JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 19, No. 3, pp. 471-477, Mar. 2018

그래프를 이용한 빈발 서비스 탐사

황 정 희
남서울대학교 컴퓨터학과

Mining Frequent Service Patterns using Graph

Jeong-Hee Hwang
Department of Computer Science, Namseoul University, Cheonan 31020, Korea

[요 약]

시간의 변화에 따라 사용자의 관심도는 변화한다. 이 논문에서는 유비쿼터스 환경에서 연령, 시기, 계절 등에 따라 변화하는 사용자의 서비스 관심도를 고려하기 위하여 서비스에 대한 관심도를 동적 가중치로 부여하여 사용자에게 적합한 서비스를 추천하기 위한 방법을 제안한다. 사용자에게 제공한 서비스 이력 데이터를 기준으로 시기나 연령에 따른 일반적인 서비스 규칙을 저장하고, 실시간으로 변화하는 서비스의 관심도를 고려한 최신의 서비스 규칙을 지속적으로 추가하여 사용자의 관심 변화를 반영하는 서비스를 제공하기 위한 방법이다. 이를 위해 사용자에게 제공하는 일련의 서비스는 트랜잭션으로 고려하고 서비스는 항목으로 고려하여 서비스의 연관관계를 그래프로 표현하고, 이를 기반으로 빈발 서비스 항목을 발견한다. 발견된 빈발 서비스 항목은 사용자에게 유용한 최신의 정보 서비스를 의미한다.

[Abstract]

As time changes, users change their interest. In this paper, we propose a method to provide suitable service for users by dynamically weighting service interests in the context of age, timing, and seasonal changes in ubiquitous environment. Based on the service history data presented to users according to the age or season, we also offer useful services by continuously adding the most recent service rules to reflect the changing of service interest. To do this, a set of services is considered as a transaction and each service is considered as an item in a transaction. And also we represent the association of services in a graph and extract frequent service items that refer to the latest information services for users.

색인어 : 데이터 마이닝, 연관규칙, 빈발패턴, 스트림 데이터, 가중치

Key word : Data Mining, Association Rule, Frequent Pattern, Stream Data, Weight

http://dx.doi.org/10.9728/dcs.2018.19.3.471

Received 29 December 2017; **Revised** 22 January 2018
Accepted 25 March 2018

***Corresponding Author; Jeong-hee Hwang**

Tel: +82-41-580-2108
E-mail: jhhwang@nsu.ac.kr

Ⅰ. 서 론

센서와 무선 통신 기술의 발달로 인하여 시공간의 제약 없이 데이터를 수집할 수 있는 스트림 데이터 시스템 환경이 출현하였다. 이러한 환경의 응용 분야는 이동 물체의 경로 추적, 사용자의 시공간 환경을 고려한 서비스 제공, U-health 등 매우 다양하다. 그러나 스트림 데이터 시스템은 전통적인 데이터베이스 시스템에 비해 작은 용량의 베터리와 메모리, 소형 프로세서 등 제약사항이 존재한다. 스트림 데이터 마이닝은 비한정적인 데이터 집합 즉, 데이터 스트림에 대한 실시간 분석 능력을 지원한다. 즉, 어느 시점에서나 해당 시점까지의 모든 트랜잭션을 포함하는 현재 데이터 스트림에 대한 마이닝 결과를 얻을 수 있도록 지원한다. 그러나 스트림 데이터 마이닝은 최대 한 번 탐색하고, 메모리 사용량은 무한히 증가되지 않으며 최신의 데이터에 대한 빠른 분석을 제공하는 것을 목적으로 한다 [1,2,3].

데이터의 빈발 항목 탐사에서는 일반적으로 트리 구조를 많이 이용한다[4,5,6]. 트리구조는 트랜잭션내의 항목간의 연관성을 나타내기 위해 같은 항목이라도 다른 트랜잭션에 존재하는 경우 서로 다른 트랜잭션의 항목관계를 표현하기 위해 중복해서 표현해야 하는 어려움이 있다. 이에 반해 그래프를 이용한 마이닝에서는 트랜잭션을 그래프로 모델링하는 방법을 이용한다[8,9,10]. 그래프는 상하관계로 항목간의 관계를 표현하는 트리 구조와는 다르게 항목간의 관계를 다양하게 표현할 수 있기 때문에 트리 구조보다 더 단순하게 항목의 관계를 표현하여 빈발 항목을 발견할 수 있다.

