The Journal of the Convergence on Culture Technology (JCCT)
Vol. 8, No. 6, pp.647-653, November 30, 2022. pISSN 2384-0358, eISSN 2384-0366

http://dx.doi.org/10.17703/JCCT.2022.8.6.647
JCCT 2022-11-79

# 시간 단위 그룹핑을 이용한 빈발 아이템셋 마이닝

# Mining Frequent Itemsets using Time Unit Grouping

황정희*

**Jeong Hee Hwang***

**요 약** 데이터 마이닝은 데이터를 탐색하고 분석하여 데이터 사이의 관계나 패턴 등의 지식을 탐사하는 기법이다. 실세계에서 발생하는 데이터는 시간 속성을 포함한다. 시간 속성을 포함하는 데이터에서 유용한 지식을 찾아내기 위한 시간 데이터마이닝 연구는 미래를 예측할 수 있는 예측 판단에 효율적으로 활용될 수 있다. 본 논문은 데이터베이스를 일정한 시간 간격 단위로 구분하고, 시간 단위에서 빈발한 패턴 아이템셋을 발견하기 위한 시간 단위 그룹핑을 이용하는 알고리즘을 제안한다. 제안하는 알고리즘은 시간 단위에 포함된 트랜잭션과 아이템 정보를 매트릭스로 구성하고, 그룹핑을 통한 시간 단위에서의 빈발한 아이템셋을 발견한다. 성능평가의 실험 결과에서 수행시간은 기존의 알고리즘보다 1.2배 소요되지만, 2배 이상의 빈발 아이템셋이 탐사되었다.

**주요어** : 시간 데이터 마이닝, 데이터 마이닝, 시간 속성, 시간 단위, 빈발 아이템셋

**Abstract** Data mining is a technique that explores knowledge such as relationships and patterns between data by exploring and analyzing data. Data that occurs in the real world includes a temporal attribute. Temporal data mining research to find useful knowledge from data with temporal properties can be effectively utilized for predictive judgment that can predict the future. In this paper, we propose an algorithm using time-unit grouping to classify the database into regular time period units and discover frequent pattern itemsets in time units. The proposed algorithm organizes the transaction and items included in the time unit into a matrix, and discovers frequent items in the time unit through grouping. In the experimental results for the performance evaluation, it was found that the execution time was 1.2 times that of the existing algorithm, but more than twice the frequent pattern itemsets were discovered.

**Key words :** Temporal Data Mining, Data Mining, Temporal Property, Time Unit, Frequent Itemsets

## Ⅰ. 서 론

데이터 마이닝 기법 중 가장 많이 연구되고 있는 분야는 데이터 사이의 연관성을 찾는 연관규칙 탐사이다. 연관규칙 탐사를 통하여 데이터 항목들을 포함하는 서로 다른 그룹간에 존재하는 항목의 연관성을 탐사할 수 있다[1, 2]. 트랜잭션에 존재하는 항목들은 일정한 시간 간격으로 발생할 수 있다. 데이터베이스 전체에서는 빈발하지 않지만 어떤 특정 기간에는 빈발한 항목이 될 수 있다. 특정한 시간 간격 주기를 갖는 데이터에 대한 마이닝은 계절이나 시기적 특성을 고려할 수 있으므로 의미가 있다[3, 4]. 빈발 패턴(frequent pattern)은 데이터 집합에서 빈번하게 발생하는 항목들이다. 빈발 패턴을 발견하는 것은 데이터 사이의 연관성 그리고 우리가

*정회원, 남서울대학교 컴퓨터소프트웨어학과 (단독저자)
접수일: 2022년 9월 1일, 수정완료일: 2022년 9월 30일
게재확정일: 2022년 10월 20일

Received: September 1, 2022 / Revised: September 30, 2022
Accepted: October 20, 2022
*Corresponding Author: jhhwang@nsu.ac.kr
Dept. of Computer Software, Namseoul Univ, Korea

알지 못했던 데이터간의 흥미로운 관계를 탐사하는 기본적인 역할을 한다[5, 6].

