http://dx.doi.org/10.3745/KTSDE.2013.2.8.565

Efficient Sequence Pattern Mining Technique for the Removal of Ambiguity in the Interval Patterns Mining

Hwan Kim† · Pilsun Choi† · Daein Kim†† · Buhyun Hwang†††

ABSTRACT

Previous researches on mining sequential patterns mainly focused on discovering patterns from the point-based event. Interval events with a time interval occur in the real world that have the start and end point. Existing interval pattern mining methods that discover relationships among interval events based on the Allen operators have some problems. These are that interval patterns having three or more interval events can be interpreted as several meanings. In this paper, we propose the I_TPrefixSpan algorithm, which is an efficient sequence pattern mining technique for removing ambiguity in the Interval Patterns Mining. The proposed algorithm generates event sequences that have no ambiguity. Therefore, the size of generated candidate set can be minimized by searching sequential pattern mining entries that exist only in the event sequence. The performance evaluation shows that the proposed method is more efficient than existing methods.

Keywords : Data Mining, Temporal Pattern, Sequential Patterns, Interval-based Events

인터벌 패턴 마이닝에서 모호성 제거를 위한 효율적인 순차 패턴 마이닝 기법

김 환† · 최 필 선† · 김 대 인†† · 황 부 현†††

요 약

기존의 순차 패턴 마이닝 기법은 주로 시점 기반 이벤트를 중심으로 연구되었다. 그러나 실생활에는 시작 시점과 종료 시점과 같은 시간 간격을 갖는 인터벌 이벤트가 많이 발생한다. Allen 연산자를 기반으로 두 인터벌 이벤트 사이의 인터벌 패턴을 탐사하는 기존의 기법은 세 개 이상의 인터벌 이벤트 사이에서 인터벌 패턴이 여러 의미로 해석될 수 있는 문제점을 가지고 있다. 이 논문은 인터벌 패턴 탐사에서 모호성 제거를 위한 효율적인 순차 탐색 마이닝 기법인 I_TPrefixSpan 알고리즘을 제안한다. 제안하는 기법은 인터벌 이벤트에 대한 이벤트 시퀀스를 생성함으로써 모호성을 제거하고 이벤트 시퀀스에 존재하는 항목만을 대상으로 순차 탐색함으로써 후보 집합 생성을 최소화 할 수 있다. 성능 평가를 통하여 제안하는 방법이 기존의 방법에 비하여 보다 효율적임을 보인다.

키워드 : 데이터 마이닝, 시간 패턴, 순차 패턴, 인터벌 이벤트

1. 서 론

데이터 마이닝은 대량의 데이터에서 숨겨진 패턴과 관계를 찾아내어 정보를 추출하는 과정이다. 이러한 정보는 시장 분석, 의사 결정, 침입 감지 등 다양한 분야에서 유용하게 사용된다[1, 2, 3]. 순차 패턴 마이닝은 시간 속성을 갖는 이벤트들의 순차적인 발생 관계를 탐사하는 방법으로 이벤트 사이의 인과 관계를 도출할 수 있다. 순서를 갖는 이벤트 사이에서 빈발하게 발생하는 순차 패턴을 탐사하는 기법은 [4]을 시작으로 많은 연구가 이루어졌다. 후속 연구로 기존의 Apriori 기법을 기반으로 하는 순차 패턴 마이닝의 비효율적인 부분을 개선한 PrefixSpan 알고리즘[5]이 제안되었다. 그 이후로 순차 패턴 마이닝 기법에 대한 다양한 후속 연구들이 발표되었다[6, 7, 8, 9, 10].

기존의 순차 패턴 마이닝 기법은 이벤트의 발생 시점과 같이 특정 시점 정보를 기반으로 하는 이벤트에 초점을 맞추고 있다. 실세계에서 발생하는 이벤트들은 특정 시점 뿐 아니라 이벤트가 발생하여 지속되고 종료하는 시점까지의 정보를 갖는 인터벌 이벤트가 많이 발생한다. 이러한 인터

※ 본 연구는 교육과학기술부와 한국연구재단의 지역혁신인력양성사업으로 수행된 연구결과임.
† 준 회 원 : 전남대학교 전자컴퓨터공학부 석사과정
†† 정 회 원 : 전남대학교 전자컴퓨터공학부 시간강사
††† 종신회원 : 전남대학교 전자컴퓨터공학부 교수
논문접수 : 2013년 3월 27일
수 정 일 : 1차 2013년 5월 28일
심사완료 : 2013년 5월 28일
* Corresponding Author : Buhyun Hwang(bhhwang@jnu.ac.kr)

