가상 트랜잭션을 이용한 시계열 데이터의 데이터 마이닝

김 민 수[†] · 김 철 환[††] · 김 응 모[†††]

요 약

대용량의 데이터들로부터 사용자가 원하는 데이터를 찾기 위하여 많은 데이터 마이닝 기술들이 연구되어 실제 응용프로그램에서 많이 적용되고 있다. 이러한 데이터 마이닝 기술들은 시계열 데이터를 이용하는 경우보다 트랜잭션 데이터를 이용하여 유용한 정보를 찾는 경우에 초점이 맞춰져 있다. 본 논문에서는 시계열 데이터를 트랜잭션 데이터로 변환하는 접근방법을 소개한다. 가상 트랜잭션은 서로 상대적으로 근접한 시간에 발생하는 이벤트의 집합이라고 정의하며, 가상 트랜잭션 생성기는 가상 트랜잭션을 생성시 시간윈도우와 이벤트 윈도우 방법을 사용한다. 본 논문의 접근 방법을 사용하여 기존의 트랜잭션 데이터를 이용하는 많은 데이터 마이닝 알고리즘들을 수정 없이 시계열 데이터에 적용하여 유용한 정보를 찾을 수 있다.

Data Mining Time Series Data With Virtual Transaction

Min-Soo Kim[†] · Chul-Hwan Kim[††] · Ung-Mo Kim[†††]

ABSTRACT

There has been much research on data mining techniques for applying more advanced applications. However, most of these techniques has focused on transaction data rather than time series data. In this paper, we introduce a approach to convert time series data into virtual transaction data for more useful data mining applications. A virtual transaction is defined to be a collection of events that occur relatively close to each other. A virtual transaction generator uses time window or event window methods. Our approach based on time series data can be used with most conventional transaction algorithms without further modification.

키워드 : 데이터 마이닝(data mining), 시계열 데이터(time series data), 가상 트랜잭션(virtual transaction), 이벤트 데이터(event data), 패턴(pattern), 트랜잭션(transaction)

1. 서 론

데이터 마이닝은 지식탐사 과정에서 핵심적인 엔진 역할을 담당하며, 사용 목적에 따라 알고리즘을 선택하여 사용한다. 데이터 마이닝은 다양한 유형을 가진 대량의 데이터들로부터 데이터 상호간의 관련성, 데이터에 함축적으로 들어 있는 지식이나 패턴 및 각 도메인에서 관심을 가지는 정보를 추출하는 일련의 과정을 말한다.

데이터 마이닝 기술들은 실제로 많이 적용되고 있다. 백화점에서 물건을 진열할 시 고객의 이동을 줄여 판매를 늘리는데 활용되기도 하고 고객의 구매 패턴을 찾아내어 소비자가 살 상품을 미리 예측하고 쿠폰을 발행하여 판매를 늘리는데도 사용된다. 보험회사에서는 고객이 다른 보험회사로 옮기는 것을 방지하거나 고객의 위험도에 따라 보험료를 차등 적용하는데 사용되며 신용 카드 회사에서는 불법의 신용카드 사용을 막는데 사용되기도 한다. 그 외에 전자상거래, 의학, 주식시장 예측에도 적용되기도 한다. 또한 최근에는 DNA 지도를 만드는데도 적용되어 일익을 담당하고 있다. 많은 분야에서 적용되고 있는 데이터 마이닝 알고리즘은 보다 정확한 정보를 찾기 위하여 여러 가지 마이닝 기술이 함께 적용되는 경우가 많다.

이러한 데이터 마이닝은 크게 연관규칙, 순차패턴, 클래시피케이션, 클러스터링, 기계학습 등과 같이 분류되고 그 분류 안에서 많은 데이터 마이닝 기술들이 현재에도 활발이 연구가 진행되고 있다. 이러한 데이터 마이닝 알고리즘들은 대부분 트랜잭션 데이터로부터 숨어 있는 정보를 찾아낸다. 그러나 실 세계에는 시스템이벤트와 같이 트랜잭션으로 구성되어 있지 않은 시계열 이벤트 데이터(서버에서의 시스템로그, 전화통신회사에서의 알람)가 많이 존재한다.

본 논문의 주요 접근방법은 시간정보를 담고 있는 이벤트 중심의 시계열 이벤트 데이터로부터 가상의 트랜잭션을 생성한다면, 기존의 트랜잭션 데이터를 사용하는 많은 알고

※ 본 연구는 한국과학재단 목적기초연구 (과제번호 : R01-2000 00250)지원으로 수행되었음.
† 준 회 원 : 데이터게이트 인터네셔널 보안연구소
†† 정 회 원 : 성균관대학교 정보통신공학부 교수
††† 종신회원 : 성균관대학교 전기전자및컴퓨터공학부
논문접수 : 2001년 12월 31일, 심사완료 : 2002년 3월 11일

리즘의 데이터 입력으로 사용할 수 있다는 것이다. 즉 시계열 데이터를 잘 가공하여 트랜잭션 데이터화한다면 시계열 데이터에 대한 새로운 알고리즘 개발 없이 많은 정보를 기존의 알고리즘을 사용하여 얻을 수 있다. 본 논문에서는 시계열 데이터로부터 가상 트랜잭션을 생성할 때 시간윈도우 기법과 이벤트윈도우 기법을 이용한다. 이렇게 생성된 가상 트랜잭션을 연관규칙 알고리즘들[1, 2, 14, 17]을 적용하여 유용한 패턴을 찾는다. 그리고 가상 트랜잭션들의 발생시간을 트랜잭션에 속한 이벤트들의 평균발생시간으로 사용함으로써 패턴들의 주기를 발견할 수 있는 주기패턴 알고리즘[4, 7, 8, 12, 13]과 패턴들의 순서를 발견할 수 있는 순차패턴 알고리즘[5, 9-11]을 적용할 수 있다.