일반적으로 실생활에서 모든 서비스가 같은 중요도를 갖지 않는다. 가중치 패턴 탐사(weighted pattern mining)는 항목들이 다른 가중치를 가질 경우 가중치를 고려하는 높은 빈발도를 나타내는 패턴을 탐사하는 것이다[1,2]. 서비스에 가변의 가중치를 정의하여 빈발 항목을 탐사하는 것은 빈발하지 않더라도 상대적으로 관심있는 서비스에 대한 연관규칙을 탐사할 수 있는 효율적인 방법이고, 빈발도만 가지고 연관 서비스를 탐사하는 것보다 최근의 관심 정도를 반영하는 것이므로 더 의미있는 정보를 탐색할 수 있다.

이 논문에서는 사용자에게 최적의 서비스를 제공하기 위하여 최근에 이용되는 서비스 스트림 데이터에 대한 빈발 패턴을 탐사하는 방법을 기술한다. 사용자의 관심은 시간과 공간, 시기, 연령에 따라 변화한다. 그러므로 사용자의 관심 정도를 반영하는 가중치를 서비스 데이터에 부여하여 지속적으로 입력되는 서비스 스트림 데이터에 대한 빈발 패턴을 탐사한다. 이를 위해 윈도우 사이즈에 포함되는 배치(Batch)의 트랜잭션 항목들을 노드로, 항목 관계를 에지로 표현하여 그래프를 구성하고, 이를 이용하여 가중치와 빈발도를 기반으로 임계치를 만족하는 빈발 항목들을 발견한다. 제안된 방법은 사용자의 연령이나 시기, 관심 분야를 세분화하여 적용하면 사용자에게 더 높은 만족도의 서비스를 제공할 수 있는 기반이 된다.

이 논문의 구성은 다음과 같다. 2장에서는 빈발 패턴 마이닝의 관련 연구를 기술하고, 3장은 이 논문에서 제안하는 그래프를 이용한 서비스 빈발 항목 탐사과정과 알고리즘을 설명한다. 4장은 제안된 방법의 성능을 분석하는 실험 결과를 기술하고, 5장은 결론으로 맺는다.

Ⅱ. 관련연구

빈발패턴 마이닝은 트랜잭션에서 미리 정의된 최소 지지도를 만족하는 빈발 항목집합을 찾아내고, 이들 빈발 항목집합들 간의 연관성 정도를 반영하는 연관 규칙을 찾아내는 것이다. 즉, 트랜잭션 데이터베이스에서 나타난 여러 패턴중에서 주어진 임계값을 만족하는 빈도의 패턴을 발견하는 방법이다. Apriori 알고리즘을 기반으로 하는 빈발패턴 마이닝에서 만일 어떤 패턴 a가 빈발하지 않은 패턴이면 a의 모든 슈퍼셋(Super Set)은 빈발하지 않은 패턴이 된다. 이를 Anti-monotone 성질이라고 한다[7].

스트림 데이터는 크기가 무한하고 연속적이며 데이터의 경계가 없다. 그러므로 데이터 스트림을 모두 저장하거나 여러 번 스캔하여 데이터를 처리하는 방법은 불가능하다. 따라서 연속적으로 발생하는 데이터를 요약하거나, 일정 구간의 윈도우 단위로 분할하여 관심있는 데이터에 대한 탐사가 이루어져야 한다[3]. [1]에서는 데이터 스트림에 대한 슬라이딩 윈도우 기반으로 FP-tree를 이용하여 빈발 항목을 찾는 방법을 제안하였다. 슬라이딩 윈도우 기반 마이닝은 사용자의 정보 요청이 들어오면 현재 윈도우에 저장된 최신 데이터로부터 최소 임계치를 만족하는 모든 빈발 패턴 항목들을 마이닝한다. [11,12]에서는 이벤트 시간 관계에 대한 연관 규칙을 탐사하는 방법을 제안하였고, [13]에서는 정의된 윈도우 구간 동안에 최소 빈발 임계값을 만족하는 이벤트를 탐사하는 방법을 제안하였다. [3]에서는 스트림 데이터 환경에서 연관 규칙을 탐사하는 MILE(Mining from muLtiple strEams) 방법을 제안하였다. 센서에서 수집된 이벤트에 대한 트리 기반 인덱스를 구축하고 다차원 스트림 데이터 사이의 연관규칙을 탐사한다. 제안된 방법은 정의된 윈도우 구간 동안에 수집된 이벤트에 대한 연관 규칙만을 탐사한다.

