, compute continuous event sequences from MSME data set
3. Summarize each continuous event sequence into one interval-based event
4. Discover temporal interval relations from interval-based events
5. Discover causal relations from temporal interval relations
6. Discover frequent causal relations from a set of causal relations
7. Compute the net effect of events on other events from component graphs for frequent causal relations.

Generally, patient medical records are stored in a hospital database in the timestamp order. From the hospital database, we aim to discover rules (or knowledge) to properly treat patients. Our method can be easily applied to the medical field for discovering useful and effective knowledge (frequent causal relations).

The remainder of the paper is organized as follows. In Chapter II, we discuss the related works. In Chapter III, the terminologies and definitions related to temporal relations and causal relations are described. In Chapter IV, our algorithm for mining temporal interval relations among interval-based events and their causal relations is described. In Chapter V, the proposed method is analyzed through the experiments. In Chapter VI, the conclusion and future works are discussed.

II . Related Works

Temporal data mining discovers useful knowledge from a temporal data set. Temporal data mining methods and their

mining results can contain frequent event sequences [4], [16], frequent episodes [3], periodic patterns [6], frequent temporal interval relations [7], causality of events [9], [10], [11], [12], [13], [14], [15], or similarity [8]. Several variations of the Apriori algorithm have been proposed to improve basic sequential pattern mining algorithms such as GSP [17] and SPIRIT [18]. GSP is a technique used to discover generalized sequential patterns that incorporate user-specified time constraints. SPIRIT is a technique used for discovering all frequent sequences that satisfy a user-defined regular expression constraint. An episode has a collection of events with a partial order. Mannila, et al. [19] introduced the framework of frequent episodes discovery in event sequences. Laxman et al. [3] introduced generalized episodes. A generalized episode incorporates event duration information explicitly with the episode structure. Each event in the episode has a duration which is previously determined by the domain specialists. Frequent generalized episodes assist the diagnosis of root causes of persistent fault situations. Periodic patterns are recurring patterns that have temporal regularities in a time-series data set. Huang et al. [6] proposed a more general model of asynchronous periodic patterns from a sequence of event sets where a time slot can contain multiple events.

More recently, studies on mining useful patterns from temporal interval data have been initiated [20], [21], [22]. However, since the temporal relation rule discovery from interval data is very complicated, no noticeable progress has been made thus far. Also, summarizing a sequence of events into one interval-based event is unreasonable because some events in the sequence cannot be continuous or persistent. The sequence of events needs to be split into continuous subsequences and each subsequence needs to be summarized into one interval-based event. Periodic patterns exist in many types of data. For example, tides, planet trajectories, daily traffic patterns, and power consumptions present certain periodic patterns. Many emerging applications have been made, including stock market price movement, earthquake prediction, telecommunication network fault analysis, repeat detection in DNA sequences, and occurrence of recurrent illnesses. In [9], [14], [15], a method was proposed to detect adverse drug reactions. This is a domain-driven data mining method.

In [22], a method for finding the temporal interval relation rules from a temporal data set was proposed. The method can extract temporal interval relation rules from a temporal interval data set by using Allen's theory. However, the method does not measure the extent to which one event affects another event. If we can determine the degree of influence of each event, we can find a major source event for a specific event. If we have

knowledge of the time-gap between interval-based events, we can also find very useful knowledge that can predict the occurrence time of future events. In [23], a method of mining positive and negative association rules was proposed. Positive association rules have a positive effect and negative association rules have a negative effect on association rules. However, there is a limit to analyzing the causality among events since the method computes an influence using association rules based only on the support of events and the confidence of association rules. Thus, it is necessary to develop a new measurement to represent the causality among events. Terminologies and definitions related to temporal interval relations and causal relations will be given in this section. Also, the way in which causal relations are discovered from temporal interval relations will be discussed.

III. Temporal Interval Relation and Causal Relations

Terminologies and definitions related to temporal interval relations and causal relations will be given in this section. Also, the way in which causal relations are discovered from temporal interval relations will be discussed.

3-1 Terminologies and Definitions

A temporal data set DB is a database of time-stamped transactions. In this paper, a patient and a customer will be interchangeably used as an issuer of a transaction. In general, for one medical check-up of a patient, one medical record will be produced. The medical record is a transaction. A patient can be considered as a source of a stream of data (or transactions). Let TS and ETS be a set of primitive time-points and a set of event types, respectively. An event e is defined by triple attributes such as $e = (Cid, E, t)$, where Cid is a customer who issues an event e , $E(E \in ETS)$ is an event type, and $t(t \in TS)$ is a time-point at which an event e has occurred. Examples of events include patient symptoms, the failures of products, environmental changes, visited web pages, etc. A transaction is a set of events which have occurred at a time-point. Each transaction has attributes such as a customer Cid , a transaction-time t , and a set of events ES . A patient can sequentially issue several transactions. For example, a patient can periodically take a medical check-up. Each medical check-up is represented by a transaction. One medical check-up can show multiple symptoms. Each symptom is an event in a transaction. Each customer can issue only one transaction at

one time-point. Thus, all events in a transaction have the same timestamp. DB is a set of transactions which are expressed by a form (Cid, ES, t) , where ES is a set of events contained in a transaction which is issued at a time-point t by a customer Cid . Transactions contained in DB are ordered by Cid and t . Let a sequence of transactions issued by a customer Cid be $SeqTrans(Cid) = \langle T1, T2, \dots, Tn \rangle$. An event sequence consists of the same type of events selected from $SeqTrans(Cid)$. A sequence of events for a customer Cid and event type E is defined, as in *Definition 1* below. In this paper, an event sequence is a sequence of the same type of events.

Temporal relations between time-point-based events cannot easily represent the causality between events since events generally persist for some duration. To capture the overall temporal relation among events, we will focus on the sequences of events.