순차패턴은 연관규칙에 시간 정보를 고려하여 순차적으로 빈발한 항목집합을 찾아낸다. 예를 들면, 순차패턴은 환자 이력, 구매 이력, 로그 이력 등 다양한 이력 데이터에 숨겨진 지식을 탐사한다. 여기서 이력은 이벤트의 시퀀스를 의미하고, 이벤트는 발생 시점과 종료 시점의 시간 속성을 갖는다. 타임 스탬프된 이벤트 시퀀스는 발생한 이벤트 식별자와 발생시각을 하나의 쌍으로 하는 순차적인 서열로 나타내고 이들 시퀀스 이벤트를 대상으로 이벤트들의 연관 관계와 패턴을 검색하는 연구들이 있다. 이들 연구는 시계열 분석, 고객의 구매패턴, 네트워크 침입 탐지와 같은 응용에 사용된다[7, 8]. 모든 데이터는 시간 정보를 수반하여 발생하므로 시간을 고려하는 데이터 마이닝은 실세계의 응용에서 중요하다. 일반적인 연관규칙에서 시간적인 변이가 추가된 규칙에 따라 사건이 발생된다면 순차패턴 규칙이라 한다[9]. 즉, 어떤 고객 그룹의 구매 이력에서 상품 S의 구매가 발생한 후 일정시간이 경과하고 상품 T의 구매가 이루어지는 시퀀스가 최소 지지도를 만족하면 빈발 시퀀스라고 한다. 축적된 이벤트 시퀀스에 대해 연관규칙을 탐사함으로써 과거의 데이터를 바탕으로 발생 가능한 미래의 사건을 예측할 수도 있다[7].

빈발 에피소드(frequent episode) 탐사[10]는 일련의 사건 시퀀스(event sequence) 데이터로부터 빈번하게 발생하는 에피소드를 찾는 순차패턴 기법이다. 에피소드는 이벤트 타입의 순서화된 사건 시퀀스를 의미하며, 시퀀스를 구성하는 사건은 서로 밀접하게 관련되어 있다. 이 기법은 임의의 윈도우 크기의 윈도우 집합에서 해당 에피소드의 발생 비율이 임계치를 만족하는 빈발 에피소드를 발견하는 것이다. 예를 들어, 발생한 사건 시퀀스 {a, b, f, h, a, c, d, u, a, c, d, g}에 대해 윈도우 크기를 4초, 최소 임계치를 0.5라 할 때, (a, c, d)는 이를 만족하는 빈발 에피소드에 속한다. 이러한 기법은 환자의 치료나 주식 시장의 예측과 같은 과학적인 요소가 필요한 분야 또는 염기 서열 및 유전체 분석 등에 사용된다.