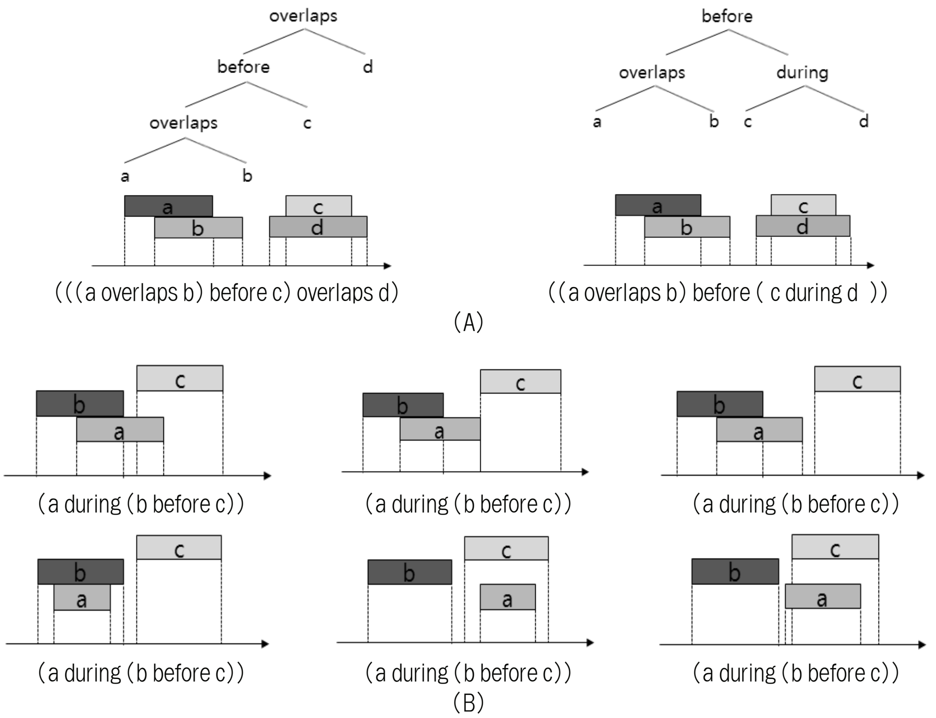


Fig. 1. Ambiguous Interval Patterns

벌 이벤트는 질병의 경과, 주식의 변동, 교통 정보 등 실생활에서 많이 찾아볼 수 있다. 이에 대한 관심으로 관련 연구도 활발하게 이루어지고 있다[11, 12, 13, 14].

Allen[13]은 인터벌 이벤트 사이에 존재하는 관계를 표현하기 위한 다음 13개의 관계 연산자를 제안하였다. “before”, “after”, “during”, “contain”, “meet”, “met by”, “overlap”, “overlapped by”, “start”, “started by”, finish“, ”finished by“, “equal”. Allen 연산자는 두 개의 인터벌 이벤트 사이에 존재하는 관계를 표현하는데 효과적이지만 세 개 이상의 인터벌 이벤트 사이에 존재하는 관계를 표현할 때 두 가지 이상으로 해석 가능한 문제점이 있다. 이러한 모호함을 해결하기 위해 [14]에서는 TPrefixSpan 알고리즘을 제안하였다. 그러나 TPrefixSpan 알고리즘은 후보 패턴 집합을 생성하는 과정에서 인터벌 이벤트 사이에서 생성 가능한 모든 경우의 수를 만들기 때문에 이벤트 시퀀스에 존재하지 않는 후보 패턴까지 생성하는 문제점을 가지고 있다.

이 논문에서는 인터벌 패턴 마이닝에서 모호성을 제거를 위한 효율적인 순차 패턴 마이닝 기법을 제안한다. 2절에서는 관련 연구와 문제점을 기술하고 3절에서는 기본 개념 및 I-TPrefixSpan 알고리즘을 기술한다. 4절은 성능 평가를 통하여 제안하는 알고리즘의 성능을 분석하고 마지막으로 5절은 결론 및 향후 연구에 대하여 기술한다.

2. 관련 연구

Kam과 Fu의 [12]에서는 Allen 연산자[13]를 기반으로 인터벌 이벤트들 사이에서 인과관계와 같은 유용한 정보를 탐사하기 기법을 제안하였다. Allen 연산자를 기반으로 셋 이상의 인터벌 이벤트 사이의 관계를 표현할 경우 다음과 같은 문제점이 있다. 첫째는 인터벌 이벤트간의 관계가 두 가지 이상의 인터벌 패턴으로 표현될 수 있다. 예를 들면 Fig. 1(A)에서 주어진 인터벌 이벤트 간의 관계에 대해서 (((a overlaps b) before c) overlaps d)와 ((a overlaps b) before (c during d)의 두 가지 인터벌 패턴으로 표현 할 수 있다. 둘째는 하나의 인터벌 패턴이 둘 이상의 인터벌 이벤트 사이의 관계로 해석 가능하다. Fig. 1(B)의 경우 주어진 인터벌 패턴 (a during (b before c))에 대해서 인터벌 이벤트가 서로 다른 6개의 관계로 해석될 수 있음을 보인다. 이러한 모호함은 다음과 같은 문제점을 야기한다.

1. 인터벌 이벤트의 관계를 어떤 인터벌 패턴으로 표현해야 할지 명백하지 않다.
2. 하나의 인터벌 패턴이 여러 관계로 해석이 가능할 때 어떤 관계로 해석해야 할지 명백하지 않다.