Definition 1. (An event sequence for a customer Cid and event type E) An event sequence for a customer Cid and event type E , $ESeq(Cid, E)$, is defined by a sequence $\langle e1, e2, \dots, em \rangle$, where $ei = (Cid, E, ti)$, $ti < ti+1$ for each $i = 1, 2, \dots, m-1$. $e1 \in T1 \in T(Cid)$, $e2 \in T2 \in T(Cid)$, \dots , $em \in Tm \in T(Cid)$, and $Etype(e1) = Etype(e2) = \dots = Etype(em) = E$. $Etype(e)$ denotes the type of event e . $T(Cid)$ is a set of transactions issued by a customer Cid . Also, a set of event sequences for a customer Cid is defined by $SS(Cid) = \{ESeq(Cid, E) | E \in ETS(Cid)\}$. $ETS(Cid)$ is a set of event types issued by a customer Cid . A set of event sequences is defined as follows. $DBseq = \cup_{Cid \in Cust} SS(Cid)$, where $Cust$ is a set of customers.

From DB , we can obtain a database of event sequences, $DBseq$, defined by *Definition 1*. Let a set of customers be $Cust = \{Cid_1, Cid_2, \dots, Cid_n\}$. $DBseq$ is represented by $SS(Cust) = SS(Cid_1) \cup SS(Cid_2) \cup \dots \cup SS(Cid_n)$.

In this paper, we will consider only the event sequences of the same type of events of each customer. If the events in the sequence persist, it is reasonable to summarize each event sequence into one interval-based event defined by *Definition 2*. When an event E is included in $SS(Cid)$, we state that a customer Cid supports E . The support count of E is denoted by $SupCount(E)$ and it is the number of customers supporting E in DB . $Sup(E)$ is represented by an expression $SupCount(E)/Ncust$. $Ncust$ is the number of customers in DB . If $Sup(E) \geq MinSup$ (the minimum support chosen by a user), E is a frequent event. $MinSup$ is the ratio of the support count for the total number of customers. $MinSup$ is determined by applications. If an application requires high $MinSup$, highly frequent events are considered for mining temporal relations. As the $MinSup$

increases, the number of general temporal relations we can obtain from a data set also increases.

Definition 2. (An interval-based event) An interval-based event, ie , is denoted as $ie = (Cid, E, [vs, ve])$, where vs and ve represent the start time and end time of an event E , respectively. For an interval-based event ie , its start and end time are represented as $ie.vs$ and $ie.ve$, respectively. A set of interval-based events for a customer Cid is denoted as $IES(Cid)$.

An event sequence can be converted into one interval-based event if its event is continuous or persistent. To meaningfully summarize an event sequence into an interval-based event, the time-gap ($Tgap$) between adjacent events in the event sequence must be less than a given time-threshold value τ_{gap} . The time-threshold value τ_{gap} can differ according to the application areas. We can define $Constraint_{Tgap}$ as the expression, $Tgap < \tau_{gap}$. If $Constraint_{Tgap}$ is not satisfied, we can consider that the event is not continuous in the time-gap. For example, if there are two symptoms, a cold (due to influenza) on the 1st of January and a cold on the 1st of May, the first cold would not generally persist up to the 1st of May.

For each sequence in $SS(Cust)$, if it is continuous, it can be summarized into an interval-based event. An interval-based event means that the event occurs continuously in its time interval. A continuous event sequence $ESeq(Cid, E)$ can be summarized into one interval-based event $(Cid, E, [vs, ve])$. $(Cid, E, [vs, ve])$ means that an event E has occurred continuously in the temporal interval $[vs, ve]$.

Summarizing a discontinuous event sequence into one interval-based event is not reasonable. For example, if the time-gap between adjacent events in an event sequence is larger than a given time-threshold value τ_{gap} , we can assume that the two events around the time-gap are discontinuous within the time-gap. Thus, it is reasonable to divide the discontinuous sequence into continuous subsequences and summarize each subsequence into one interval-based event. For example, let an event sequence be $ES(Cid, E) = \langle (Cid, E, 1) (Cid, E, 3) (Cid, E, 4) (Cid, E, 8) (Cid, E, 10) \rangle$. If a given time-threshold τ_{gap} is 3, the sequence is split into two subsequences $\langle (E, 1)(E, 3)(E, 4) \rangle$ and $\langle (E, 8)(E, 10) \rangle$. These are then summarized into $(Cid, E, [1, 4])$ and $(Cid, E, [8, 10])$, respectively.

3-2 Temporal Interval Relations

For two interval-based events, x and y , they have a binary temporal interval relation, $\Phi(x, y)$, where Φ denotes a binary temporal interval relation between two interval-based events x

and y . The binary temporal interval relations are based on the temporal relation operators proposed by Allen [24]. To discover causal relations, we will consider only three operators of Allen's temporal relation operators as in *Definition 3* below.

Definition 3. (*Temporal interval relation*) A temporal interval relation $\Phi(x, y)$ is defined as follows. Φ is an operator in a set of operators such as *before*, *during*, and *overlap*.

- $before(x, y) : (x.ve \leq y.vs)$
- $overlap(x, y) : (x.vs < y.vs) \wedge (x.ve < y.ve) \wedge (y.vs < x.ve)$
- $during(x, y) : (y.vs \leq x.vs) \wedge (x.ve \leq y.ve)$

The following operators such as '*equals*' and '*meets*' of Allen's temporal relation operators can be represented by '*during*' and '*before*', respectively.

- $equals(x, y) : (x.vs = y.vs) \wedge (x.ve = y.ve)$
- $meets(x, y) : x.ve = y.vs$

We can assume that $equals(x, y)$ is a special case of $during(x, y)$ and $meets(x, y)$ is a special case of $before(x, y)$. However, from $equals(x, y)$, we cannot determine the cause and effect between them because their start-times are identical.

3-3 Causal Relations

Knowledge of the causal relations of events is very valuable for predicting future events. We will define causal relations between interval-based events as shown in *Definition 4*.

Definition 4. (*Causal relation between interval-based events x and y*) A causal relation occurs between x and y if one of the following conditions is satisfied.

- Condition 1: $before(x, y) \wedge Constraint_{before}$, where $Constraint_{before}$ is $(y.vs - x.ve \leq t_{before})$
- Condition 2: $overlap(x, y) \wedge Constraint_{overlap}$, where $Constraint_{overlap}$ is $(y.vs - x.vs \geq t_{overlap})$
- Condition 3: $during(x, y) \wedge Constraint_{during}$, where $Constraint_{during}$ is $(x.vs - y.vs \geq t_{during})$

A causal relation between interval-based events x and y is denoted by the notation " $x \rightarrow y$ ". t_{before} , $t_{overlap}$, and t_{during} are time-threshold values determined by domain experts. They are required for temporal relations to become effective casual relations.