일반적인 연관규칙 알고리즘은 모든 항목에 대한 중요성을 동일한 것으로 간주한다. 그러나 실제 응용 분야에서 단위 항목들은 서로 다른 중요성을 가진다. 그러므로 빈발패턴 탐사에서 단위 항목에 대한 차별화된 중요성을 고려하는 경우 빈도가 낮아도 사용자의 흥미도나 관심도가 높은 서비스 정보를 추출할 수 있다. 기존 연구[1,2,14,15,16]에서도 가중치를 부여하여 마이닝하는 방법을 제안하였다. 이들 연구에서는 시간의 변화와는 관계없이 데이터의 특성만을 고려하여 가중치를 부여하는 방법 또는 같은 항목의 데이터에 대해 항상 동일한 가중치를 부여하여 빈발 항목을 발견하였다. [14]는 패턴의 중요도에

따라 가중치를 부여하여 패턴 사이에 존재하는 연관규칙을 탐사하는 WIP(Weighted Interesting Patterns)방법을 제안하였다. WIP는 패턴의 중요도 및 관심도에 따라 가중치를 부여하여 더 많은 관심을 갖는 패턴을 연관 규칙으로 탐사한다. 그러나 수집된 데이터를 대상으로 탐사하기 때문에 스트림 데이터에 적용하기는 어렵다.

빈발 항목 탐사를 위해 [5]에서는 스트림 데이터를 위한 DSTree를 구성하였고, [4]에서는 점진적 마이닝을 위하여 FP-tree[15]를 변형한 CanTree트리를 구성하는 방법을 제안하였다. 그리고 [1]에서는 스트림 데이터의 특성을 고려하여 FP-Tree에 이동 윈도우 기법을 적용하였다. 그래프를 이용한 방법으로 [8]에서는 다중레벨에서 각 레벨에서의 빈발 패턴을 발견하기 위해, 인접 매트릭스를 구성하고 각 레벨에서의 빈발 패턴을 탐사하였다. [9]에서는 FP-Graph를 제안하여 그래프 기반의 빈발 항목 집합 탐사 알고리즘을 제안하였다. [10]은 그래프를 기반으로 연관된 항목간의 관계를 매트릭스로 표현하여 빈발패턴을 마이닝하는 방법을 제안하였다. 그러나 스트림 데이터에는 적용하기 어렵다. 스트림 데이터를 위한 마이닝에서는 윈도우의 변화에 따라 트리의 형태를 재구성하는 데 많은 비용이 들기 때문에 그래프 구조가 효율적이다.

Ⅲ. 가중치 그래프를 이용한 빈발 항목 추출

사용자에게 제공하는 서비스 항목집합을 I={I_1, I_2, I_3,..}로 표현하여 마이닝 과정을 설명한다. 데이터를 윈도우 단위로 분할하여 마이닝하는 예를 보이기 위해, 하나의 배치에는 3개의 트랜잭션을 포함하고, 윈도우 사이즈는 두 개의 배치를 포함하는 것으로 가정한다. 즉, 배치 B1, B2는 하나의 윈도우 단위로 처리되고, B3가 입력되면 가장 오래전에 입력된 B1은 삭제되고 B2, B3가 현재 윈도우가 된다. 각 항목의 가중치 지지도 Iwgt(Item weight)는 배치에서 항목의 가중치와 빈도를 곱하여 계산한다. 서비스 항목의 관심도는 시간이 지남에 따라 변화한다는 것을 고려하므로 동일 항목일지라도 각 배치에서 다른 가중치를 부여한다.

이 논문에서 제안하는 마이닝 과정은 세 단계로 이루어진다. 첫번째 단계에서는 배치에 있는 항목중에서 최대 가중치 LMwgt(Local Maximum weight)보다 작은 항목가중치 지지도를 갖는 항목은 필터링하여 가지치기한다. 두 번째 단계에서는 필터링된 항목들에 대한 항목간의 관계를 그래프로 나타낸다. 그리고 마지막으로 그래프에서 항목간의 관계와 임계치를 만족하는 가중치 지지도를 고려하여 빈발 서비스 항목을 발견한다.

표 1은 각 배치의 항목들에 대한 가중치의 예를 보여준다. 표 1의 배치 B1에서 LMwgt은 0.9이고, 항목 b의 Iwgt(b)는 0.6*2=1.2이다.