앞에서 제시된 문제점을 해결하기 위해 [14]에서는 새로운 형식의 인터벌 패턴 표현법을 제안하고 이를 탐사하기 위해 기존의 순차 패턴 마이닝 기법인 PrefixSpan 알고리즘을 기반으로 하는 TPrefixSpan 알고리즘을 제안하였다. 그러나 TPrefixSpan 알고리즘은 후보 집합 생성에 있어 가능한 모든 조합을 후보 집합으로 생성하기 때문에 트랜잭션에 존재하지 않은 후보들까지 생성하는 문제점이 있다. 이렇게 생성된 후보 집합은 빈발 항목을 탐사하기 위한 과정에서 불필요하게 많은 시간을 소모한다. 따라서 기존의 인터벌 관계의 모호한 문제점을 해결하면서 효율적으로 인터벌 패턴을 탐사할 수 있는 마이닝 기법이 필요하다.

3. I_TPrefixSpan

3.1 기본개념

환자 Cid가 질병이 발생하여 질병의 발생 시점과 종료 시점에 대한 정보를 가지고 있을 때 이는 하나의 인터벌 이벤트로 표현할 수 있다. 또한 환자 Cid는 여러 질병이 동시에 나타날 수 있다. 발생한 질병들에 대한 인터벌 이벤트를 시점에 따라 나열하여 인터벌 패턴으로 표현하며 한 환자 Cid에 대한 인터벌 이벤트를 시점에 따라 나열하여 한 환자 Cid에 대한 이벤트 시퀀스로 표현한다.

Table 1. Interval Event D

Cid	Disease	Interval	Cid	Disease	Interval
1	a	[2, 4]	2	b	[4, 7]
1	b	[3, 6]	2	c	[7, 10]
1	c	[6, 9]	3	a	[12, 15]
1	d	[7, 12]	3	b	[13, 16]
1	b	[13, 15]	3	d	[17, 19]
2	a	[3, 5]	3	b	[22, 26]

1) 인터벌 이벤트, Ei[ei+, ei-]

인터벌 이벤트 Ei는 이벤트 시작 시점 e_i^+과 이벤트 종료 시점 e_i^- 를 갖으며 이는 이벤트 E_i가 시점 e_i^+ 부터 e_i^- 까지 지속적으로 발생했음을 의미한다. Table 1에서 환자 Cid 1의 질병 a는 시점 2에서 발생하여 시점 4에서 종료였으며 그 동안 지속적으로 발생했음을 의미한다. 인터벌 이벤트 u와 v의 시점(시작 시점 또는 종료 시점)이 존재할 때 u가 v보다 먼저 일어났을 경우 u < v로 표기하며 동시에 발생했을 경우 u = v로 표기한다.

2) 인터벌 패턴

인터벌 패턴(IP : Interval Pattern)은 하나 이상의 인터벌 이벤트에 대해서 시작 시점과 종료 시점에 따라 정렬하여 나열하는 것을 의미한다. 하나의 인터벌 이벤트 E_i는 $e_i^+ < e_i^-$ 와 같이 표현하며 두 개 이상의 인터벌 이벤트는 각 인터벌 이벤트의 시작 시점과 종료 시점을 비교하여 순차적으로 나타내며 같은 시점에 발생한 인터벌 이벤트 시점의 경우 알파벳순으로 정렬한다. Fig. 2의 세 인터벌 이벤트에 대한 인터벌 패턴은 b+ < a+ < b− = c+ < a− < c− 와 같다.

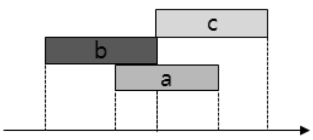


Fig. 2. Interval Relation among three Interval events

3) 이벤트 시퀀스

이벤트 시퀀스(ES : Event Sequence)는 한 환자의 Cid에 대한 인터벌 이벤트들을 시작 시점과 종료 시점에 따라 정렬하여 나열한 것을 의미한다. Table 1에서 환자 Cid 1에 대한 이벤트 시퀀스는 a+ < b+ < a− < b− = c+ < d+ < c− < d− < b+ < b− 이며 각 환자 Cid에 대한 이벤트 시퀀스는 Table 2와 같다.

Table 2. Sequential Database SD with respect to D

Cid	Event Sequences
1	a+ < b+ < a− < b− = c+ < d+ < c− < d− < b+ < b−
2	a+ < b+ < a− < b− = c+ < c−
3	a+ < b+ < a− < b− < d+ < d− < b+ < b−

4) 이벤트 시퀀스에 대한 인터벌 패턴 포함 관계

이벤트 시퀀스 ES와 인터벌 패턴 IP에 대해서 다음의 조건을 만족하면 IP는 ES에 포함 관계이다. 1. 이벤트 시퀀스의 인터벌 이벤트 E_i에 대해서 인터벌 패턴의 인터벌 이벤트 E_j가 모두 존재함. 2. 인터벌 패턴이 이벤트 시퀀스에 모두 존재함.