시간 데이터 마이닝(temporal data mining)은 데이터 마이닝에 시간 정보를 접목하여 시간적 의미와 관계를 갖는 연관 지식을 탐사한다[11]. 시간 데이터 마이닝을 통해 질병 치료과정, 주가 변화, 교통 정보 등과 같은 시간과 관련된 데이터로부터 다양한 형태의 지식을 발견할 수 있다. 예를 들면 “Outerware를 사는 사람은 Hiking Boots를 산다”와 같은 연관규칙뿐만 아니라, “더운 여름 6월~8월에 아기 기저귀를 사는 남성 고객의 70%는 자신이 마실 맥주를 함께 산다“와 같은 시간 개념을 포함하는 연관규칙을 찾아낼 수 있다. 그리고 ”어떤 연령대의 A 구매패턴을 가진 고객은 우수 고객 부류에 속했으나 퇴직 연령대에 속하는 시점 이후에는 고객 부류에 변화가 있다“와 같이 시간의 흐름을 고려할 때 변화되는 사회적, 환경적 요인으로 인해 새로운 규칙이 발견될 수도 있다. 그러므로 시간 연관규칙 탐사 기법은 연관규칙 탐사, 분류, 특성화와 같은 기존의 데이터 마이닝 기법을 확장한 기법으로 시간과 관련된 인과관계의 연관규칙을 탐사한다[11]. 그리고 시간 연관규칙은 반복되는 연관규칙을 발견하기 위한 순환 연관 관계 탐사와 캘린더 형태로 표현된 시간 패턴에 대한 연관규칙을 발견하는 캘린더 연관 관계 탐사[12]를 포함한다. 시간 간격을 갖는 이벤트에 대한 시간 연관규칙 탐사 방법을 [7]에서 제안하였다. 한 번의 스캔으로 이벤트의 인터벌을 시작 시점과 종료 시점으로 표현하여 인터벌 이벤트로 요약하고, 요약된 인터벌 이벤트에서 시간 관계 연산자를 이용하여 연관규칙을 탐사한다. 기존의 연관규칙 연구에서는 이벤트에 대한 트랜잭션의 특정 시점만을 고려하였고, 지속적으로 이어지는 시간 그룹 단위에서의 빈발한 데이터의 분석은 이루어지지 않는다. 이벤트 발생의 인과관계에 대한 초점을 두지 않더라도 관심있는 일정 시간 단위에서 유용한 데이터를 발견하는 것은 중요한 의미가 있다. 따라서 본 논문에서는 시간 데이터베이스에서 트랜잭션의 일정 수를 포함하는 시간 간격 단위를 구분하고, 시간 단위에 포함된 트랜잭션들에서 빈발 아이템셋을 탐사하는 데이터 마이닝 기법을 제안한다. 제안 알고리즘에서는 먼저, 주어진 기본적인 시간 단위에서 빈발한 아이템셋을 발견한다. 그리고 연속적인 시간 간격 단위를 그룹핑하여 계층적인 구조가 형성되며, 그룹핑된 각 시간 단위에서 빈발 아이템셋을 발견한다. 우리가 제안하는 시간 단위 그룹핑의 구조는 [13]의 가장 작은 단위의 작은 단위의 클러스터들을 그룹핑하면서 클러스터를 확장해 가는 상향식 계층적 클러스터링 방법과 구조가 유사하다. 제안된 알고리즘은 시간 간격 단위의 그룹핑을 통해 빈발항목 집합을 발견하므로 작은 시간 단위에서의 빈발

아이템셋뿐만 아니라 사용자가 원하는 일정한 시간 그룹 단위에 대한 빈발 아이템셋 탐사도 가능하다.

본 논문의 구성은 2장에서 우리가 제안하는 마이닝 기법의 기본적인 개념과 빈발 패턴 아이템셋의 탐사 방법 그리고 알고리즘을 설명한다. 그리고 3장에서는 실험을 통해 제안된 탐사기법의 성능을 평가하고, 4장에서 결론을 맺는다.

## II. 시간 단위 그룹핑을 이용하는 FPTuG 알고리즘

시간 단위를 고려한 빈발 패턴 아이템셋을 탐사하기 위해 시간 데이터베이스가 주어진다고 가정하고, 데이터베이스를 구성하는 트랜잭션에는 항목들이 존재한다. 그리고 마이닝을 하기 위해 사용자가 지정한 시간 단위가 주어지고, 빈발 아이템셋을 발견하기 위한 최소 지지도가 주어진다. 이 가정을 토대로 본 논문에서는 주어진 시간 그룹 단위에서 트랜잭션에 포함된 항목들에 대한 빈발 아이템셋을 탐사한다. 이를 위해 기초 데이터에서 사용자의 관심 항목과 최소 임계치를 만족하는 데이터 항목을 일차적으로 필터링하여 초기 데이터를 생성한다.

시간에 따라 발생하는 데이터를 포함하는 트랜잭션은 T로 표기하고, 트랜잭션에 대한 일련번호를 부여한다. 이에 대한 표현은 트랜잭션 T = {T1, T2, T3,...Tn}이고, Ti은 i번째 트랜잭션을 의미한다. 각 트랜잭션은 항목들을 포함하며, 1-size 이상으로 구성된 항목집합을 Itemset(I)로 정의한다. 항목집합 I = {I1, I2, I3,..,Ik}로 표현하고, 항목 Ik은 k번째 항목집합이다.