The constraints in *Definition 4* are determined by domain experts. " $before(x, y) \wedge (y.vs - x.ve) \leq t_{before}$ " indicates that an event x precedes another event y while satisfying a given $Constraint_{before}$. If $Constraint_{before}$ is not satisfied, then a

sufficient time gap occurs in which an event x does not have an effect on event y . " $overlap(x, y) \wedge (y.vs - x.vs) \geq t_{overlap}$ " shows that after an event x occurred, an event y has occurred after at least a given time-threshold value $t_{overlap}$ has elapsed. In addition, y continues until after x has finished. Thus, we can assume that x is the cause of y . " $during(x, y) \wedge (x.vs - y.vs) \geq t_{during}$ " indicates that an event x occurs and ends in the lifetime of event y and event x must occur after at least the time threshold value t_{during} from the start of event y . Also, we can assume that y is the cause of x . If $Constraint_{overlap}$ and $Constraint_{during}$ are satisfied, enough time must have elapsed for the "cause-and-effect" relation to come into being. Time-threshold values t_{before} , $t_{overlap}$, and t_{during} can differ according to the applications.

IV. Algorithm for Mining Temporal Interval Relations

In this section, we introduce an algorithm for mining temporal interval relations among interval-based events and their causal relations. It consists of the summarization of event sequences into interval-based events, as well as the discovery of temporal interval relations among interval-based events, and their causal relations. From causal relations, the degree of the cause-and-effect between interval-based events will be computed.

4-1 Algorithm

DB is a database of transactions. Each transaction consists of a customer Cid , a transaction-time t , and a set of events. A customer can issue one transaction at one time-point. That is, all of the timestamps of transactions issued from a customer differ. All events in a transaction have the same timestamp. Our algorithm can be summarized as shown in Table 1. In Step 3, a sequence of time-point-based events was split into subsequences if the sequence is not continuous. If a discontinuous sequence of the same type of events is summarized into one interval event, the interval event might not reasonably represent the actual interaction of events, as in Example 1. In Step 5, $t_{interval-gap}$ is needed to remove meaningless temporal interval relations. $t_{interval-gap}$ is greater than t_{gap} .

Example 1. Suppose that event sequences occur such as $ES(Cid, A) = \langle (Cid, A, 1)(Cid, A, 3)(Cid, A, 4)(Cid, A, 10)(Cid, A, 12) \rangle$, $ES(Cid, B) = \langle (Cid, B, 3)(Cid, B, 6) \rangle$, and $ES(Cid, C) = \langle$

(*Cid*, *C*, 7)(*Cid*, *C*, 9) >.

Case 1: Tgap is not considered.

Three interval events <*Cid*, *A*, [1,12]>, <*Cid*, *B*, [3,6]>, and <*Cid*, *C*, [7,9]> are discovered. From these three interval events, the causal relations, $A \rightarrow B$ and $A \rightarrow C$, are discovered.

Case 2: Tgap is considered and τ_{gap} is 3.

<*Cid*, *A*, [1,12]> is split into two interval relations, <*Cid*, *A*, [1,4]> and <*Cid*, *A*, [10,12]>. In this case, causal relations, $A \rightarrow B$ and $C \rightarrow A$, are discovered.

Case 2 is more reasonable than Case 1 because an event *A* does not persist in an interval [5,9] ($T_{gap} > \tau_{gap}$). Case 1 and Case 2 have different meanings because their causal relations differ.

All causal relations among events can be expressed by a directed graph, called *CR-Graph*. *CR-Graph* is defined in *Definition 5*. In *CR-Graph*, a node represents an interval-based event and an edge denotes a causal relation. Also, each edge $x \rightarrow y$ has a label $Cust(x \rightarrow y)$, which is a set of customers who support the causal relation.

Definition 5. (Causal Relation Graph: CR-Graph) *CR-Graph* is a directed graph. Each node represents an interval-based event. Each edge represents the causal relation between two interval-based events. The label on an edge denotes a set of customers who support the causal relation.

We can find frequent component graphs for causal relations from *CR-Graph* by examining the labels on the edges of *CR-Graph*. Assume that a component graph ($A \rightarrow C \leftarrow B$) in *CR-Graph* is frequent and its support count is 50. If $Cust(A \rightarrow C)$ and $Cust(B \rightarrow C)$ have 100 customers and 50 customers, respectively, and the support counts of interval-based events *A* and *B* are 200 and 50, respectively, we can infer that event *B* is the essential cause of *C* rather than event *A* because *C* always occurs when *B* occurs.

Discovering the cause-and-effect among events from causal relations considering only their supports has some shortcomings. Assume that the support of causal relation $A \rightarrow B$ and the support of event *A* are 0.3 and 0.3, respectively. If the confidence is used as the measure of causality, we can only infer that event *A* has an absolute effect on event *B* since the confidence of causal relation $A \rightarrow B$ is 1. If the support of *B* is 0.7, *B* can be due to other causes. Thus, we need a new evaluation measurement to show the rate of the effect of *A* on *B*. Let $A \rightarrow B$ be a causal relation in *CR-Graph*. We will define a new measurement $NetEff(A \rightarrow B)$ which computes the rate of the net effect of *A* on *B*. The measurement $NetEff(A \rightarrow B)$ is defined in *Definition 6*.