ES : a+ < b+ < a− < b− = c+ < c−, IP_1 : b+ < b− = c+ < c−, IP_2 : b+ < b− < c+ < c− 일 때 IP_1와 IP_2의 인터벌 이벤트가 이벤트 시퀀스 ES에 모두 존재하기 때문에 조건 1은 IP1, IP2 모두 만족한다. 조건 2에서 IP1은 ES에 존재하기 때문이 포함 관계이지만 IP2에서 인터벌 이벤트 b의 종료 시점과 인터벌 이벤트 c의 시작 시점의 관계가 b− < c+ 이지만 ES에서는 b− = c+ 이므로 서로 다른 인터벌 패턴을 갖고 있기 때문에 IP_2는 ES에 포함 관계가 아니다.

5) 지지도

시퀀스 데이터베이스 SD에 대한 인터벌 패턴 IP의 지지도는 다음과 같다.

$$\mathrm{Support}(\mathrm{IP},\ SD) = \frac{IP\text{를 포함하는 이벤트시퀀스 수}}{\text{전체 이벤트시퀀스 수}}$$

3.2 알고리즘

I-TPrefixSpan 알고리즘은 순차 패턴 탐사를 위해 제안된 PrefixSpan 알고리즘의 아이디어를 기반으로 제안한다. PrefixSpan 알고리즘의 경우 시점 기반 이벤트를 다룬다. 두 개의 시점 기반 이벤트 사이에 존재하는 시간 관계는 “before”, “equal”, “after” 세 가지 경우만 존재한다. 또한 빈발한 패턴의 길이 l에 대해서 길이 l + 1의 후보 패턴을 생성도 단순하다. 반면에 I-TPrefixSpan의 경우 인터벌 데이터를 다루기 때문에 두 개의 인터벌 이벤트 사이의 관계는 Allen 연산자를 인용할 경우 13개의 경우가 존재하고 후보 패턴을 생성함에 있어서 기존의 시점 기반 이벤트를 다룰 경우에 비해 다양한 경우의 수를 고려해야 한다. 따라서 인터벌 이벤트를 다루기 위해서는 PrefixSpan 알고리즘에 대한 변형이 필요하다.

1) Temporal Prefix

이벤트 시퀀스 ES와 이벤트 패턴 IP에 대해서 인터벌 패턴 IP가 이벤트 시퀀스 ES에 포함 관계에 있을 때 IP는 ES의 Temporal Prefix 이다.

ES : $a^{+} < b^{+} < a^{-} < b^{-} = c^{+} < d^{+} < c^{-} < d^{-} < b^{+} < b^{-}$

IP_1 : $a^{+} < b^{+} < a^{-} < b^{-} = c^{+} < d^{+} < c^{-} < d^{-}$

IP_2 : $a^{+} < b^{+} < a^{-} < b^{-} = c^{+} < d^{+} = c^{-} < d^{-}$ 에서 IP_1는 ES에 대해 포함 관계에 있으므로 IP_1는 ES의 Temporal Prefix이다. 반면에 IP_2는 ES에 대해서 인터벌 이벤트가 모두 존재하지만 인터벌 이벤트 d의 시작 시점 d^{+}와 인터벌 이벤트 c의 종료 시점 c^{-}의 인터벌 패턴이 ES에서는 $d^{+} < c^{-}$ 이고 IP_2에서는 $d^{+} = c^{-}$ 이므로 IP_2는 ES에 포함 관계에 있지 않기 때문에 Temporal Prefix가 아니다.

2) Projection

이벤트 시퀀스 ES에 대해서 인터벌 패턴 IP에 대한 Projection ES′는 IP가 ES에 포함 관계에 있을 경우, ES에서 IP의 인터벌 이벤트 중 마지막 이벤트 시작 시점부터 ES의 마지막 이벤트 종료 시점 사이에서 인터벌 이벤트의 시작 시점과 종료 시점의 짝이 맞지 않은 인터벌 이벤트가 존재하는 경우 해당 인터벌 이벤트를 삭제한 이벤트 시퀀스 ES의 나열로 나타낸다.

ES : $a^{+} < a^{-} < b^{+} = c^{+} < b^{-} < e^{+} < d^{+} < c^{-} < d^{-} < e^{-}$

ES′ : $a^{+} < a^{-} < c^{+} < e^{+} < d^{+} < c^{-} < d^{-} < e^{-}$

IP : $a^{+} < a^{-} < c^{+} < c^{-}$

주어진 이벤트 시퀀스 ES에서 인터벌 패턴 IP의 마지막 이벤트 시작 시점인 c^{+}과 ES의 마지막 이벤트 종료 시점인 e^{-} 사이에서 짝이 맞지 않은 이벤트는 b^{-}이다. ES에서 짝이 맞지 않은 인터벌 이벤트 b^{+}, b^{-}를 삭제하고 이벤트 시퀀스를 나열하면 $a^{+} < a^{-} < c^{+} < e^{+} < d^{+} < c^{-} < d^{-} < e^{-}$ 이고 이는 IP에 대한 ES의 Projection ES′이다.