본 논문에서 제안하는 알고리즘을 수행하기 위한 기본 시간 단위(Time Unit)는 TU로 표기하고, TU는 일정한 수의 트랜잭션으로 구성된다. 시간 단위 TU= {$TU_{X1}$, $TU_{X2}$, $TU_{X3}$,..,$TU_{XY}$}로 표현하고, 여기서 X는 계층 레벨을 의미한다. TU는 같은 레벨의 인접한 시간 단위와 그룹화하여 상위 레벨의 TU를 형성한다. 예를 들면, $TU_{11}$과 $TU_{12}$가 그룹화되면 $TU_{21}$이 된다. 빈발항목 집합을 발견하기 위한 항목 지지도(Item support)는 시간 단위 $TU_{XY}$(Time Unit)에 포함되어 있는 트랜잭션 수에 대해 해당 항목을 포함하고 있는 트랜잭션의 수(Num)를 항목 지지도 Sup($I_i$)로 표기하고 이에 대한 식은 아래와 같다. 여기서 |$TU_{XS}$|는 시간 단위에 포함된 트랜잭션 수를 의미한다.

$$\mathrm{Sup}(I_i) = \frac{\sum Num(I_i \in TU_{XY})}{|TU_{XY}|} \quad \text{(식 1)}$$

최소 지지도 min_sup는 $TU_{XY}$에 포함되어 있는 모든 트랜잭션에서 아이템을 포함하고 있는 트랜잭션의 비율을 만족해야 하는 임계값이다.

본 논문에서 제안하는 빈발 패턴 마이닝 알고리즘 FPTuG(Frequent Pattern itemsets using Time unit Grouping)에서는 사용자에 의해 주어진 시간 단위 TU을 기반으로 하고, TU에는 일정 개수의 트랜잭션을 포함한다. TU의 크기를 |TU|로 표기한다. 예를 들어, 하나의 TU에 포함된 트랜잭션의 수가 5이면 |TU|=5이다. TU는 계층의 레벨과 시퀀스 정보를 포함하며 $TU_{XY}$으로 표현한다. 여기서 X는 그룹핑된 TU의 레벨을 의미하고, Y는 같은 레벨에서의 일련번호를 의미한다. TU를 그룹핑할 때는 같은 레벨의 일련된 순서에 의해 하나의 시간 그룹이 된다. 그림 1은 TU를 그룹핑하는 구조의 예를 보여준다. 그룹핑 된 TU의 계층 레벨과 시퀀스는 $TU_{XY}$=$TU_{X-1Y} \cup TU_{X-1Y+1}$ 에 의해 만들어진다. 표 1은 제안된 알고리즘을 설명하기 위한 트랜잭션의 예와 1-size의 빈발 패턴 아이템셋, FP(Frequent Pattern itemset)를 보여준다.

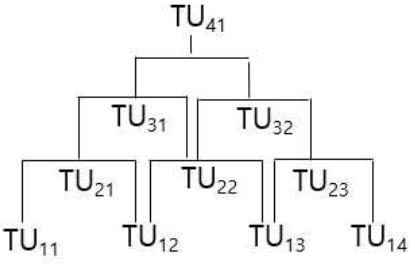


그림 1. 시간 단위(TU) 그룹핑 구조
Figure 1. Time unit grouping structure

빈발 아이템셋을 발견하는 과정은 초기 레벨 1의 각 TU 시퀀스에서 최소지지도 min_sup를 만족하는 빈발 아이템셋을 추출한다. 1-size의 빈발 아이템은 서로 다른 아이템과의 조인을 통해 2-size의 빈발 아이템을 발견하고, 이와 같은 방법으로 Size를 확장하고, 더 이상의

빈발항목이 발견되지 않을 때까지 반복한다. 레벨1에서의 빈발 아이템셋 발견을 마치면 레벨 1의 TU 시퀀스에 대한 그룹핑을 통하여 레벨 2의 TU를 구성한다. 레벨 1의 TU에서 추출한 빈발 아이템셋을 통합하여 레벨 2의 해당 TU의 아이템셋이 되고, 레벨 2의 그룹핑된 TU에서 최소 지지도를 만족하는 빈발 아이템셋을 탐사한다. 이와 같은 방법으로 레벨 2의 TU를 그룹핑하여 레벨 3의 TU에서 빈발한 아이템셋을 탐사한다. 이러한 과정은 더 이상 그룹핑할 TU가 존재하지 않을 때까지 수행한다. 따라서 일부의 시간 단위에서 뿐만 아니라 DB 전체에서의 빈발 아이템셋 탐색이 가능하다.

우리가 제안하는 FPTuG 알고리즘은 다음과 같은 3단계로 구성된다.