표 1. 항목의 가중치 예
Table 1. Example of item weight

Batch	T-id	Items	Weight					
1st	T_1	a, b	a	b	c	d	e	f
	T_2	a, b, c	0.9	0.6	0.7	0.3	0.5	0.8
	T_3	a, d						
2nd	T_4	a, b, c	a	b	c	d	e	f
	T_5	b, c, e	0.6	0.5	0.8	0.7	0.8	0.4
	T_6	c, d, f						
3rd	T_7	c, e	a	b	c	d	e	f
	T_8	a, d	0.3	0.6	0.9	0.7	0.7	0.5
	T_9	c, d, e						

이 논문에서는 마이닝을 위해 빈발하게 발생하는 연관된 항목의 패턴을 발견하기 위해 가중치를 갖는 서비스 항목의 조합을 나타내는 패턴 가중치 Pwgt(Pattern weight)를 사용하며, 다음과 같은 식을 이용한다.

$$\text{Pwgt(I)} = \frac{\sum_{i=1}^{length(I)} weight(I_i)}{length(I)} \quad (1)$$

여기서 length(I)는 패턴을 구성하는 항목의 길이이고, weight(I_i)는 각 항목의 가중치를 의미한다. 그러므로 Pwgt(I)는 패턴을 구성하는 각 항목의 가중치 합을 패턴의 길이로 나눈 값으로 패턴의 평균값이다.

그리고 빈발 후보항목을 줄이기 위해 항목의 최대 가중치를 나타내는 IMwgt(Item Maximun weight)를 이용한다. 가지치기 하는 기준이 되는 항목의 IMwgt은 다음 식과 같다.

$$\text{IMwgt}(I_i) = \sum_{k=1}^{n} LMwgt(B_k) \times Freq(I_i) \quad (2)$$

여기서 LMwgt(Local Maximum Weight)는 항목 I_i가 발생하는 배치(B_k)의 항목중 가장 큰 가중치를 의미하고, n은 배치의 수이고, $Freq_i$(I)는 배치 B_k에서의 항목 발생빈도를 의미한다. 가지치기의 기준은 임의의 항목에 대한 IMwgt(I)가 임계치인 최소 가중치 지지도 min_wsup를 만족하지 못하면 해당 항목은 마이닝의 전처리 과정에서 제외된다. 연관규칙 마이닝에서 개별 항목은 다른 항목들과 함께 발생하는 패턴항목의 빈발여부를 결정하는 데, 개별 항목의 IMwgt가 주어진 임계치를 만족하지 못하면 다른 항목과 함께 발생하는 패턴항목의 가중치도 임계치를 만족하지 못할 것임으로 미리 제거하는 것이다. 즉, IMwg는 항목의 가중치를 LMwgt을 이용한 최대값으로 계상하여 빈발 가능성을 예측하고, 이를 만족하지 못하면 가지치기 한다.

마이닝 과정의 각 단계에서의 처리 과정을 표 1의 예제를 이용하여 기술한다. 첫 번째 단계는 마이닝을 위한 그래프를 생성하기 전에 가지치기 하는 것이다. 표 1의 윈도우 W1에 포함되어 있는 배치 B1, B2의 항목들에서 LMwgt를 이용하여

IMwgt를 계산하고 임계치보다 작은 항목은 그래프 생성에서 제외한다. 윈도우 W1의 항목들에 대한 IMwgt(I)를 계산하면 a:0.9*3+0.8*1=3.5, b:0.9*2+0.8*2=3.4, c:0.9+0.8*3=3.3, d:0.9+0.8=1.7, e:0.8, f:0.8이다. 마이닝하기 위한 최소 가중치 지지도 min_sup을 1.2 가정하면, 임계치를 만족하지 못하는 e, f항목은 가지치기하여 그래프 생성에서 제외된다. 가지치기에서 고려해야 할 것은 트랜잭션 단위로 가지치기를 하면 빈발 가능한 항목이 제외될 수 있으므로, 윈도우 단위로 항목에 대한 가지치기를 한다. 그림 1은 표 1의 트랜잭션 예제에 대해 윈도우 슬라이딩에 의한 윈도우와 배치의 포함관계를 보여준다. 하나의 윈도우에 포함된 배치의 수를 윈도우 사이즈라고 하고, 각 배치는 일정한 수의 트랜잭션을 포함한다. 윈도우는 일련의 번호를 가지며, 윈도우 슬라이딩에 의해 윈도우에 포함되는 일정한 배치의 수에 의해 가장 오래된 배치는 삭제하고, 새로 입력되는 배치가 포함된다. 즉, 윈도우 W1={B1, B2}이고, 윈도우 슬라이딩 의해 윈도우 W2={B2, B3} 이다.