Definition 6. Net effect of *X* on *B*, $NetEff(X \rightarrow B)$, is defined as follows.

$$NetEff(X \rightarrow B) = \frac{1}{2} \left(\frac{sup(X \rightarrow B)}{sup(X)} + \frac{sup(X \rightarrow B)}{sup(B)} \right) - \max \left\{ \frac{1}{2} \left(\frac{sup(Y \rightarrow B)}{sup(Y)} + \frac{sup(Y \rightarrow B)}{sup(B)} \right) \mid Y \rightarrow B \in FCR \wedge X \neq Y \right\}$$

where *FCR* is a set of frequent causal relations.

$$Expression 1: Eff(X \rightarrow B) = \frac{1}{2} \left(\frac{sup(X \rightarrow B)}{sup(X)} + \frac{sup(X \rightarrow B)}{sup(B)} \right)$$

$$Expression 2: \max \left\{ \frac{1}{2} \left(\frac{sup(Y \rightarrow B)}{sup(Y)} + \frac{sup(Y \rightarrow B)}{sup(B)} \right) \mid Y \rightarrow B \in FCR \wedge X \neq Y \right\}$$

Part $\frac{sup(X \rightarrow B)}{sup(X)}$ of *Expression 1* denotes the rate that an event *B* can occur when an event *X* occurs. Part $\frac{sup(X \rightarrow B)}{sup(B)}$ of *Expression 1* denotes the portion of *B* affected by *X*. $Eff(X \rightarrow B)$ is represented by an average of two parts. *Expression 2* is the maximum effect of other events on *B*. The net effect of *X* on *B* is represented by $Eff(X \rightarrow B) - \max \{ Eff(Y \rightarrow B) \mid Y \rightarrow B \in FCR \wedge X \neq Y \}$. By using *NetEff*, we can order the net effects of interval-based events in a component graph of *CR-Graph*. The ordered *NetEff*s of causal relations in a component graph can be used to discover the major source events for a target event.

Our algorithm (see Table 1) discovers a set of frequent causal relations from a set of temporal interval relations. We call our algorithm *DFCR-CRc* (Discovering Frequent Causal Relations from Causal Relations which can be from infrequent temporal interval relations). We can consider another algorithm to discover causal relations from only frequent temporal relations, called *DFCR-FIRc* (Discovering Causal Relations from Frequent Temporal Interval Relations). *DFCR-CRc* discovers more frequent causal relations than *DFCR-FIRc* because different temporal interval relations (which can be infrequent) can be transformed into the same causal relation.

Table 1. Algorithm (DFCR-CRc) to discover frequent causal relations from causal relations

Input: a set of time-point-based events

Output: frequent causal relations

Step1: Computing a set of frequent event types.

Let *DBsort* be a set of transactions sorted according to customer identifiers and their timestamps. $Count(E)$ is a support count for an event *E*, *Cust* is a set of customers, and *FETS* is a set of frequent event types.

for each customer *Cid* in *DBsort* **begin**

for each event type *E* in *ETS* **begin**

if *E* is in $ETS(Cid)$, $count(E)++$;

end

end

for each event type *E* in *ETS* **begin**

if $(count(E)/|Cust|) > minsup$, add *E* to *FETS*;

end

delete all event types not in *FETS* from *DBsort*;

Step2: Computing a set of event sequences, $SS(Cust)$

for each customer Cid in DB_{sort} **begin**

for each event type E in $EType(Cid)$ **begin**

compute an event sequence $ESeq(Cid, E)$

for Cid and E ;

add $ESeq(Cid, E)$ to $SS(Cid)$; // $SS(Cid) = \{ESeq(Cid, E) \mid E \in FETS(Cid)\}$

end

add $SS(Cid)$ to $SS(Cust)$; // $SS(Cust) = SS(Cid1) \cup SS(Cid2) \cup \dots \cup SS(Cidn)$

end

Step3: Computing a set of continuous event sequences, $CSS(Cust)$

for each event sequence S in $SS(Cust)$ **begin**

let S be $\langle (C_{id}, E, t_1) (C_{id}, E, t_2) \dots (C_{id}, E, t_i) (C_{id}, E, t_{i+1}) \dots (C_{id}, E, t_n) \rangle$

for two adjacent events (C_{id}, E, t_i)

and (C_{id}, E, t_{i+1}) of S **begin**

if $(t_{i+1} - t_i) > \tau_{gap}$ **begin**

split S into $S_1 = \langle (C_{id}, E, t_1) (C_{id}, E, t_2) \dots (C_{id}, E, t_i) \rangle$ and

$S_2 = \langle (C_{id}, E, t_{i+1}) \dots (C_{id}, E, t_n) \rangle$;

add S_1 into $CSS(Cust)$;

end

$S = S_2$;

end

end

Step4: Summarizing each event sequence in $CSS(Cust)$ into an interval-based event

for each continuous event sequence S

in $CSS(Cust)$ **begin**

let S be $\langle (Cid, E, t_1) (Cid, E, t_2) \dots (Cid, E, t_n) \rangle$;

summarize S into an interval-based event

$IE = (Cid, E, [t_1, t_n])$;

add IE to $DB_{interval-evts}$;

end

Step5: Computing a set of temporal interval relations $IR(Cust)$ from interval-based events

in $DB_{interval-evts}$

for each customer Cid in $DB_{interval-evts}$ **begin**

for any two interval-based events

with Cid , x and y **begin**

if $(x.ve \leq y.vs) \wedge (y.vs - x.ve) < \tau_{interval-gap}$,

add $before(x, y)$ to $TR(Cid)$;

if $(x.vs < y.vs) \wedge (x.ve < y.ve) \wedge (y.vs < x.ve)$,

add $overlap(x, y)$ to $TR(Cid)$;

if $(y.vs \leq x.vs) \wedge (x.ve \leq y.ve)$,

add $during(x, y)$ to $TR(Cid)$;

end

end

end

compute $IR(Cust) = \bigcup_{Cidi \in Cust} IR(Cidi)$;

Step6: Computing a set of causal relations $CR(Cust)$ from $IR(Cust)$

for each customer Cid in $TR(Cust)$ **begin**

for each temporal relation tr in $TR(Cid)$ **begin**

if tr is $before(x, y) \wedge (y.vs - x.ve \leq \tau_{before})$,

add a causal relation $x \rightarrow y$ to $CR(Cid)$;

if tr is $overlap(x, y) \wedge (y.vs - x.vs \geq \tau_{overlap})$,

add a causal relation $x \rightarrow y$ to $CR(Cid)$;

if tr is $during(x, y) \wedge (x.vs - y.vs \geq \tau_{during})$,

add a causal relation $y \rightarrow x$ to $CR(Cid)$;

end

end

compute $CR(Cust) = \bigcup_{Cidi \in Cust} CR(Cidi)$.