Step 1 : 초기 레벨에서 시간 단위 기준의 빈발항목 집합 탐색

Step 2 : 초기 레벨의 TU(time unit) 시퀀스를 그룹핑 하여 그룹내 포함된 트랜잭션에 대한 빈발 아이템셋 탐색

Step 3 : 각 레벨의 TU 그룹핑 과정을 통해 상위 레벨의 TU 시퀀스가 생성되고, 그룹핑 할 TU가 없을 때까지 반복하여 최상위 레벨의 TU 생성 및 빈발 아이템셋 탐색

Step 1에서는 계층 레벨 1에 해당하는 시간 단위에 대해, 해당 시간 단위에 포함된 트랜잭션들에 대한 빈발항목 집합을 발견한다. Step 2는 Step 1에서 발견된 빈발 항목집합의 시간 단위 TU를 그룹핑하여 레벨 2를 형성하고, 그룹화된 상위 레벨의 TU에서 빈발 항목집합을 탐사한다. 예를 들면, TU의 크기 |TU|=4인 경우, TU11과 TU12 포함된 전체 8개의 트랜잭션에서 빈발한 항목 집합을 발견하여 TU21에 저장한다. 즉, TU11과 TU12에서 빈발한 항목들을 통합하고 이들에 대한 최소 지지도를 만족하는지 확인하여 빈발항목의 여부를 결정한다. 다음으로 TU12와 TU13에서 공통으로 빈발한 항목을 추출하여 TU22에 저장한다. 이와 같은 방법으로 레벨 1의 TU1Y와 TU1Y+1에서(1≤Y and Y≥ n-1) 빈발한 연관 항목을 발견하고 레벨 2의 TU2Y 시퀀스에 저장한다. 그리고 레벨 2 빈발항목의 TU 시퀀스를 그룹핑하여 레벨 3의 TU3Y 시퀀스를 생성한다. 이와 같은 과정은 시간 간격 단위를 점차 확장하는 방식이며, 계층적 TU 레벨 시퀀스가 구성되므로 결과적으로 전체 DB 포함된 트랜잭션에 대한 빈발 항목을 탐사하는 결과를 생성하는 것이다. 제안된 FPTuG는 이와 같이 3단계의 과정을 통해 빈발 아이템셋을 발견하는 데, 각 레벨에서 빈발 아이템셋을 발견하기 위해 사용되는 알고리즘 TuECLAT은 다음과 같다.

**TuECLAT Function**

Input: TU in Database , min_sup
Output: Frequent Itemsets at level 1
Method:

```
1. Create Data Matrix(DM) by all itemsets in TU,
2. Create Pattern Tree(PT) with a root of null
3. for each itemset in DM do
4.       if Sup(itemset) ≥ min_sup then
5.          add itemset to PT
6. end for
7. Find frequent pattern itemsets by joining each
   itemset in PT
```

TuECLAT 알고리즘은 빈발 아이템셋 탐사에서 후보항목 생성 시간을 줄이기 위해 매트릭스를 구성하는 Eclat 알고리즘[14]을 시간 단위에 대해 적용한 것이다. 전체의 Temporal DB에 포함되어 있는 항목들은 열로 표현하고, 각 항목들을 포함하는 트랜잭션 아이디를 행으로 하는 매트릭스를 구성한다. 생성된 매트릭스 정보를 이용하여 각 시간 단위 TU에서 최소 지지도를 만족하는 빈발 아이템셋을 탐사한다.

표 1의 예제 트랜잭션을 포함하는 DB에 대해 min_sup= 0.5를 가정하여 마이닝 과정을 설명하도록 한다. $TU_{12}$의 항목에 대한 지지도의 예를 들면, Sup(a) = 3/4, Sup(c) = 3/4, Sup(ac) = 2/4, Sup(bd) = 2/4 이고, min_sup를 만족하므로 a, c, ac, bd 은 빈발항목이 된다. 표 1에서는 레벨 1의 TU에서 최소 지지도를 만족하는 사이즈 1의 빈발항목들(FP)을 보여준다. 1-size의 빈발항목들은 다른 항목들과의 조인을 통해 빈발항목 여부를 체크하여 2-size 또는 3-size의 빈발항목을 발견한다. 레벨 1의 TU 시퀀스는 그룹핑을 통하여 시간 단위가 확장된 레벨 2의 TU 시퀀스를 생성한다. 예를 들면, $TU_{11}$과 $TU_{12}$를 그룹핑하여 레벨 2의 $TU_{21}$를 생성하기 위하여 레벨 1의 TU 시퀀스에서의 빈발항목을 합집합은