두 번째 단계는 가지치기에 의해 빈발 가능성이 있는 항목들에 대한 연관성을 그래프로 구성한다. 표 1에서 배치 B1, B2를 포함하는 윈도우 W1을 그래프로 표현한 것이 그림 2의 (a)이고 윈도우W1에서 윈도우 W2로 전이되는 과정을 표현한 것이 (b)이다. 그래프에서 각 노드는 항목을 나타내고 항목으로 들어오는 에지의 레이블에는 항목관의 관계를 표현한다. 그림 2의 (a)에서 항목 a로 들어오는 에지의 레이블 a:2,1은 배치 B1에서 항목패턴 ab의 빈도가 2이고, 배치 B2에서는 빈도가 1이라는 것을 나타낸다. 배치 B1과 배치 B2는 윈도우 W1에 속하고 배치 B3의 삽입으로 윈도우 사이즈가 2라 할 때 윈도우 이동으로 배치 B1에 속한 항목들은 일괄 삭제되고, 배치 B2와 배치 B3이 윈도우 W2가 된다. 윈도우 W2의 항목들에 대한 IMwgt(I)를 계산하면 a:0.8+0.9=1.7, b:0.8*2=1.6, c:0.8*3+0.9*2=4.2, d:0.8+0.9*2=2.6, e:0.8*1+0.9*2=2.6, f:0.8이다. 여기서 임계치를 만족하지 못하는 f항목을 제외하고 그래프를 구성한다.

세 번째 과정은 그래프에 있는 각 항목을 기준으로 빈발 항목을 발견한다. 이 과정에 대한 설명을 위해 그림 3의 항목 e를 기준으로 빈발 항목을 추출하는 방법을 기술한다. 빈발 항목은 그래프에서 각 항목으로 들어오는 에지의 레이블에 있는 항목들이 최소 가중치를 만족하는지의 여부에 따라 빈발 항목을 발견하는 과정으로 이루어진다. 이 논문에서는 개별항목과 연관

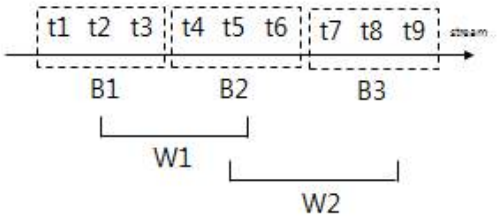
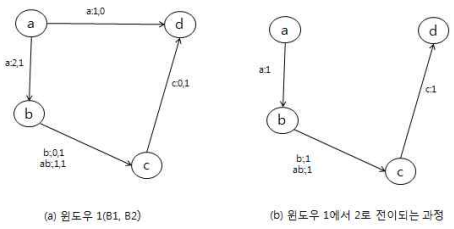


그림 1. 슬라이딩 윈도우
Fig. 1. Sliding window



(a)W1 (b)Sliding window

그림 2. 윈도우 W1과 슬라이딩 윈도우
Fig. 2. Window W1 and sliding window

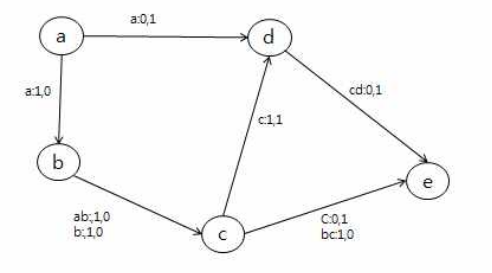


그림 3. 윈도우 W2
Fig. 3. Window W2(B2,B3)

되어 발생하는 항목중에서 가장 큰 가중치를 가지고 판단하며, 이를 연관 최대 가중치,RMwgt(Related Maximum weight)라 하고, 이를 이용하여 후보항목 가능성 여부를 결정한다. 그림 3의 그래프에 대하여 Rwgt를 이용하여 후보항목을 줄이면서 빈발패턴을 탐사하는 방법을 설명한다.