Step7: Computing a set of frequent causal relations $FCR(Cust)$ from $CR(Cust)$;

Step8: Discovering frequent component graphs;

Step9: Computing the degree of the cause-and-effect between events from frequent component graphs according to *Definition 6*.

V. Experimental Results

From Experiments were carried out while varying time-constraint parameters such as τ_{gap} , τ_{before} , $\tau_{overlap}$, τ_{during} , and $MinSupport$. The number of interval events, temporal interval relations, and causal relations will be computed by varying the parameters. The experiments are performed mainly for analyzing: (1) the *DFCR-FIR* method, for discovering frequent causal relations from frequent interval relations which have been computed without time-constraint (τ_{gap}) between the adjacent same type events, (2) the *DFCR-FIRc* method, for discovering frequent causal relations from frequent interval relations which have been computed with the time-constraint between the adjacent same type of events, and (3) the *DFCR-CRc* method, for discovering frequent causal relations from causal relations which have been computed from interval relations with the time-constraints between the adjacent same type of events.

The description of data set is shown as in Table 2. The data set has been artificially generated. Each transaction in the data set consists of three fields: transaction *ID*, transaction time, and a set of events.

Table 2. Description of the Data Set

Number of event types	100
Number of customers	1000
Number of transactions	7018
Maximum length of transactions	15

From the experiments, we analyze the change of the followings.

- (1) Number of interval events for customers (#IE)
- (2) Number of interval relations for customers (#IR)
- (3) Number of causal relations for customers (#CR)

Also, through the experiments, we will show the changes of the number of interval relations for each interval relation (IR) type. Five types of interval relations exist (IR_{before} , $IR_{overlap}$, IR_{during} , $IR_{duringx}$, and IR_{equal}). To find causal relations, only three IR types (IR_{before} , $IR_{overlap}$, IR_{during}) will be considered. $IR_{duringx}$ is a *during*(X,Y) where $X.vs=Y$.vs. Therefore, we cannot discern X and Y as a cause event. Also, from *equal*(X,Y), we cannot discern X and Y as a cause event.

Table 3. Descriptions for some abbreviations

IE	a set of interval events computed without considering time-gap (T_{gap}) constraint between adjacent events in a sequence of time-point-based events of the same type
IEc	s set of interval events computed with considering time-gap (T_{gap}) constraint between adjacent events in a sequence of time-point-based events of the same type
IR	a set of temporal interval relations discovered from IE
IRc	a set of temporal interval relations discovered from IEc
FIR	a set of frequent interval relations gotten from IR
FIRc	a set of frequent interval relations gotten from IRc
CRc	a set of causal relations gotten from IRc
FCR	a set of frequent causal relations discovered from FIR
FCR-FIRc	a set of frequent causal relations discovered from FIRc
FCR-CRc	a set of frequent causal relations discovered from CRc.

To discover frequent causal relations, we can choose one of three methods ($DFCR-FIR$, $DFCR-FIRc$ or $DFCR-CRc$) as follows. The notation “ \rightarrow ” denotes computation flows.

Method $DFCR-FIR$: $IE \rightarrow IR \rightarrow FIR \rightarrow FCR$

Method $DFCR-FIRc$: $IEc \rightarrow IRc \rightarrow FIRc \rightarrow FCR-FIRc$

Method $DFCR-CRc$: $IEc \rightarrow IRc \rightarrow CRc \rightarrow FCR-CRc$

$Constraint_{T_{gap}} (T_{gap} < \tau_{gap})$ must be satisfied to split a sequence of the same type of events into subsequences. If τ_{gap} is infinite, the sequence of events is transformed into one interval event. This might not be reasonable because an event might intermittently (rather than continuously) occur. The values of parameters such as τ_{before} , $\tau_{overlap}$, and τ_{during} are needed to discover a reasonable causal relation between two temporal interval events. Their values are generally determined by domain specialists.

While varying τ_{gap} , we analyze the number of causal relations for two methods ($DFCR-FIRc$ and $DFCR-CRc$). The $DFCR-FIR$ method is the special case of $DFCR-FIRc$ (the case τ_{gap} is infinite).

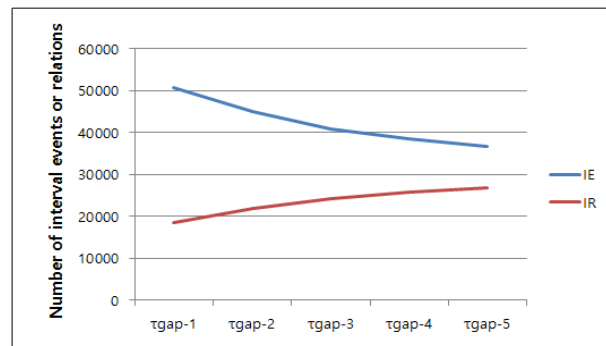


Fig. 1. Number of interval events or interval relations

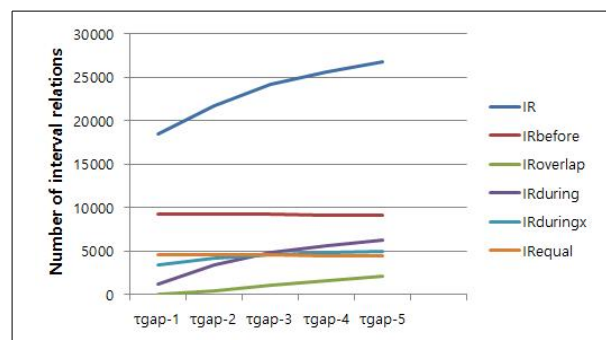


Fig. 2. Number of interval relations for IR types

We show that the number of frequent causal relations discovered by $DFCR-CRc$ is more than that by $DFCR-FIR$ or $DFCR-FIRc$. In other words, the knowledge discovered by $DFCR-CRc$ is much richer than that discovered by $DFCR-FIR$ or $DFCR-FIRc$.

In Fig. 1, we can see that the number of interval events is decreased when τ_{gap} is increased. However, we found that the number of temporal interval relations is increased while τ_{gap} increases. As the length of the τ_{gap} increases, the duration of the interval events also increases. We found that there can be more temporal interval relations between long interval events and short interval events than those between only short interval events. If long interval events do not clearly represent the persistency of events, their temporal interval relations are also not reasonable.