표 1. 트랜잭션의 예와 size-1 빈발항목 집합
Table 1. Transaction example and size-1 frequent itemsets

| TU | TID | Item | FP |
|---|---|---|---|
| $TU_{11}$ | 1 | d | c, d |
| | 2 | c, d | |
| | 3 | c | |
| | 4 | d | |
| $TU_{12}$ | 5 | a, c, d | a, b ,c, d |
| | 6 | a, b, c, d | |
| | 7 | b, c, d | |
| | 8 | a, d | |
| $TU_{13}$ | 9 | b | a, b, c |
| | 10 | a, c | |
| | 11 | a, b, c | |
| | 12 | b, c | |
| $TU_{14}$ | 13 | b, d | b, c, d |
| | 14 | b, c, d | |
| | 15 | b | |
| | 16 | b, c, d | |

$FP(TU_{11}) \cup FP(TU_{12})$ = {a, b, c, d}이고, 후보 항목이 된다. 그리고 후보 항목에 대한 $Sup(I_i)$를 계산하여 최소 지지도를 만족하는지 검사하여 빈발 아이템셋을 결정한다. 빈발 아이템셋이 결정되면 이들이 $FP(TU_{2,1})$의 항목이 된다. 이와 같은 방법으로 시간 단위 TU 시퀀스를 그룹핑하여 레벨을 증가시키면서 빈발항목들이 결정된다.

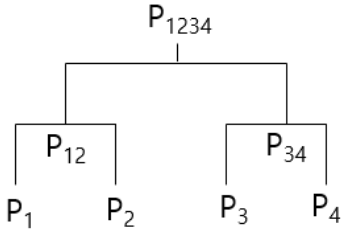


그림 2. HTAR 계층 구조
Figure 2. HTAR Hierarchical Structure

그림 2는 우리가 제안하는 FPTuG 구조와 유사한 기존 연구[11]에서 제안한 알고리즘 HTAR(Hierarchical Temporal Association Rule)의 계층 구조이다. FPTuG의 계층 구조는 각 레벨에 있는 모든 TU 시퀀스를 연속적으로 그룹핑하면서 시간 단위를 확장하는 방식의 구조이다. 반면 HTAR의 구조는 시간 간격의 그룹핑을 연속적으로 구성하지 않고, 하나의 TU는 하나의 그룹핑에만 속하게 하는 방식이기 때문에 시간 단위 사이에 공백(empty)의 부분이 발생하게 된다. 즉, $TU_{12}$~ $TU_{13}$의 시간 구간이 그룹핑구조에서 제외되므로 빈발항목 탐사가 이루어지지 않고, 이로 인해 빈발항목 탐색의 정확성을 보장하기 어렵다. FPTuG의 구조는 그룹핑된 시간 간격의 단위가 연속적이므로, 시간 변화에 따른 빈발 아이템의 변화를 감지할 수 있고, 원하는 어떤 시간 구간에서도 데이터의 요구나 분석이 필요할 때 손실 없는 데이터를 제공할 수 있다.

## III. 실험 및 평가

우리가 제안하는 알고리즘 FPTuG의 효율성을 평가하기 위해 HTAR 알고리즘[11]과의 실험 결과를 비교한다. 실험은 R로 수행하였으며, 평균 항목의 수 10개를 포함하는 트랜잭션으로 하고, 트랜잭션 10K개의 실험 데이터 셋을 생성하였다. 그리고 알고리즘의 성능 평가를 위해 생성한 데이터셋의 전체 아이템 수는 100 ~ 4000개이다. T10I1000D10K 데이터셋은 1000개의 아이템으로 구성된 10K 데이터 셋을 의미한다. 트랜잭션의 수를 625로 할 경우 10K 데이터셋에서는 16개의 시간 간격의 단위가 만들어진다.