3) Temporal Postfix

이벤트 시퀀스 ES와 인터벌 패턴 IP에 대한 Temporal Postfix는 Projection ES′에서 IP의 마지막 이벤트 시작 시점부터 그 뒤의 이벤트 시퀀스로 표현한다.

ES′ : $a^{+} < a^{-} < c^{+} < e^{+} < d^{+} < c^{-} < d^{-} < e^{-}$

IP : $a^{+} < a^{-} < c^{+} < c^{-}$

IP에 대한 ES의 Postfix는 IP의 마지막 이벤트 시작 시점인 c^{+}를 시작으로 하는 ES′의 이벤트 시퀀스 나열로 나타내고 이는 $c^{+} < e^{+} < d^{+} < c^{-} < d^{-} < e^{-}$ 이다.

4) 알고리즘

Algorithms I_TPrefixSpan(α, l, $SD|\alpha$)

Parameters : α : 인터벌 패턴, l : 인터벌 패턴 길이, $SD|\alpha$: α-projected database, SD : 시퀀스 데이터베이스.

Method :

1. l = 0 이면, SD에서 빈발한 길이 1 항목 인터벌 패턴(L_1)을 찾는다. 빈발하지 않은 1 항목 인터벌 패턴은 SD에서 지운다. L_1에 속한 b를 α에 접합 시킨 인터벌 패턴을 α'로 표기한다. 빈발한 인터벌 패턴 $F_1|$<> = L_1을 생성한다.
2. $l > 0$ 이면, $SD|\alpha'$에서 빈발한 1항목 $L_1|\alpha'$를 찾는다.
$F_{l+1}|\alpha$ = GenFrequent($L_1|\alpha$, α, $SD|\alpha$) 호출하여 α를 prefix로 하는 $l+1$ 항목의 빈발한 인터벌 패턴 a′의 집합($F_{l+1}|\alpha$)을 생성한다.
3. $F_{l+1}|\alpha$ 의 항목들인 α'의 projected database를 생성하고, $|SD|\alpha'|$ > 최소 지지도를 만족하면 다시 I_TPrefixSpan(α', $l+1$, $SD|\alpha'$)를 호출한다.

최소 지지도를 50%로 정하고 I_TprefixSpan 알고리즘을 Table 2에 적용하면 l = 0 인 경우 SD에서 빈발한 1 항목 인터벌 패턴 L_1 = {($a^{+} < a^{-}$), ($b^{+} < b^{-}$), ($c^{+} < c^{-}$), ($d^{+} < d^{-}$)}를 얻는다. 이에 대한 projected database는 Table 3과 같다.

Table 3. Projected Database for L1

prefix	Projected database
$a^{+} < a^{-}$	$a^{+} < b^{+} < a^{-} < b^{-} = c^{+} < d^{+} < c^{-} < d^{-} < b^{+} < b^{-}$ $a^{+} < b^{+} < a^{-} < b^{-} = c^{+} < c^{-}$ $a^{+} < b^{+} < a^{-} < b^{-} < d^{+} < d^{-} < b^{+} < b^{-}$
$b^{+} < b^{-}$	$b^{+} < b^{-} = c^{+} < d^{+} < c^{-} < d^{-} < b^{+} < b^{-}$ $b^{+} < b^{-} = c^{+} < c^{-}$ $b^{+} < b^{-} < d^{+} < d^{-} < b^{+} < b^{-}$
$c^{+} < c^{-}$	$c^{+} < d^{+} < c^{-} < d^{-} < b^{+} < b^{-}$
$d^{+} < d^{-}$	$d^{+} < d^{-} < b^{+} < b^{-}$ $d^{+} < d^{-} < b^{+} < b^{-}$

($a^{+} < a^{-}$)에 대한 projected database에서 빈발한 1 항목은 ($b^{+} < b^{-}$), ($c^{+} < c^{-}$), ($d^{+} < b^{-}$)이고 ($a^{+} < a^{-}$)를 prefix로 갖는 길이 2인 후보 인터벌 패턴을 생성하면 ($a^{+} < b^{+} < a^{-} < b^{-}$), ($a^{+} < a^{-} < b^{+} < b^{-}$), ($a^{+} < a^{-} < d^{+} < d^{-}$), ($a^{+} < a^{-} < b^{+} < b^{-}$)이다. 생성된 4개의 인터벌 패턴은 모두 $|C_{l+1}|a|$ > 최소 지지도를 만족하므로 빈발한 인터벌 패턴이고 이에 대한 각각의 projected database를 생성하고 위의 알고리즘을 반복하여 빈발한 인터벌 패턴을 생성한다. 모든 과정을 종료하고 출력하는 결과는 아래와 같다.