2. 관련 연구

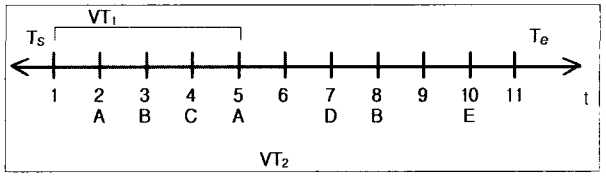
시간 기반의 데이터에 대한 연구가 많이 진행되었고 현재도 계속 연구되고 있다. 이런 연구들은 시계열 데이터에 대한 연구와 시간 기반의 트랜잭션 데이터에 대한 연구로 진행되고 있다. 시계열 데이터는 이벤트발생시간과 이벤트 및 기타 정보를 한 개의 레코드로 구성되어 있으며 그 항목들 중 시간과 이벤트가 관심 항목이다. 반면에, 트랜잭션 데이터는 트랜잭션 발생시간과 항목들의 집합으로 구성되어 있고 시간과 항목집합이 관심 필드이다. 시계열 데이터에 관련된 연구와 트랜잭션 데이터에 관련된 연구는 같은 결과를 얻기 위하여 각기 다른 알고리즘을 사용한다. 그 이유는 위에서 언급했듯이 데이터 소스가 다르기 때문이다. 데이터 마이닝에서는 트랜잭션 데이터를 데이터소스로 이용한 많은 알고리즘들이 존재한다.

시계열 데이터에 대한 대표적 연구로서는 통신회사에서 발생하는 시간적인 순서를 가지는 네트워크 이벤트 로그로부터 각 이벤트들 사이의 연관성을 찾는 빈발 에피소드 기법이 있다. 이 알고리즘에서는 시간정보를 담고있는 전체 이벤트집합이 주어질 때, 이벤트 집합 간격 [t1, t2]는 시간 t1부터 t2까지 발생한 이벤트 집합을 말하며 간격의 크기는 t2 - t1으로 정의된다. 이렇게 정의된 이벤트의 집합을 에피소드라 한다. 이러한 에피소드의 발생회수를 저장하여 알고리즘의 설정 값인 min-fr(최소지지도) 보다 큰 것들만 빈발 에피소드로 결정해낸다. 이렇게 결정된 빈발 에피소드는 자주 함께 발생하는 이벤트의 모음으로 정의하며 이런 빈발 에피소드를 사용하여 이벤트간의 연관 규칙을 찾는다. 기본적인 알고리즘은 첫째, 주어진 시간 이벤트 집합으로부터 두번 이상 발생한 모든 에피소드를 찾는다. 둘째, 주어진 최소지지도를 사용하여 빈발 에피소드를 찾는다[3]. 이렇게 찾아진 빈발에피소드는 이벤트들간의 순차패턴과 일치한다. 이 알고리즘은 현재 침입탐지시스템의 비정상행위 탐지모델에도 적용되고 있다. 이벤트기반의 시계열 데이터에는 주기를 갖는 패턴도 중요하지만 주기에 어긋나는 피턴이 중요한 관심사항이 될 수 있다. 예를 들어 병원에서 어떤 환자에게 약을 1일 3회 투여하고 있다. 그런데 어느 날 환자의 상태가 악화되어 약을 좀 빨리 투여하였을 경우, 주기패턴 분석에 의하면 하루에 9시, 13시, 19시의 주기를 찾을 수 있다. 하지만 환자의 상태에 따라 빨리 투여 된 약의 경우는 분석이 안 된다. 이런 경우의 이벤트 패턴을 찾는 방법이 제안되었다[4]. 트랜잭션 데이터 관련 연구는 연관규칙을 찾는 기법[1, 2, 12]은 항목 집합으로 구성된 데이터 트랜잭션들로부터 각 항목간의 연관성을 반영하는 규칙을 찾는 기법이다. 지지도와 신뢰도를 설정 값으로 트랜잭션 내에 존재하는 항목 집합들간의 연관규칙을 찾는다. 지지도란 전체 트랜잭션에서 해당 연관규칙이 차지하는 확률이다. 예를 들어 X, Y -> Z (X이고 Y이면 Z이다) 는 X, Y, Z간의 연관규칙이 있다. 이 연관규칙의 지지도가 10%라면 전체 트랜잭션 중에서 이 규칙을 따르는 트랜잭션이 10%를 차지한다는 것이다. 그리고 신뢰도가 50% 라는 것은 X, Y를 포함한 트랜잭션 중 50%는 Z항목을 포함한다는 것이다. 트랜잭션 데이터들 속에서 순차패턴을 찾는 알고리즘[9-11]은 순서정보를 가지고 있는 트랜잭션 데이터 베이스로부터 트랜잭션 내의 데이터에 공통적으로 나타나는 순차적인 패턴을 찾는 기술이다. 고객의 트랜잭션을 담고 있는 대용량 데이터베이스를 입력 데이터로 한다. 여기서 트랜잭션은 고객 ID와 트랜잭션 시간정보를 담고 있어야 한다. 한 개의 트랜잭션에는 한번에 구매하는 아이템들이 담겨져 있다. 예를 들어 비디오 대여점에서 고객들은 '쉬리'를 빌려보는 사람들은 'JSA'를 빌려본다라는 순차패턴을 찾음으로써 '쉬리'를 본 고객들에게 'JSA'를 추천할 수 있다. 어떤 이벤트가 발생했을 때 얼마만큼의 시간이 지난 후에는 어떤 이벤트가 발생할 것인가를 예측하는 방법들이 많이 연구되고 있다. 이러한 예측은 주식시장에서 주가를 예측하는 곳에 사용되고 있다[8]. 대부분의 패턴을 찾는 알고리즘은 짧은 길이의 패턴을 찾는데 효과적이다. 그러나 실제 데이터베이스에는 긴 길이의 패턴이 많이 포함되어 있다. 긴 패턴을 효과적으로 찾을 수 없는 기존의 Apriori관련 알고리즘을 개선한 MAX-Miner 알고리즘을 고안했다[12].