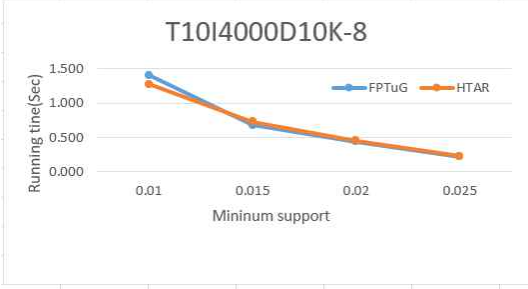


그림 3. 실행시간 (T10I4000D10K-8)
Figure 3. Running time (T10I4000D10K-8)


그림 4. 실행시간 (T10I4000D10K-16)
Figure 4. Running time (T10I4000D10K-16)

그림 3과 그림 4는 T10I4000D10K의 8 time periods와 16 time periods에서 최소 지지도에 따른 실행시간을 보여준다. 실험 결과에서 최소 지지도가 낮을 때는 FPTuG 알고리즘의 실행시간이 더 많이 소요되지만, 지지도가 높아지면서 FPTuG 알고리즘과 HTAR 알고리즘의 실행시간이 비슷해지는 것을 알 수 있었다. 최소 지지도가 높아지면 임계치를 만족하는 빈발항목 수가 적어지기 때문으로 판단된다. 또한 FPTuG의 계층 구조가 HTAR의 구조보다 더 촘촘한 구조를 갖기 때문에 레벨 높이가 증가하므로 마이닝 수행시간이 더 많이 소요되는 것으로 사료된다.

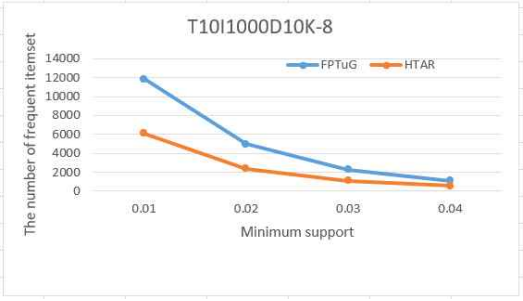


그림 5. 빈발항목 수 (T10I1000D10K-8)
Figure 5. The number of frequent itemsets(T10I1000D10K-8)


그림 6. 빈발항목 수 (T10I4000D10K-8)
Figure 6. The number of frequent itemsets(T10I4000D10K-8)

다음은 최소 지지도를 다르게 하면서 생성되는 빈발 아이템의 수를 비교하였다. 그림 5와 그림 6은 데이터 셋 T10I1000D10K와 T10I4000D10K에서 8 time periods인 경우, 최소 지지도가 0.01에서 0.04까지 변화하면서 수행한 결과를 보여준다. 실험 결과에서 FPTuG 알고리즘이 기존 알고리즘보다 많은 아이템셋을 탐사하는 것을 볼 수 있다. 낮은 지지도에서는 기존 알고리즘보다 2배 이상의 빈발 아이템셋을 생성하는 것으로 나타났다. 이것은 시간 단위의 그룹핑 구조의 차이로 인해 탐사되는 빈발 아이템셋이 다르고, 추출 개수에도 영향을 주는 것으로 판단된다.

FPTuG 알고리즘과 HTAR 알고리즘의 실험 결과를 정리하면, 지지도에 따른 빈발항목의 수는 FPTuG이 HTAR보다 평균 2~3배의 차이를 보였고, FPTuG의 실행시간은 평균 1.29배와 1.10배의 결과를 보였다. 결과적으로 본 논문에서 제안된 FPTuG 알고리즘은 HTAR 알고리즘보다 많은 실행시간이 소요되지만, 빈발항목의 수는 약 2배 이상 탐사되는 것을 알 수 있다.