($a^{+} < a^{-}$), ($b^{+} < b^{-}$), ($c^{+} < c^{-}$), ($d^{+} < d^{-}$),
($a^{+} < b^{+} < a^{-} < b^{-}$), ($a^{+} < a^{-} < d^{+} < d^{-}$),
($a^{+} < a^{-} < c^{+} < c^{-}$), ($a^{+} < a^{-} < b^{+} < b^{-}$),
($a^{+} < b^{+} < a^{-} < b^{-} < b^{+} < b^{-}$),
($a^{+} < b^{+} < a^{-} < b^{-} < d^{+} < d^{-}$),
($a^{+} < b^{+} < a^{-} < b^{-} = c^{+} < c^{-}$),
($a^{+} < b^{+} < a^{-} < b^{-} < d^{+} < d^{-} < b^{+} < b^{-}$),
($a^{+} < a^{-} < d^{+} < d^{-} < b^{+} < b^{-}$),
($b^{+} < b^{-} < b^{+} < b^{-}$),
($b^{+} < b^{-} = c^{+} < c^{-}$),
($b^{+} < b^{-} < d^{+} < d^{-}$),
($b^{+} < b^{-} < d^{+} < d^{-} < b^{+} < b^{-}$),
($d^{+} < d^{-} < b^{+} < b^{-}$)

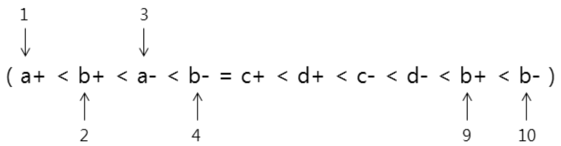
GenFrequent($L_1|\alpha$, α, $SD|\alpha$)

Parameters : $L_1|\alpha$: $SD|\alpha$ 에서 빈발한 1항목 인터벌 이벤트, α : 인터벌 패턴, l : 인터벌 패턴 길이, $SD|\alpha$: α-projected database.

Method :

1. $SD|\alpha$의 각 트랜잭션에서 α의 시퀀스 위치와 $L_1|\alpha$의 항목이 존재하면 시퀀스 위치를 찾는다.
2. α와 존재하는 $L_1|a$ 항목의 시퀀스 위치를 비교하여 각 트랜잭션에 존재하는 후보 집합($C_{l+1}|\alpha$)을 생성 하고 Hash table에 각 후보 집합 별 빈도수를 체크한다.
3. $SD|\alpha$를 모두 탐사한 후 후보 집합 중 $|C_{l+1}|\alpha|$ > 최소 지지도를 만족하는 항목 $F_{l+1}|\alpha$를 출력한다.

(a+ < a-)를 Prefix로 하는 길이 2 항목 후보 패턴을 생성할 경우 SD|a의 트랜잭션에서 (a+ < a-)와 빈발한 1항목 (b+ < b-), (c+ < c-), (d+ < b-)의 시퀀스 위치를 찾는다. Table 3의 (a+ < a-)에 대한 Projected Database의 첫 트랜잭션에서 (a+ < a-)과 (b+ < b-)를 갖는 후보 패턴을 생성하기 위해 시퀀스 위치를 찾으면 (a+ < a-)은 (1, 3) 이고 (b+ < b-)은 (2, 4)와 (9, 10) 이다. 시퀀스 위치를 순서대로 정렬하면 (1, 2, 3, 4)와 (1, 3, 9, 10)이 되고 이를 바탕으로 후보 패턴 (a+ < b + < a- < b-)와 (a+ < a- < b+ < b-)을 생성한다.



기존의 TPrefixSpan에서는 l + 1빈발한 인터벌 패턴을 탐사하기 위해 생성 가능한 모든 경우의 후보 패턴을 생성한다. 예를 들어 빈발한 1항목인 인터벌 이벤트 (a+ < a-)과 (a+ < a-)에 대한 projected database에서 빈발한 1항목 인터벌 이벤트 (b+ < b-)에 대해서 후보 패턴을 생성할 경우 (a+ < a-)에서 삽입 가능한 위치를 찾고 그 위치에 (b+ < b-)를 조합하면 (a+ < b+ < a- < b-), (a+ = b+ < a- < b-), (a+ < b+ < b- < a-), (a+ < b+ < b- = a-), (a+ < a- < b+ < b-), (a+ < a- = b+ < b-) 와 같은 많은 후보 패턴을 생성한다. 생성된 후보 패턴은 지지도를 체크하고 최소 지지도를 만족하면 빈발한 패턴으로 출력한다. 기존에 제안된 TPrefixSpan 알고리즘의 경우 빈발한 인터벌 이벤트의 수가 많거나 후보 패턴 항목의 길이가 길어질수록 더 많은 경우의 수가 발생하게 되고 생성되는 후보 패턴의 수가 많아지는 문제점이 있다. 또한 트랜잭션에 존재하지 않은 패턴도 생성하기 때문에 지지도를 체크하는데 있어 많은 수행시간이 필요하다. 반면에 제안하는 I_TPrefixSpan의 경우 후보 패턴을 트랜잭션에 존재하는 패턴만을 생성하기 때문에 생성하는 후보 패턴 수를 줄이고 수행시간도 단축시킬 수 있었다.