3. 가상트랜잭션

본 논문의 궁극적인 목적은 시계열데이타를 트랜잭션화하여 트랜잭션을 데이터 소스로 사용하는 많은 알고리즘에 적용시킬 수 있도록 하는 것이다. 여기서는 가상 트랜잭션 개념을 소개한다. 가상 트랜잭션(Virtual Transaction)은 시계열데이타를 타임 윈도우 기법과 이벤트 윈도우 기법을 사용하여 만든 이벤트의 집합이다. 시계열데이터에는 시간 컬럼과 이벤트 컬럼이 존재한다. 이러한 시계열데이터를 시

간 축으로 나열한 후에 정해진 윈도우의 길이만큼 이벤트들을 집합으로 묶는다. 시계열데이터로부터 가상 트랜잭션 집합을 생성해 내는데 있어서 두 가지의 접근방법을 시도한다. 한가지는 고정타임 윈도우를 이용하는 방법이고 다른 한가지는 고정이벤트 윈도우를 이용하는 방법이다. 위의 두 가지 기법에 '겹침깊이(Overlapping Depth)'를 사용하여 가상 트랜잭션을 생성한다. 이 방법으로 만들어진 가상 트랜잭션 데이터 집합은 본 논문의 실험에서 연관규칙 알고리즘의 입력데이터로 사용하여 이벤트 패턴을 분석하는데 사용되었다. 다음 (그림 1)은 시간 축에서의 가상 트랜잭션을 설명한 그림이다.



(그림 1) 시간 축에서의 가상 트랜잭션

E : 이벤트유형의 집합 = { A, B, C , , , ⋯ }

Ε : (E, t) = 시간 t에 발생한 이벤트, 예)

(A, 2) : 시간 2에 발생한 이벤트 A

S : 발생한 이벤트(Ε)들의 집합 = S(Ts, Te)

$$= \sum_{Ts}^{Te} \epsilon = \sum_{Ts}^{Te} (E, t)$$

= { (E1, t1), (E2, t2), ⋯, (En, tn) }

여기서 Ei ∈ E, ti에서의 이벤트, i = 1, 2, 3, ⋯, n

Ts : start time of event sets

Te : end time of event sets

Ts ≤ ti ≤ Te

LengthTime(S) = Te - Ts

CountEvent(S) = # of Event in S

가상 트랜잭션 VT1 = (A, 2), (B, 3), (C, 4), (A, 5)

타임윈도우 : VT(E, 1, 5),

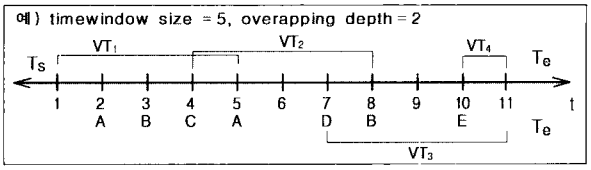
LengthTime : 4, *CountEvent* : 4

이벤트윈도우 : VT(E, 2, 5),

LengthTime : 3, *CountEvent* : 4

4. 타임윈도우와 이벤트윈도우

타임 윈도우 기법은 빈발 에피소드[3]를 생성하는 알고리즘에서 아이디어를 얻어왔다. 시계열데이터에서 관련 있는 이벤트는 비슷한 시간에 많이 발생한다. 이러한 속성을 이용하여 이벤트 시간 축에서 가상 트랜잭션을 만드는데 타임 윈도우 길이와 겹침 깊이를 사용한다. 얼마만큼의 시간 간격으로 가상 트랜잭션을 유지할 것인가를 결정하는 요인이 타임 윈도우 길이이다. 그리고 겹침 깊이는 인접한 가상 트랜잭션간 얼마만큼의 시간이 겹치게 가상 트랜잭션을 만들 것인가에 대한 요인이다.



(그림 2) 타임 윈도우

타임 윈도우에서 겹침 깊이는 시간 기준이다. (그림 2)는 타임 윈도우 길이가 5이고 겹침 깊이가 2인 경우의 생성되는 가상 트랜잭션을 표현한 것이다.

가상 트랜잭션 VT1 = VT(Ei, 1, 5)

= { (A, 2), (B, 3), (C, 4), (A, 5) }

가상 트랜잭션 VT2 = VT(Ei, 4, 8)

= { (C, 4), (A, 5), (D, 7),(B, 8) }

VT4는 {(E,10)}한 개로만 구성되므로 가상 트랜잭션으로 생성하지 않음

w = Time Window : $\sum_{i=ts}^{te}$ (Ei, ti), ts ≥ Ts , te ≤ Te

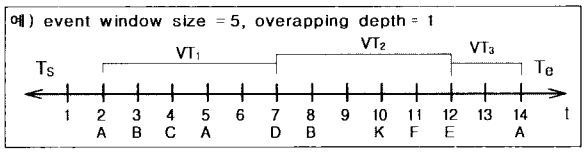
ℓ = LengthT(w) : ts - te : time window size

Overlapping Depth : 가상 트랜잭션간의 겹침 깊이.

타임 윈도우에서의 가상 트랜잭션 : VT(E, ts, te)

= {(Es, ts), (Es+1, ts+1), ⋯, (Ee, te) }

이벤트 윈도우 기법은 시계열 데이터에서 이벤트가 발생되는 시점이 넓은 시간 간격을 가지고 발생될 때 관련 있는 이벤트들은 시간적인 가까움보다는 발생 이벤트의 개수로서 가상 트랜잭션을 구성할 때 의미가 높을 수 있다. 이 경우는 이벤트가 드물게 발생하는 시계열 데이터에서 보다 의미가 있다. 이벤트 시간 축에서 가상 트랜잭션을 만드는데 이벤트 윈도우 길이와 겹침 깊이를 사용한다. 얼마만큼의 이벤트 개수 간격으로 가상 트랜잭션을 유지할 것인가를 결정하는 요인이 이벤트 윈도우의 길이이다. 그리고 겹침 깊이는 인접한 가상 트랜잭션간 얼마만큼 이벤트 개수가 겹치게 가상 트랜잭션을 만들 것인가에 대한 요인이다. 이벤트 윈도우에서의 겹침 깊이는 발생한 이벤트 기준이다. 다음 (그림 3)은 이벤트 윈도우 길이가 5이고 겹침 깊이가 1인 경우에 생성되는 가상 트랜잭션을 표현한 것이다.