항목 e에 들어오는 에지는 c:0,1 bc:1,0 cd:0,1 이다. 이 패턴들을 개별항목으로 분리하여 빈발도를 구하면 b:1,0 c:1,2 d:0,1 이고, 표 2는 이를 나타낸 것이다. 이 연관 발생 항목중 가장 큰 가중치는 0.9이므로 Rwgt =0.9이고 Rwgt를 적용하여 임계치를 만족하는 후보항목만을 선별한다. b:1*0.9 = 0.9, c:3*09=2.7, d:0.9*1=0.9이므로 임계치를 만족하는 c항목만이 후보항목이 되고, 패턴 항목 ce:1,2에 대한 실제 가중치를 적용하여 빈발 여부를 검사한다. 패턴 ce에 대한 빈발 가중치는 (0.8+0.8)/2+(0.9+0.7)/2*2 =2.4 이므로 임계치를 만족하므로 빈발하다.

다음으로 항목 d에 대한 빈발 항목을 발견하는 방법을 설명한다. d로 들어오는 에지의 항목은 a:0,1, c:1,1이 있다. 이들 항목에서 가장 큰 가중치는 항목 c의 가중치 0.9이다. 그러므로 Rwgt=0.9를 이용하여 후보 가능 항목을 검사하면 a:0.9 *1 =0.9, c:0.9*2=1.8이므로 임계치를 만족하는 항목 c만이 후보항목이 된다. 그러므로 패턴 cd:1,1에 대한 실제 가중치를 적용

표 2. 항목 e의 프로젝션 DB
Table 2. Projected DB of item e

	b	c	d	weight
B2	1	1	0	b:0.5, c:0.8, d:0.7
B3	0	2	1	b:0.6, c:0.9, d:0.7

하면 (0.8+0.7)/2 + (0.9+0.7)/2 =1.55가 되어 임계치를 만족하며 빈발하다. 이와 같은 방법으로 모든 항목에 대해 들어오는 에지의 항목 기준으로 빈발 여부를 검사한다. 빈발 여부를 검사할 때 만약 d항목으로 들어오는 에지의 항목 a와 c가 모두 빈발하면 패턴 ad, cd은 당연히 빈발하지만, 패턴 acd의 빈발 여부는 개별항목의 실제 가중치를 적용하여 결정해야 한다. 패턴 ac, cd이 빈발하다고 하여 패턴 acd이 항상 빈발하지는 않기 때문이다.

일반적으로 개별항목보다는 두 개 이상의 항목으로 이루어진 연관패턴 항목의 빈발 여부에 관심이 더 많다. 그러나 개별항목의 빈발 여부를 검사할 때는 그래프에서 기준이 되는 항목으로 들어오는 에지와 나가는 에지의 빈발도를 함께 고려해야 더 정확한 결과를 얻을 수 있다. 그림 4에서 항목 a의 경우, 항목 b와 항목 c 기준으로 빈발여부를 검사할 때 항목 a은 빈발하지 않다. 그러나 항목 a 기준으로 나가는 에지의 항목 ab, ac를 모두 고려하면 빈발도 2를 기준으로 적용할 때 a항목은 빈발하다. 그러므로 개별항목의 빈발여부는 나가는 에지와 들어오는 에지를 모두 고려하여 빈발도에 대한 가중치를 적용하여 발견한다.

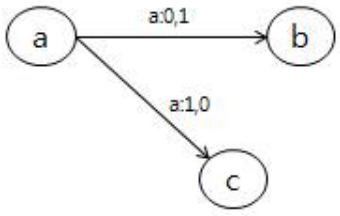


그림 4. 항목 a의 그래프 에
Fig. 4. Example of item a

표 3은 그림 3의 그래프에 대한 후보 항목집합의 가중치와 빈발여부를 나타낸 것이다.