As shown in Fig. 2, when τ_{gap} is increased, three types of interval relations, including $IR_{overlap}$, IR_{during} , and $IR_{duringx}$ have an effect on the increase of the number of temporal interval relations. For example, long duration interval events can be more closely related to short duration events by “during”, “duringx”, or “overlap”. From Fig.1, we can see that the size of IR is larger than that of IRc , because IR is computed when τ_{gap} is infinite.

Fig. 3 shows that for all values of τ_{gap} , $FCR-CRc$ contains significantly more information than $FCR-FIR$ and $FCR-FIRc$. FCR is a set of causal relations when τ_{gap} is infinite (∞). As a result, $DFCR-CRc$ discovers much more information from the time-point data set.

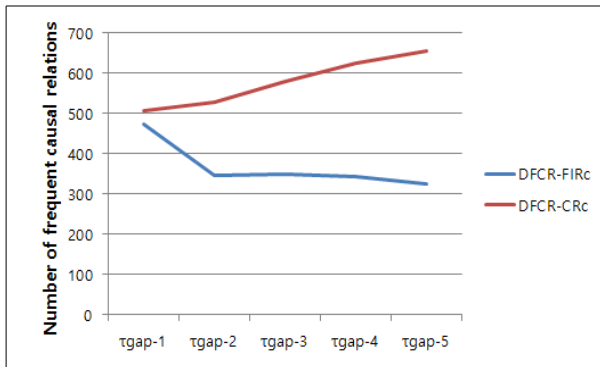


Fig. 3. Number of causal relations for varying τ_{gap}

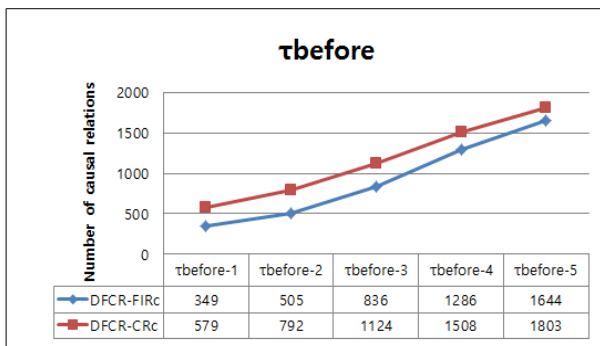


Fig. 4. Number of causal relations discovered while varying a parameter τ_{before} (for the following fixed

parameters of $\tau_{gap}=3$, $\tau_{interval-gap}=3$, $\tau_{during}=1$, $\tau_{overlap}=1$, $MinSupport=0.1$)

To determine the interval relation types that have an effect on the frequent causal relations, we inspect the change of the number of frequent causal relations while varying the time-constraints of τ_{before} , $\tau_{overlap}$, and τ_{during} . Only these three constraints have an effect on causal relations.

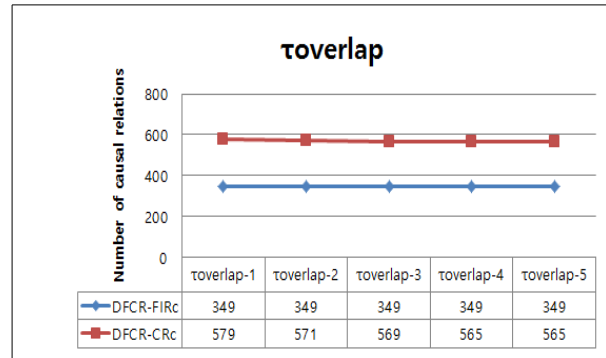


Fig. 5. Number of causal relations found while varying a parameter $\tau_{overlap}$ (for the following fixed parameters such as $\tau_{gap}=3$, $\tau_{interval-gap}=3$, $\tau_{during}=1$, $\tau_{before}=1$, $MinSupport=0.1$)

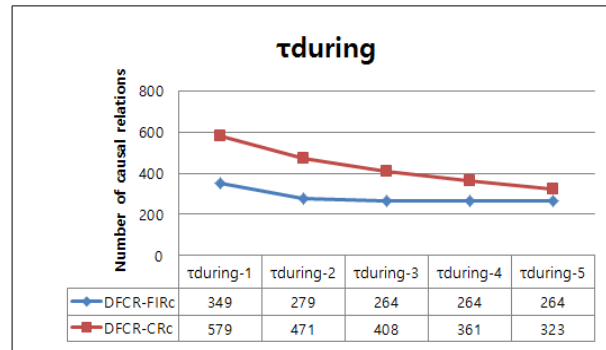


Fig. 6. Number of causal relations discovered while varying a parameter τ_{during} (for the following fixed parameters such as $\tau_{gap}=3$, $\tau_{interval-gap}=3$, $\tau_{during}=1$, $\tau_{before}=1$, $MinSupport=0.1$)

Also, we compare only two methods, $DFCR-CRc$ and $DFCR-FIRc$, because $DFCR-FIR$ is a special case of $DFCR-FIRc$ when τ_{gap} is infinite. In Fig. 4, for the given parameter values of $\tau_{gap}(=3)$, $\tau_{during}(=1)$, $\tau_{overlap}(=1)$, and $MinSupport(=0.1)$, we find the degree to which a parameter τ_{before} has an effect on the number of causal relations. In Fig. 5, for given parameter values of $\tau_{gap}(=3)$, $\tau_{during}(=1)$, $\tau_{before}(=1)$, and $MinSupport(=0.1)$, we find the degree to which a parameter $\tau_{overlap}$ has an effect on the number of causal relations. In Fig. 6, for given parameter values of $\tau_{gap}(=3)$, $\tau_{overlap}(=1)$, $\tau_{before}(=1)$,

and $MinSupport(=0.1)$, we find how a parameter τ_{during} has an effect on the quality of causal relations. As shown in Fig. 4, Fig. 5, and Fig. 6, for all parameter values, *DFCR-CRc* discovers considerably more causal relations than *DFCR-FIRc*. This is because different infrequent interval relations can be transformed into the same causal relation. For example, $overlap(A,B)$ and $before(A,B)$ produce the same causal relation of $A \rightarrow B$. Let $\#rel(X,Y)$ be the number of a relations $rel(X,B)$. Even though both temporal interval relations are infrequent, if $(\#overlap(A,B) + \#before(A,B)) > MinSupportCount$, the causal relation $A \rightarrow B$ is frequent.