(그림 3) 이벤트 윈도우

가상 트랜잭션 VT1 = VT(Ei, 2, 7)

= {(A, 2), (B, 3), (C, 4), (A, 5), (D, 7)}

가상 트랜잭션 VT3 = VT(Ei, 12, 14)
= { (E, 12), (A, 14) }

w = Event Window : $\sum_{i=ts}^{te}$ (Ei, ti), ts ≥ Ts, te ≤ Te

ℓ = LengthE(w) : ts - te : time window size

c = CountE(w) : # of evnet in a time window

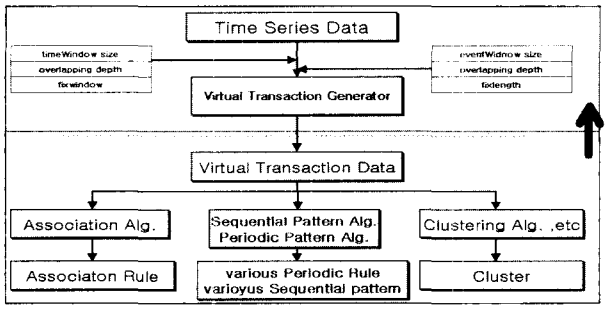
Overlapping Depth : 가상 트랜잭션간의 겹침 깊이

이벤트 윈도우에서의 가상 트랜잭션 :

VT(E, ts, te) = { (Es, ts), (Es+1, ts+1), ···, (Ee, te) }

5. 가상 트랜잭션의 활용

많은 데이터 마이닝 기술은 트랜잭션 데이터로부터 필요한 정보를 찾는다. 그래서 시계열 이벤트 데이터를 트랜잭션 데이터화 한다면 현존하는 트랜잭션 데이터로부터 숨겨진 정보를 찾는 알고리즘을 특별한 수정 없이 적용하기 때문에 많은 이점을 얻을 수 있다. 아래 (그림 4)는 시계열 이벤트 데이터를 가상 트랜잭션 데이터로 가공한 결과를 연관규칙 알고리즘, 순차패턴 알고리즘, 주기분석 알고리즘, 클러스터링 알고리즘에 적용하는 것을 보여주고 있다.



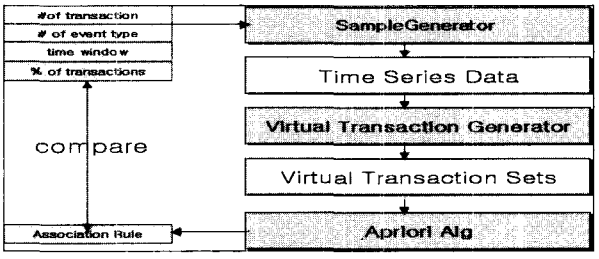
(그림 4) 가상 트랜잭션 활용 모델

연관 규칙 알고리즘을 사용하여 연관 규칙을 찾으면 이것은 시계열 이벤트 데이터에서 이벤트들간의 순차패턴을 찾는 것과 일치한다. 순차패턴 알고리즘을 적용하면 시계열데이터에서 패턴들의 패턴들을 찾는 것과 일치한다. 다양한 주기분석 알고리즘을 적용하면 패턴들의 주기를 찾을 수 있다. 또 클러스터링 알고리즘을 적용함으로써 패턴들을 그룹화하여 볼 수 있는 등의 많은 정보를 얻을 수 있다. 본 논문에서는 전기전자 컴퓨터 공학부 웹 서버의 로그파일로부터 가상 트랜잭션을 생성하고 이 생성된 웹 로그 가상 트랜잭션을 Apriori연관규칙 알고리즘을 적용하여 방문 되는 웹 페이지들의 연관성에 대해서 분석한다. 예를 들어 'A' 라는 페이지를 보고 'B'라는 페이지를 본 사람은 'D'라는 웹 페이지를 보는 지지도가 어느 정도 인지를 찾을 수 있다.

6. 실험 방법

본 논문의 실험에서는 두 가지의 시계열 데이터를 사용하였다. 한 가지는 샘플생성기로 생성한 샘플 데이터이[15] 고 다른 한 가지는 웹 서버의 로그 데이터이다. 샘플데이터는 가상 트랜잭션 생성기의 정확성 검증실험을 하는데 사용하였고, 웹 로그데이터는 가상 트랜잭션의 활용성을 확인하는 실험에 사용하였다.

가상 트랜잭션 생성기의 정확성 검증실험 방법은 구현된 샘플 시계열이벤트 데이터 생성기를 사용하여 입력 값으로 주어지는 트랜잭션의 지지도에 의해서 시계열데이터를 생성한다. 이렇게 생성된 시계열 데이터를 가상 트랜잭션 생성기에 입력 값으로 하여 가상 트랜잭션을 만들어 낸다. 다음 Apriori알고리즘을 적용[6]시켜 생성되는 연관규칙이 처음에 샘플 시계열이벤트 데이터 생성기에 입력된 트랜잭션과 같은 결과를 갖는지 또 같은 지지도를 갖는 지를 확인 함으로써, 가상 트랜잭션 생성기가 올바른 알고리즘으로 만들어졌는지 확인한다. 가상 트랜잭션 생성기의 정확성 검증 실험은 직접 구현한 시계열 데이터 생성기와 가상 트랜잭션 생성기가 정확히 작동되는지 확인하기 위한 실험이다. 아래 (그림 5)는 데이터 생성기의 신뢰성을 확인하기 위한 실험 모델이다. 샘플 생성기에 입력 값으로 이벤트 윈도우, 이벤트 윈도우 길이는 4, 그리고 'ABC' 트랜잭션을 20%의 지지도로 총 1000개의 트랜잭션을 생성하였을 경우에는 가상 트랜잭션 생성기에서 입력 값을 이벤트 윈도우, 윈도우 길이는 4로 하여 가상 트랜잭션을 생성한다. 이 경우에 가상 트랜잭션은 총 1000개가 생성되어야 한다. 그리고 이 생성된 가상 트랜잭션을 연관규칙 알고리즘에 적용시켰을 때 샘플 생성기에서 입력한 트랜잭션 'ABC' 의 지지도 20%와 같은 결과가 나와야 한다. 이 결과가 맞으면 실험에서 사용하는 가상 트랜잭션 생성기의 알고리즘이 올바르게 작동된다는 것을 확인할 수 있다.