4. 성능 평가

이 절에서는 빈발한 인터벌 패턴을 탐사하기 위해 기존에 제안된 TPrefixSpan 알고리즘과 본 논문에서 제안한 I_TprefixSpan 알고리즘을 수행 시간을 비교한다. 실험 환경은 Intel Core i3 3.30GHz, RAM 8.00GB이며 Visual Studio 2010에서 C# 언어로 작성하였다.

Table 4. Data Parameter

Data Set	|T|	|EN|	|IN|
T10KEN5IN3	10000	5	3
T10KEN10IN3	10000	10	3
T50KEN10IN3	50000	10	3

Parameter	Meaning
|T|	Number of Transactions
|EN|	Number of event types
|IN|	Number of average interval events in Transaction

인터벌 데이터 마이닝 기법에 적합한 데이터 셋을 생성하는 일반화된 방법이 존재하지 않으므로 [4]의 생성 기법을 기반으로 두고 데이터를 생성하였다. 데이터 생성에 사용한 매개변수는 Table 4와 같다.

TPrefixSpan 알고리즘과 I_TPrefixSpan 알고리즘을 데이터 셋 T10KEN5IN3와 T10KEN10IN3, T50KEN10IN3에 적용한 결과는 각각 Fig. 3과 Fig. 4, Fig. 5와 같다. 최소 지지도가 낮으면 빈발한 이벤트 수가 많이 검출 되므로 다음 l + 1 항목에 대한 후보 패턴을 만드는 경우의 수가 많아지고 두 알고리즘의 후보 패턴 생성 방법이 다르기 때문에 수행시간에 많은 차이를 보였다. 최소 지지도를 높여가면 빈발한 이벤트 수가 상대적으로 적게 검출되므로 두 알고리즘의 수행시간 차이도 점점 줄어들었음을 알 수 있다. 또한 Fig. 3과 비교하여 Fig. 4에서 이벤트 타입 종류의 수가 증가하였으나 트랜잭션에 존재하는 인터벌 이벤트의 수가 적으면 수행시간이 줄어들었음을 알 수 있다. 이는 트랜잭션 내에 존재하는 평균 인터벌 이벤트의 수가 고정되어 있으므로 이벤트 타입의 종류의 수가 하면 상대적으로 최소 지지도를 만족하는 이벤트의 수가 적어지고 따라서 후보 패턴을 생성하는 이벤트의 수가 줄어들게 되어 수행시간이 줄어든다. Fig. 5의 경우 트랜잭션 수가 늘었으므로 수행시간이 전반적으로 증가했음을 보인다. 실험결과 모두 기존의 TPrefix Span 알고리즘과 동일한 빈발 인터벌 이벤트 결과를 출력하였으며 제안하는 I_TPrefixSpan 알고리즘의 경우 최소 지지도가 낮았을 경우 수행시간이 상대적으로 많이 줄었음을

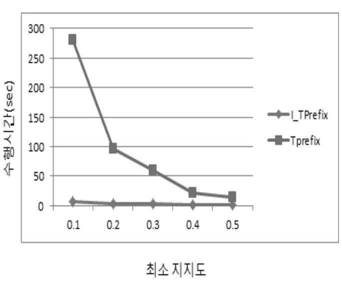


Fig. 3. T10KEN5IN3

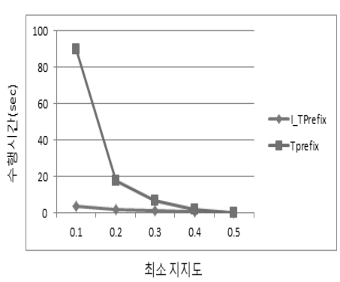


Fig. 4. T10KEN10IN3

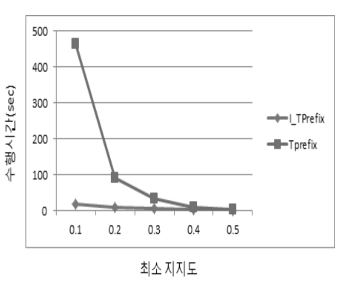


Fig. 5. T50KEN10IN3

볼 수 있다. 이는 후보 패턴을 생성하고 빈발한 인터벌 패턴을 출력하는데 있어 기존 알고리즘 보다 제안하는 I_TPrefixSpan 알고리즘이 더 효율적임을 보인다.