## Ⅳ. 결 론

기존의 일반적인 시간 데이터 마이닝에서는 대부분 트랜잭션의 발생 시점만을 고려하였고, 일정한 시간 간격 단위는 고려하지 않았다. 본 논문에서는 시간 데이터베이스에서 일정한 시간 간격 단위를 기준으로 시간 단위에 포함된 트랜잭션들에서 빈발 패턴 항목을 발견하는 FPTuG 알고리즘을 제안하였다. 제안된 알고리즘에서는 시간 단위에 대한 계층적 그룹핑 구조를 기반으로 확장된 시간 단위에서의 빈발 아이템셋을 발견한다. 제안된 알고리즘의 성능평가를 위해 기존 연구의 HTAR 알고리즘과 비교하는 실험을 하였고, 실험 결과에서 FPTuG 알고리즘이 기존 알고리즘보다 수행시간은 1.2배 정도 더 많이 소요되지만, 추출된 빈발항목의 수는 2배 더 많이 탐사되는 것을 알 수 있었다. FPTuG는 연속적인 시간 단위 그룹핑을 적용하므로 시간의 흐름에 따른 정확한 빈발항목의 데이터 분석이 가능하다. 그러므로 주어진 시간 범위에서 빈발 아이템을 탐색해야 하는 네트워크 트래픽 분석, 웹 서버 부하, 주식 거래 등 시간에 따른 데이터의 변화와 관련된 응용 분야에 활용될 수 있다.

## References

[1] Y. Lee, J. Lee, D. Chai, B. Hyun and K. Ryu, “Mining temporal interval relation rules from temporal data,“ The journal of systems and software, Vol.82, No.1, pp. 155-167, 2012. DOI: 10.1016/j.jss.2008.07.037

[2] Z. Zhang and Q. Fu, “Data mining algorithm of frequent probability item based on sliding window,” Applied Mechanics and materials, Vol.

602-605, pp.3268-71, 2014. DOI:10.4028/www.scientific.net/AMM.602-605.3268
[3] C. H. Lee, C. R. Lin and M. S. Chen, "On mining general temporal association rules in a publication database," The IEEE International Conference on Data Mining, pp. 337-344, 2001. DOI:10.1109/ICDM.2001.989537
[4] J. M. Ale and G. H. Rossi, "An Approach to Discovering Temporal Association Rules," in Proceedings of the 2000 ACM symposium on Applied computing ACM, 2000. DOI:10.1145/335603.335770
[5] Y. Kim, W. Kim and U. Kim, "Mining Frequent Itemsets with Normalized Weight in Continuous Data Streams," Journal of Information Processing Systems, Vol. 6, No. 1, pp. 79-90, 2010. DOI:10.3745/JIPS.2010.6.1.079
[6] C. K. Leung and B. Hao, "Mining of Frequent Itemsets from Streams of Uncertain Data," IEEE International Conference on Data Engineering, 2010. DOI:10.1109/ICDE.2009.157
[7] Y. Lee, J. Lee, D. Chai, B. Hwang and K. Ryu, "Mining Temporal Interval Relational Rules from Temporal Data," Journal of System and Software, Vol. 82, pp. 155-167, 2009. DOI:10.1016/j.jss.2008.07.037
[8] L. Sacchi, C. Larizza, C. Combi and R. Bellazzi, "Data mining with Temporal Abstractions: learning rules from time series," Data Mining and Knowledge Discovery, Vol. 15, No. 2, pp. 217-247, 2007. DOI:10.1007/s10618-007-0077-7
[9] J. Pei, J. Han, B. M. Asi, J. Wang, H. Pinto, Q. Chen, U. Dayal and M. Hsu, "Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach," IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No.11, pp.1424-1440, 2004. DOI:10.1109/TKDE.2004.77
[10]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.
[11]T. P. Hong, G. C. Lan, J. H. Su, P. S. Wu and S. L. Wang, "Discovery of temporal association rules with hierarchical granular framework," Applied Computing and informatics, Vol. 12. No. 2. pp. 134-141, 2016. DOI:10.1016/j.aci.2016.01.003
[12]V. Srinivasan and M. Aruna, "Mining Association Rules to Discover Calendar Based Temporal Classification," International Conference on Computing, Communication and Networking, pp. 1-12. 2008. DOI:10.1109/ICCCNET.2008.4787754
[13]E. Boudaillier and G. Hebrail, "Interactive Interpretation of Hierarchical Clustering," Intelligent Data Analysis 2, pp. 229-244, 1998. DOI:10.1016/S1088-467X(98)00026-2

※ 이 논문은 2022년도 남서울대학교 학술연구비 지원에 의해 연구되었음.