(그림 5) 샘플생성기와 가상트랜잭션 생성기 신뢰성모델

가상 트랜잭션 활용성 검증 실험 방법은 웹 로그 데이터를 가상 트랜잭션 생성기를 사용하여 가상 트랜잭션을 생성한다, 생성된 가상 트랜잭션을 연관 규칙 알고리즘에 적용하여 각 요청된 페이지에 대한 항목집합을 얻는다. 이렇게 얻은 항목집합의 연관성을 확인한다. 실험에서 사용할 웹 로그 데이터는 웹 서버가 HTTP 요청 받은 시간, URL, 브라우저의 IP주소 등 여러 가지 정보를 시간에 따라 담고 있다. 하지만 웹 로그의 시간 값이 문자열 형식을 취하고 있으므로 실험에 필요한 데이터로 가공하는 절차를 거쳤다.

사용한 웹 로그 파일은 성균관대학교 전기전자 컴퓨터 공학부의 웹 서버의 로그 파일이다. 2000년 5월 7일 하루 동안 발생한 로그 파일을 사용하여 한 번은 이미지(GIF, JPG, JPEG, BMP, PCX)에 대한 요청을 분석 대상에서 제외시켜 실험했고 또 한번은 페이지(HTM, HTML)가 요청된 로그만으로 실험하였다. 실험은 인텔 펜티엄Ⅲ 450MHz, 196M 메모리의 윈도우98에서 VC++을 사용하였다.

7. 실험 결과 및 분석

<표 1>연관규칙에 대한 입력값들 중 실험 1-D의 경우 최소 아이템 개수를 4로 지정한 이유는 포함되는 트랜잭션을 제외시키기 위해서 이다. 예를 들어 'LAZO'라는 가상트랜잭션은 'LAZ', 'LZO'등과 같은 2차 가상 트랜잭션을 포함하고 있기 때문이다. 최소 지지도는 10%를 주었다. 샘플 생성시에 10% 이하의 트랜잭션은 생성하지 않았기 때문이다. 4가지 실험 모두 연관규칙 생성기에 의해서 연관규칙을 생성한 결과 샘플 생성기에서 입력된 트랜잭션과 지지도를 정확히 찾아냈다. <표 1>의 검증 결과표에 샘플데이터 생성시의 트랜잭션과 지지도 그리고 최종 연관규칙 생성기에서 찾아낸 연관규칙을 나타내고 있다. 실험 1의 경우에 입력된 트랜잭션 보다 많은 트랜잭션이 생성되었다. 이유는 'ABCK' 트랜잭션이 'ABC', 'ABK', 'BCK', 'ACK'를 포함하기 때문에 맞는 결과이다. 실험 1의 경우 연관규칙 생성기에서 minimal number를 4로 지정하면 결과로는 'XYZ'은 나오지 않고 'ABCK'만 나오게 된다.

〈표 1〉 실험 1 검증 결과

input \ output		실험 1-A	실험 1-B	실험 1-C	실험 1-D
		T1000_10.apts	E1000_10.apts	E10000_10.apts	E10000_10.apts
실험 1-A	ABCK(20) XYZ(30)	ABCK(20) ABC(20) ABK(20) XYZ(30)	·	·	·
실험 1-B	KJU(10) SGH(15)	·	KJU(10) SGH(15)	·	·
실험 1-C	BCK(10) LOP(20)	·	·	BCK(10) LOP(20)	·
실험 1-D	LAZO(30) SECU(10)	·	·	·	LAZO(30) SECU(10)

<표 1>의 4가지의 실험 1-A부터 1-D를 통해서 시계열 데이터 샘플 생성기와 가상 트랜잭션 생성기가 정확히 작동되는 것을 확인할 수 있었다.

웹 로그 실험 결과를 보면, 실험 4에서는 웹 서버의 로그에서 이미지 관련된 요청을 제외한 로그를 사용하였고 실험 5에서는 웹 페이지만을 요청한 로그를 사용하였다. 1일 동안의 로그 41,818건 중에서 이미지를 제외한 요청건수는 41,482건이었고 요청파일의 종류는 총 7,787가지였다. 그리고 웹 페이지만 요청한 경우의 요청 건수는 36,893건이었고 요청 페이지의 종류는 7,040가지였다. 로그 파일을 가상 트랜잭션 생성기에의 입력으로 사용하기 위하여 전 처리 프로그램에서 로그의 시간 문자열을 UTC시간으로 변경하였고, 요청한 페이지의 경로를 이벤트로 항목으로 변환하였다. 그리고 웹 서버의 로그는 시간으로 정렬되어 있지 않아 전 처리 프로그램에서 시간을 기준으로 올림차순으로 정렬하였다.