5. 결론 및 향후 연구

세 개 이상의 인터벌 이벤트 사이에서 Allen's 연산자로 인터벌 패턴을 표현 했을 때 갖는 모호성을 제거하고 빈발한 인터벌 패턴을 탐사하기 위해 [14]에서는 시점 기반 이벤트를 대상으로 하는 PrefixSpan 알고리즘을 인터벌 이벤트에 적용한 TPrefixSpan 알고리즘을 제안하였다. 제안된 알고리즘은 후보 집합을 생성함에 있어 모든 경우의 수를 고려하기 때문에 생성하는 후보 집합의 수 뿐 아니라 탐사를 위한 많은 수행 시간이 필요하다. 이 논문에서는 기존의 모호성을 제거하고 후보 집합을 효율적으로 생성함으로써 수행시간을 단축시킨 I_TPrefixSpan 알고리즘을 제안하였다. 성능 평가에서 빈발한 인터벌 이벤트의 수가 많을수록 기존의 알고리즘에 비해 수행 시간을 상대적으로 많이 단축시킬 수 있음을 보였다. 향후 연구 방향으로는 특정 응용분야에서 인터벌 패턴의 모호함으로 나타날 수 있는 문제점을 파악하고 실 데이터에 적용하여 명백한 인터벌 패턴 탐사 규칙의 유효성을 검증하고자 한다.

참 고 문 헌

[1] M.-S. Chen, J. Han, and P.S. Yu, "Data Mining: An Overview from a Database Perspective," IEEE Trans. Knowledge and Data Eng., Vol.8, No.6, pp.866-883, Dec., 1996.

[2] W.J. Frawley, G. Piatetsky-Shapiro, and C.J. Matheus, Knowledge Discovery in Database: An Overview. AAAI/MIT Press, 1991.

[3] J. Han and M. Kamber, Data Mining: Concepts and Techniques. Academic Press, 2001.

[4] R. Srikant, R. Agrawal, "Mining sequential patterns : generalizations and performance improvements", Proceedings of International conference, on Extendng Database Technology, Avignon, France. Springer-Verlag. 1996.

[5] Jian Pei, Jiawei Han, B. Mortazavi-Asi, J. Wang, H. Pinto, Q. Chen, U. Dayal, M. Hsu, "Mining Sequential Patterns by Pattern-Growth", The PrefixSpan Approach, IEEE Transactions on Knowledge and Data Engineering, Vol.16, 2004. 11.

[6] J. Allen, "Maintaining Knowledge about Temporal Intervals", Comm. of the ACM, Vol.26(11), 1983. 11.

[7] Y. L. Chen, S. Y. Wu, "Mining temporal patterns from sequence database of interval-based events", Int. Conference on Fuzzy Systems and Knowledge Discovery, Xian, China, 2006.

[8] Minos N. Garofalakis, Rajeev Rastogi, Kyuseok Shim, "SPRIT : Sequential Pattern Mining with Regular Expression Constraints", Proceedings of the 25th VLDB Conference, Edinburgh, Scotland, pp.223-234, 1999.

[9] K. Y. Huang, C. H. Chang, "SMCA : A General Model for Mining Asynchronous Periodic Patterns in Temporal Databases", IEEE Transactions on Knowledge and Data Engineering, Vol.17, No.6, 2005. 6.

[10] Y. P. Huang, L. J. Kao, F, E. Sandnes, "A Prefix Tree-Based Model for Mining Association Rules from Quantitative Temporal Data". IEEE International Conference on Systems, Man, and Cybernetics, Vol.1, pp.158-163, 2005. 10.

[11] Y. J. Lee, J. W. Lee, D. J. Chai, B. H. Hwang, K. H. Ryu, "Mining temporal interval relational rules from temporal data", The Journal of Systems and Software, 82(2009), 155-167.

[12] P.S. Kam and A.W.C. Fu, "Discovering Temporal Patterns for Interval-Based Events", Proc. Second Int'l Conf. Data Warehousing and Knowledge Discovery (DaWaK '00), 2000.

[13] J.F. Allen, "Maintaining Knowledge about Temporal Intervals", Comm. ACM, Vol.26, No.11, pp.832-843, 1983.

[14] Shin-Yi Wu, Yen-Liang Chen, "Mining Nonambiguous Temporal Patterns for Interval-Based Events", IEEE Transactions on Knowledge and Data Engineering, Vol.19, No.6, June, 2007.

김 환

e-mail : sentual7@naver.com

2011년 전남대학교 전자컴퓨터공학부(학사)

2012년～현 재 전남대학교 전자컴퓨터 공학부 석사과정

관심분야 : Data Mining, Stream Data, Algorithm

최 필 선

e-mail : pilddong@nate.com

2009년 전남대학교 전자컴퓨터공학부(학사)

2011년～현 재 전남대학교 전자컴퓨터 공학부 석사과정

관심분야 : Data Mining, Stream Data, Algorithm

김 대 인

e-mail : dikim@jnu.ac.kr

1998년 전남대학교 전산통계학과(이학석사)

2006년 전남대학교 전산통계학과(이학박사)

2004년～현 재 전남대학교 전자컴퓨터 공학부 시간강사

관심분야 : Stream Data, Data Mining, Digital Contents

황 부 현

e-mail : bhhwang@jnu.ac.kr

1978년 숭실대학교 전산통계학과(학사)

1980년 한국과학기술원 전산학과(공학석사)

1994년 한국과학기술원 전산학과(공학박사)

1980년～현 재 전남대학교 전자컴퓨터 공학부 교수

관심분야 : Stream Data Mining, Distrubuted System, Distributed Database