As shown in Fig. 4, when τ_{before} increases, the number of causal relations increases in *DFCR-CRc* and *DFCR-FIRc* because $Constraint_{before}$ becomes stronger when the value of τ_{before} decreases. For example, for two interval-based events, $(A,[8,10])$ and $(B,[12,14])$, if $\tau_{before}=1$, there is no temporal interval relation, but if $\tau_{before}=2$, there is a temporal interval relation $before(A,B)$ and a causal relation $A \rightarrow B$. As shown in Fig. 5, for $\tau_{overlap}$, when its value is increasing, the number of causal relations is almost the same, but the number of causal relations for a high value is always less than that of its low value. This is because $Constraint_{overlap}$ becomes stronger when the value of $\tau_{overlap}$ increases. For example, for two interval-based events $(A, [10,14])$ and $(B, [12,15])$, if $\tau_{overlap}=1$ or $\tau_{overlap}=2$, there is a temporal interval relation $overlap(A, B)$ and a causal relation $A \rightarrow B$, but if $\tau_{overlap}=3$, there is no temporal interval relation because the condition of $(startTime(B) - startTime(A)) > 2$ must be satisfied. Fig. 6 shows that for τ_{during} , when its value increases, the number of causal relations decreases. This is also because $Constraint_{during}$ becomes stronger when the value of τ_{during} becomes larger. For example, for two interval-based events $(A, [10,16])$, and $(B, [12,15])$, if $\tau_{during}=1$ or $\tau_{during}=2$, a temporal interval relation $during(B, A)$ and a causal relation $A \rightarrow B$ occur, but if $\tau_{during}=3$, no temporal interval relation exists because the condition of $(startTime(B) - startTime(A)) > 2$ is not satisfied.

From the result of experiments (see Fig. 4, 5, and 6), we can find that the *DFCR-CRc* method discovers much more causal relations than *DFCR-FIRc*. This is because of the two following facts.

Fact 1: The number of different causal relations is less than or equal to the number of different interval relations.

Fact 2: The number of frequent causal relations discovered by *DFCR-CRc* is greater than or equal to the number of causal relations discovered by *DFCR-FIRc*.

Assume that $\#P$ denotes the number of relations P . For example, $\#before(A, B)$ denotes the number of temporal interval relations $before(A, B)$. *Fact 1* exists because several different temporal interval relations can be mapped into one causal relation. For example, two temporal interval relations $before(A, B)$ and $overlap(A, B)$ are both represented by one causal relation $A \rightarrow B$. In *DFIR-CRc*, even though $before(A, B)$ and $overlap(A, B)$ are both infrequent interval relations, $\#(A \rightarrow B)$ can be frequent because $(\#(A \rightarrow B)_{before} + \#(A \rightarrow B)_{overlap})$ can be greater than or equal to the given $MinSupportCount$. Therefore, *Fact 2* is always true.

It is better to discover frequent causal relations from temporal interval relations that could be infrequent than to discover frequent causal relations from frequent temporal interval relations. That is, *DFCR-CRc* can find frequent causal relations from temporal interval relations which might be infrequent. Therefore, the frequent causal relations discovered by *DFIR-CRc* can represent more important knowledge than those discovered by *DFIR-FIRc*. Parameters τ_{before} , $\tau_{overlap}$, and τ_{during} are properly adjusted by domain experts. For example, a patient can have symptom B after symptom A disappears, and another patient might have symptom B during the lifetime of symptom A . In this case, it is reasonable to infer that A is a cause symptom for B . Our proposed *DFCR-CRc* method discovers more valuable knowledge than *DFCR-FIR* or *DFCR-FIRc*.

VI. Conclusion and Future Works

We proposed a method that discovers temporal interval relations and causal relations from time-point-based events. To discover qualitative information from the event set, we found that a sequence of persistent events of the same type needs to be summarized into one interval event. It may be difficult to discover the temporal relations directly from the time-point-based events. Thus, we summarized the sequences of time-point-based events of the same type into interval-based events. From the interval-based events, temporal interval relations and causal relations were discovered. From the temporal interval relations, we can find effective causal relations if some parameters such as τ_{before} , $\tau_{overlap}$, τ_{during} and $MinSupport$ are properly selected by domain experts. We showed three methods (*DFCR-FIR*, *DFCR-FIRc*, and *DFCR-CRc*) used for discovering causal relations from temporal interval relations. Through the experiments, we found that the *DFCR-CRc* method discovers more effective and qualitative information than *DFCR-FIRc* and *DFCR-FIR*. We also proposed a new evaluation measurement that can be used

for ordering the major causes of an event.

In future work, based on the history data of patients, we intend to develop a method to predict time-points or time intervals at which an event is likely to occur. It is also important to discover temporal and causal relations among events (symptoms).