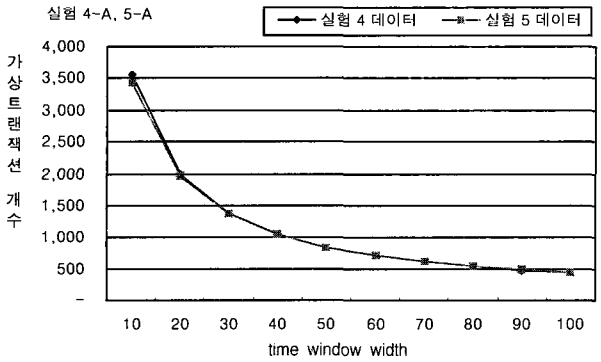
〈표 2〉 실험 4 결과

구분	트랜잭션 생성기			연관규칙 생성기	
				지지도(10%) 최소항목(3)	지지도(20%) 최소항목(3)
	Time/event	소요시간	# of VTS	항목집합 개수	항목집합 개수
실험 4-A	T : 10	451	3550	0	0
	T : 20	431	1989	5	0
	T : 30	424	1379	6	0
	T : 40	420	1051	6	0
	T : 50	418	844	6	1
	T : 60	415	713	6	1
	T : 70	414	611	16	5
	T : 80	412	537	86	5
	T : 90	413	478	174	5
	T : 100	412	431	177	5
실험 4-B	E : 10	655	4149	1	0
	E : 20	641	2075	5	0
	E : 30	636	1383	6	3
	E : 40	628	1038	19	5
	E : 50	627	830	165	5
	E : 60	637	692	271	5
	E : 70	626	593	884	6
	E : 80	623	519	1847	6
	E : 90	630	461	2620	28
	E : 100	647	415	3436	258

〈표 3〉 실험 5 결과

구분	트랜잭션 생성기			연관규칙 생성기	
				지지도(10%) 최소항목(3)	지지도(20%) 최소항목(3)
	Time/event	소요시간	# of VTS	항목집합 개수	항목집합 개수
실험 5-A	T : 10	359	3435	0	0
	T : 20	342	1956	5	0
	T : 30	335	1365	6	0
	T : 40	332	1043	6	1
	T : 50	329	843	6	1
	T : 60	327	712	6	5
	T : 70	326	609	11	5
	T : 80	326	534	106	5
	T : 90	324	480	119	5
	T : 100	324	432	126	5
실험 5-B	E : 10	511	3690	2	0
	E : 20	496	1845	6	0
	E : 30	492	1230	6	5
	E : 40	489	923	73	5
	E : 50	489	738	154	5
	E : 60	487	615	458	6
	E : 70	486	528	1114	6
	E : 80	487	462	1766	22
	E : 90	486	410	2689	91
	E : 100	486	369	4366	95

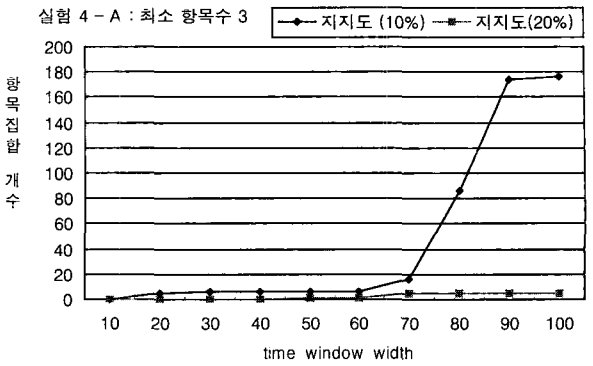
<표 2> 실험 4-A와 <표 3> 실험 5-A는 타임 윈도우 길이를 증가해 가면서 실험한 결과이고 <표 2> 4-B와 <표 3> 5-B는 이벤트 윈도우 길이를 증가해 가면서 실험한 결과이다. 위의 두 실험 결과를 보면 윈도우 길이 10일때 3,000에서 4,000개 가량의 가상 트랜잭션을 생성한다. 그런데 윈도우 길이 20인 경우를 보면 10인 경우의 반 정도의 가상 트랜잭션을 생성하는 것을 확인할 수 있다. 그리고 계속해서 윈도우 길이를 증가하게 되면 점점 적은 폭으로 가상 트랜잭션의 개수가 감소하는 것을 확인할 수 있다. <표 2> <표 3> 실험 결과를 보면, 연관규칙을 생성하는 부분에서는 최소 지지도를 3%와 5% 그리고 최소 항목(이벤트) 개수를 3개와 2개로 설정하였다. 최소 지지도를 10% 이상으로 설정하여 실험한 경우 연관 규칙 생성기에서 항목집합을 찾지 못했다. 현재 실험 데이터로 사용한 웹 서버에는 약 130,000개의 웹 페이지가 존재한다. 그래서 지지도가 높을 경우에 연관 규칙을 찾을 수가 없다. 즉 사용자들의 페이지를 방문하는 패턴을 찾기 어렵다는 것이다. 특히 가상 트랜잭션 생성시 윈도우 길이를 3에서 5까지의 짧은 길이로 설정하였기 때문에 생성된 가상 트랜잭션을 이용 연관 규칙을 찾을 확률이 작다.



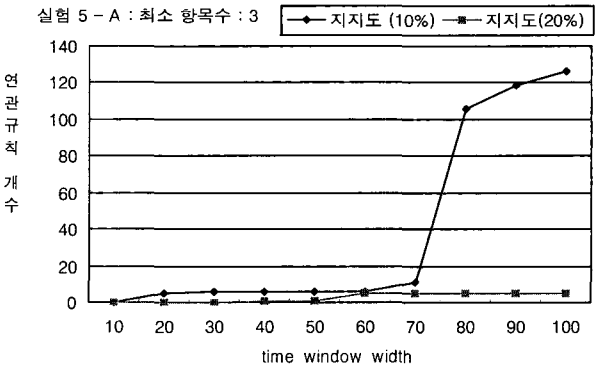
(그림 6) 타임윈도우 길이와 가상트랜잭션 개수

(그림 6)은 실험 4와 실험 5에서 타임 윈도우길이 증가에 따라 생성되는 가상 트랜잭션 개수를 그래프로 표현하였다. 생성되는 가상 트랜잭션의 개수가 타임 윈도우의 길이가 증가함에 따라 타임 윈도우 길이가 10에서 20으로 증가할 때 급격한 감소를 하고 타임 윈도우 길이 70부터는 완만한 감소를 하고 있다. 실험 4와 실험 5 모두 비슷한 그래프로 나타난 것은 웹 로그 데이터가 타임 윈도우 20 이하의 타임 윈도우에서 가장 많은 패턴을 가지고 있음을 알려주는 것이다. 여기서 가장 많은 패턴을 생성한다고 해서 높은 지지도의 연관 규칙을 가지고 있다는 것은 아니다.

(그림 7)(그림 8)는 실험 4와 실험 5에서 타임 윈도우 길이의 증가에 따라 발견된 항목집합의 개수를 그래프로 나타낸 것이다. (그림 6)에서 타임 윈도우 길이가 증가할 때 생성되는 가상 트랜잭션의 개수가 감소한 것과는 반대로,

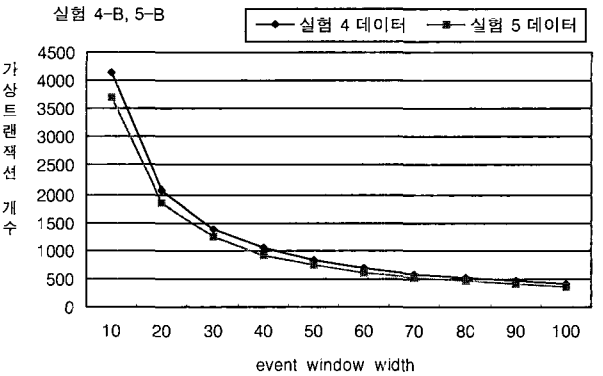


(그림 7) 실험 4의 타임윈도우 길이와 항목집합개수



(그림 8) 실험 5의 타임윈도우 길이와 항목집합개수

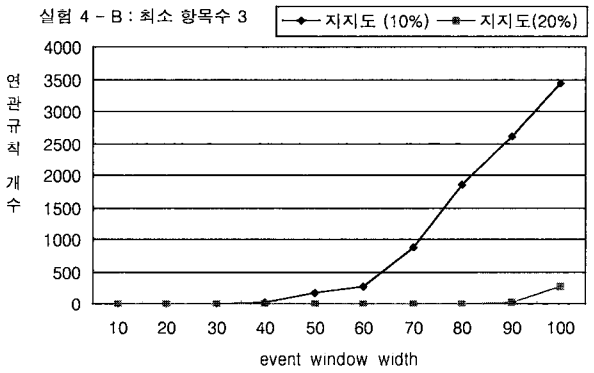
연관규칙 생성기에 의해 생성되는 연관규칙 개수는 지지도 10%의 경우 타임 윈도우 70부터 급격히 많이 생성됨을 확인할 수 있다. 이런 결과는 타임 윈도우 길이가 길수록 한 개의 가상 트랜잭션에 많은 이벤트가 들어가 있으므로 각각의 가상 트랜잭션간에 유사한 패턴을 많이 포함할 수 있다는 것을 설명해주는 것이다.



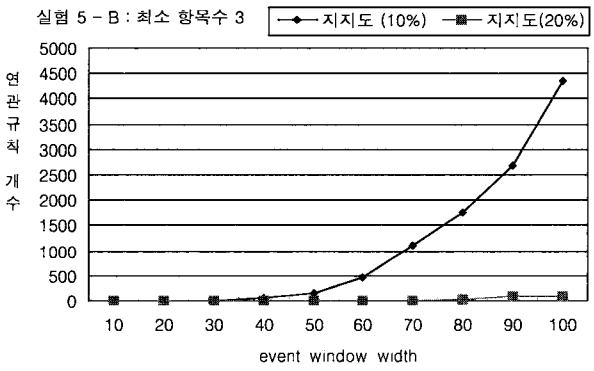
(그림 9) 이벤트 윈도우 길이와 가상 트랜잭션 개수

(그림 9)는 실험 4와 실험 5에서 이벤트 윈도우 증가에 따라 생성되는 가상 트랜잭션 개수를 그래프로 표현한 것이다. 생성되는 가상 트랜잭션의 개수는 타임 윈도우와 마찬가지로 이벤트 윈도우 길이가 10에서 20으로 증가할 때

급격히 감소하고 이벤트 윈도우 길이 60부터는 완만한 감소를 보이고 있다. (그림 6)의 타임 윈도우의 결과와 비교해 보면 더 많은 가상 트랜잭션을 생성하고 있다. 이는 웹 로그 데이터가 시간 길이 10내에 10개 미만의 로그가 존재함을 의미하는 것이다.



(그림 10) 실험 4의 이벤트윈도우 길이와 항목집합개수



(그림 11) 실험 5의 이벤트윈도우 길이와 항목집합개수

(그림 10)(그림 11)은 실험 4와 실험 5에서 이벤트 윈도우 길이의 증가에 따라 연관규칙 생성기에서 찾아진 항목집합 개수를 그래프로 나타낸 것이다. (그림 9)에서 이벤트 윈도우의 길이가 증가할 때 생성되는 가상 트랜잭션의 개수가 감소한 것과는 반대로, 연관 규칙 생성기에 의해서 생성되는 연관규칙 개수는 지지도 10%의 경우 이벤트 윈도우 길이 50부터 많이 생성됨을 확인할 수 있다. 이런 결과는 이벤트 윈도우 길이가 길수록 한 개의 가상 트랜잭션에 많은 이벤트가 들어가 있어 각각의 가상 트랜잭션간에 유사한 패턴을 많이 포함하고 있음을 설명해주는 것이다.

타임 윈도우를 이용한 실험 결과 (그림 7)(그림 8)와 비교해 볼 때 윈도우의 길이가 길어질수록 더 큰 개수의 연관규칙을 찾는 것을 확인할 수 있다. 이것은 이벤트 윈도우를 이용하여 가상 트랜잭션을 생성할 때, 타임 윈도우를 이용하여 가상 트랜잭션을 생성하는 것보다 더 많은 로그를 포함한다는 것을 알려주는 것이다. 웹 로그를 사용한 실험에서 찾아지는 항목집합들을 확인해 본 결과 웹 서버에서 가장 많이 요청되는 페이지임을 알 수 있었다. 이것은 본 논문의 접근 방법이 옳은 방법임을 말해주는 것이다.