References

- [1] S. Laxman, P.S. Sastry and K.P. Unnikrishnan, "Discovering frequent episodes and learning hidden markov models: A formal connection," *IEEE Trans, Knowledge and Data Engineering*, vol. 17, no. 11, pp. 1505-1517, 2005
- [2] S. Laxman and P.S. Sastry, "A survey of temporal data mining," *Sadhana*, vol. 31, no. 2, pp. 173-198, 2006
- [3] S. Laxman, P.S. Sastry and K.P. Unnikrishnan, "Discovering frequent generalized episodes when events persist for different durations," *IEEE Trans, Knowledge and Data Engineering*, vol. 19, no.9, pp. 1188-1201, 2007
- [4] H.Y. Yun, D.S. Ha, B.H. Hwang and K.H. Ryu, "Mining association rules on significant rare data using relative support," *Journal of Systems and Software*, vol. 67, no. 3, pp. 181-191, 2003
- [5] C.H. Lee, M.S. Chen and C.R. Lin, "Progressive partition miner: an efficient algorithm for mining general temporal association rules," *IEEE Trans, Knowledge and Data Engineering*, vol. 15, no. 4, pp. 1004-1017, 2003
- [6] K.Y. Huang and C.H. Chang, "SMCA: a general model for mining asynchronous periodic patterns in temporal databases," *IEEE Trans, Knowledge and Data Engineering*, vol. 17, no. 6, pp. 774-785, 2005
- [7] S.Y. Wu and Y.L. Chen, "Mining nonambiguous temporal patterns for interval-based events," *IEEE Trans, Knowledge and Data Engineering*, vol. 19, no. 6, pp. 742-758, 2007
- [8] J.S. Yoo and S. Shekhar, "Similarity-profiled temporal association mining," *IEEE Trans, Knowledge and Data Engineering*, vol. 21, no.8, pp. 1147-1161, 2009
- [9] H. Jin, J. Chen, H. He, C. Kelman, D. McAullay and C.M. O'Keefe, "Signaling potential adverse drug reactions from administrative health databases," *IEEE Trans, Knowledge and Data Engineering*, vol. 22, no. 6, pp. 839-853, 2010
- [10] Y. Mohammad and T. Nishida, "Mining causal relationships in multidimensional time series," *Smart information and knowledge management*, Springer Berlin Heidelberg, pp. 309-338, 2010
- [11] J. Li, T.D. Le, L. Liu, J. Liu, Z. Jin and B. Sun, B. "Mining causal association rules," *Proc. 2013 IEEE 13th International Conf. Data Mining Workshops (ICDMW)*, pp. 114-123, Dec. 2013
- [12] S. Bleisch, M. Duckham, A. Galton, P. Laube and J. Lyon, J. "Mining candidate causal relationships in movement patterns," *International Journal of Geographical Information Science*, vol. 28, no. 2, pp. 363-382, 2014
- [13] H.D. Kim, M. Castellanos, M. Hsu, C. Zhai, T. Rietz and D. Diermeier, "Mining causal topics in text data: iterative topic modeling with time series feedback," *Proc. 22nd ACM international conf. information & knowledge management*, pp. 885-890, Oct. 2013
- [14] Y. Ji, H. Ying, J. Tran, P. Dews, A. Mansour and R.M. Massanari, "A method for mining infrequent causal associations and its application in finding adverse drug reaction signal pairs," *IEEE Trans, Knowledge and Data Engineering*, vol. 25, no. 4, pp. 721-733, 2013
- [15] I. Gurumurthy and R. Jayamma, "A Technique for Mining Uncommon Causal Relations and Its Submission in Finding Adverse Drug Reaction Indication Pairs," *International Journal of Computer Science and Mobile Applications*, vol. 2, no. 11, pp. 55-61, 2014
- [16] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal and M.C. Hsu, "Mining sequential patterns by pattern-growth: The prefixspan approach," *IEEE Trans, Knowledge and Data Engineering*, vol. 16, no. 11, pp. 1424-1440, 2004
- [17] R. Srikant and R. Agrawal, *Mining sequential patterns: Generalizations and performance improvements*, Springer Berlin Heidelberg, pp. 1-17, 1996
- [18] M.N. Garofalakis, R. Rastogi and K. Shim, "SPIRIT: Sequential pattern mining with regular expression constraints," *Proc. VLDB*, vol. 99, pp. 7-10, Sep. 1999
- [19] H. Mannila, H. Toivonen and A.I. Verkamo, "Discovery of frequent episodes in event sequences," *Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 259-289, 1997
- [20] Y.L. Chen and S.Y. Wu, "Mining temporal patterns from sequence database of interval-based events," *Fuzzy Systems and Knowledge Discovery*, Springer Berlin Heidelberg, pp. 586-595, 2006
- [21] Y.P. Huang, L.J. Kao and F.E. Sandnes, "A prefix tree-based model for mining association rules from quantitative temporal data," *Proc. 2005 IEEE International Conf. on Systems, Man and Cybernetics*, vol. 1, pp. 158-163, Oct. 2005
- [22] Y.J. Lee, J.W. Lee, D.J. Chai, B.H. Hwang and K.H. Ryu, "Mining temporal interval relational rules from temporal data," *Journal of Systems and Software*, vol. 82, no. 1, pp. 155-167, 2009
- [23] X. Wu, C. Zhang and S. Zhang, "Efficient mining of both

positive and negative association rules,” *ACM Trans. Information Systems*, vol. 22, no. 3, pp. 381-405, 2004

- [24] J.F. Allen, Maintaining knowledge about temporal intervals. *Communications of the ACM*, vol. 26, no. 11, pp. 832-843, 1983
- [25] Giang-Truong Nguyen, Van-Quyet Nguyen, Sinh-Ngoc Nguyen, Kyungback Kim, Efficient Association Rule Mining based SON Algorithm for a Bigdata Platform. *Journal of Digital Contents Society*, Vol.18, No.8, pp.1593-1601, Dec. 2017



Buhyun Hwang

1980: Korea Advanced Institute of Science and Technology (KAIST) (MS Degree).
1994: Korea Advanced Institute of Science and Technology (KAIST) (Ph. D. Degree).

2015.2~2016.2: visiting professor at the University of Virginia, USA

1981~2018 현재 : a professor at the School of Electronics and Computer Engineering, Chonnam National University, Korea

※Research Interests : data mining, stream data processing, and transaction processing

Jaemin Kim



2009: School of Electronics and Computer Engineering, Chonnam National University, South Korea (MS Degree)
2013: School of Electronics and Computer Engineering, Chonnam National University, South Korea (Ph.D. Degree)

2012~2018 현재: Electronics & Telecommunications Research Institute, Korea

※Research Interests : data mining, sensor stream data processing

Jaehyung Park



1993: Korea Advanced Institute of Science and Technology (KAIST) (MS Degree).
1997: Korea Advanced Institute of Science and Technology (KAIST) (Ph. D. Degree).

1997~1998: Center for AI Research , KAIST

1998~2002 : ETRI, Korea

2002~20018 현재: a professor at the School of Electronics and Computer Engineering, Chonnam National University, Korea

※Research Interests : internet routing, network security, stream data processing