8. 결 론

본 논문에서는 시계열 데이터에 대한 마이닝에 있어서 새로운 접근 방법으로 시계열 데이터를 마이닝 하는데 있어 기존의 트랜잭션 관련 알고리즘을 적용할 수 있도록 가상 트랜잭션 알고리즘을 제안하였다. 실험에서는 우선, 가상 트랜잭션 생성 알고리즘의 정확성을 확인하기 위한 실험을 하였다. 다음으로, 웹 로그데이터를 가상 트랜잭션 알고리즘을 사용하여 웹 페이지간의 연관 규칙을 찾는 실험을 하였다. 첫 번째 실험결과 가상 트랜잭션 생성기가 정확히 가상 트랜잭션을 생성하는 것을 확인 할 수 있었다. 그리고 두 번째 실험에서는 웹 로그 데이터를 가상 트랜잭션 생성기와 연관 규칙 알고리즘[1]을 사용하여 요청된 웹 페이지들의 연관된 항목 집합들을 찾아보았다. 그리고 윈도우의 길이를 증가해 가며 생성되는 가상 트랜잭션의 개수와 항목집합 개수의 의미를 분석하였으며, 이벤트 윈도우 기법과 타임 윈도우 기법의 결과를 비교해 보았다. 그 결과 가상 트랜잭션을 이용하여 웹 로그로부터 연관규칙이 잘 생성됨을 확인할 수 있었다. 그러나 본 논문에서는 가상 트랜잭션 생성 알고리즘의 고안 시 성능을 고려하지 않아 생성 시 많은 시간이 소요되었다. 차후 알고리즘의 개선에 관한 연구가 더욱 필요하다. 트랜잭션 데이터를 사용하는 알고리즘에 가상 트랜잭션 데이터를 적용하는 연구도 필요하며, 아울러 가상 트랜잭션을 적용할 알고리즘의 특성과 시계열 이벤트 데이터의 특성에 따른 보정항목의 연구가 필요하다.

참 고 문 헌

[1] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases. In Proc. of the ACM SIGMOD Conference on Management of Data, Washington, D.C., pp.207-216, May, 1993.

[2] Rakesh Agrawal and Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules. In Proc. of the 20th Int'l Conference on Very Large Databases, Santiago, Chile, September, 1994.

[3] H. Mannila, H. Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences. Data Mining and Knowledge Discovery, 1(3), 1997.

[4] Jiong Yang, Wei Wang, and Philip Yu, Mining asynchronous periodic patterns in time series data, Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD), pp.275-279. 2000.

[5] C. Bettini, X. Sean Wang, and S. Jajodia. Mining temporal relationships with multiple granularities in time sequences. Data Engineering Bulletin, 21 : pp.32-38, 1998.
[6] Http : //fuzzy.cs.uni-magdeburg.de/~borgelt/index.html.
[7] J. Han, G. Dong, and Y. Yin. Efficient mining of partial periodic patterns in time series database. In Proc. 1999 Int. Conf. Data Engineering (ICDE'99), Sydney, Australia, April, 1999.
[8] J. Han, W. Gong, and Y. Yin. Mining segment-wise periodic patterns in time-related databases. In Proc. 1998 Int'l Conf. on Knowledge Discovery and Data Mining (KDD'98), New York City, NY, August, 1998.
[9] Rakesh Agrawal and Ramakrishnan Srikant. Mining Sequential Patterns. In Proc. of the 11th Int'l Conference on Data Engineering, Taipei, Taiwan, March, 1995.
[10] Srikant, R., & Agrawal, R., Mining sequential patterns : Generalizations and performance improvements, Proc. of the Fifth Int'l Conference on Extending Database Technology (EDBT). Avignon, France. 1996.
[11] F. Masseglia, F. Cathala, and P. Poncelet. The PSP Approach for Mining Sequential Patterns. In Proceedings of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'98), LNAI, Nantes, France, Vol.1510, pp.176-184, September, 1998.
[12] R. J. Bayardo. Efficiently mining long patterns from databases. In Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data, Seattle, Washington, pp.85-93, June, 1998.
[13] Sheng Ma, Joseph L. Hellerstein. mining partially periodic event patterns, IEEE, 2001.
[14] Rakesh Agrawal and Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules. In Proc. of the 20th Int'l Conference on Very Large Databases, Santiago, Chile, September, 1994.
[15] B. Lent, A. Swami, and J. Widom. Clustering association rules. In Proc. 1997 Int. Conf. Data Engineering (ICDE'97), Birmingham, England, pp.220-231, April, 1997.
[16] S. Ramaswamy, S. Mahajan, and A. Silberschatz. On the discovery of interesting patterns in association rules. In Proc. 1998 Int. Conf. Very Large Data Bases, New York, NY, pp.368-379, August, 1998.
[17] Ramakrishnan Srikant and Rakesh Agrawal. Mining Generalized Association Rules. In Proc. of the 21st Int'l Conference on Very Large Databases, Zurich, Switzerland, September, 1995.

김 민 수

e-mail : lasarus@datagate.co.kr
1996년 성균관대학교 화학과 졸업(학사)
1996년~1997년 한라정보시스템
1997년~1999년 코다정보통신
2002년 성균관 대학교 전기전자컴퓨터공학부 졸업(공학석사)
2002년~현재 데이터게이트 인터내셔널 보안연구소
관심분야 : 데이터마이닝, 침입탐지시스템, 시스템 보안

김 철 환

e-mail : chkim@speed.skku.ac.kr
1982년 성균관대학교 전기공학과 학사
1990년 성균관대학교 전기공학과 박사
1990년~현재 성균관대학교 정보통신공학부 교수

김 응 모

e-mail : umkim@yurim.skku.ac.kr
1981년 성균관대학교 수학과 학사
1986년 Old Dominion University 전산학과 석사
1990년 Northwestern University 전산학과 박사
1997년~1998년 University of California, Irvine 전산학과 방문교수
1991년~현재 한국정보처리학회논문지 편집부 위원장
1990년~현재 성균관대학교 전기전자및컴퓨터공학부 교수
관심분야 : 데이타마이닝, Web데이타베이스, 객체지향DB, 트랜잭션관리