표 3. 항목집합의 가중치와 빈발도
Table 3. Weight and frequency of itemsets

pattern	Pwgt	frequent
a	0.6+0.3=0.9	false
b	0.5*2 =1.0	false
bc	(0.5+0.8)/2 = 1.3	true
c	0.8*3+0.9*2=4.2	true
d	0.7+0.7*2=2.1	true
cd	(0.8+0.7)/2+(0.9+0.7)/2=1.55	true
e	0.8+0.7=1.5	true
ce	(0.8+0.8)/2+(0.9+0.7)/2*2=2.4	true

이 논문에서 제안하는 마이닝 알고리즘은 다음과 같다.

```
Input: window size |W|, batch Bi, min_sup, LMwgt
output: frequent item set

if the number of batch > |W|
   then slide the oldest batch Bi from FSGraph
else
   insert Bi into window set;
for each Bi in |W|
   if the weight sum of each item ≤ LMwgt
     then pruning the item;
end for;
make the FSGraph with pathway label;
for each item in FSGraph
    compute frequent weight of items, Iwgt;
    if Iwgt ≥ min_sup
       add item into frequent item set;
end for;
```

Ⅳ. 실험 및 평가

제안된 방법의 효율성을 평가하기 위해 윈도우8.1 환경에서 Java(JDK1.8)으로 구현하였다. 동적 가중치를 이용하여 빈발패턴의 항목들을 트리로 생성하는 방법(SPrefix-tree)과 이 논문에서 제안하는 그래프로 생성하는 방법(SP-graph)을 비교하였다. 임의로 데이터를 생성하여 10,000개의 트랜잭션으로 실험하였고, 트랜잭션의 평균 항목의 수는 6개, 하나의 배치에는 3개의 트랜잭션으로, 윈도우 사이즈는 2로 하였다. 그리고 하나의 윈도우에 속한 항목들의 평균 중복비율은 2.3이다.

첫 번째 실험은 트리와 그래프에서 생성되는 노드 수를 비교하였다. 노드 수는 메모리에도 영향을 미치고, 수행시간에도 영향을 준다. 그림 5는 생성되는 노드 수를 비교한 결과이다.

실험 결과에서 트리의 노드 수가 그래프의 노드 수보다 상대적으로 많은 차이를 보이는 것을 알 수 있었다. 임계치가 커지면 최소 가중치를 만족하는 항목의 수가 적어 트리와 그래프의 노드 수가 많이 줄어드는 것을 볼 수 있다.

트리는 계층적 구조의 특성이 유지되면서 항목들의 빈발도를 반영해야 하므로 항목을 표현하는 노드 생성의 순서가 구조에 많은 영향을 미친다. 이것은 항목의 노드 생성 순서가 다르면 서브트리를 구별하여 생성해야 하므로 노드 수가 많아지게 된다. 이러한 트리를 좀 더 효율적으로 표현한 것이 FP-tree인데, 이 방법은 빈발도가 높은 항목 순으로 트리의 계층을 만들고 노드 수를 줄이기 위해 다른 서브트리에 같은 항목이 존재하면 연결되어 있음을 표현하였다. 그러나 동적 가중치를 고려하는 빈발항목의 노드 생성 순서는 가중치가 변화하므로 항목

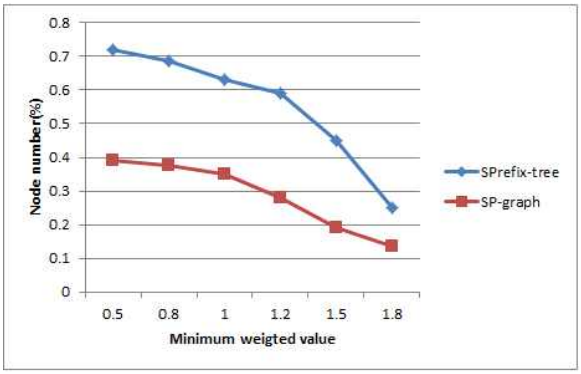


그림 5. 트리와 그래프의 노드 수
Fig. 5. The number of nodes in a tree and graph

의 가중치나 빈발도를 고려하는 노드 생성 순서를 설정하고 유지하는 것에 어려움이 있다. 반면에 그래프는 알파벳순으로 항목의 노드 생성 순서를 정하고 노드간의 연관 관계를 에지로 표현하므로 생성되는 노드 수가 트리보다 상대적으로 적다는 것을 알 수 있다. 이것은 노드의 생성과 유지에 필요한 저장공간과 소요되는 시간은 수행시간에도 영향을 미친다.

두 번째 실험은 트리와 그래프를 이용하여 마이닝이 수행되는 시간을 비교하였고, 실험 결과를 그림 6에서 보여준다. 마이닝에서 일반적으로 최소 가중치가 작을수록 만족하는 항목 수가 많아 수행시간이 많이 소요된다. SPrefix-tree는 최소 가중치가 작을수록 SP-graph보다 더 많은 소요시간을 필요로 하는 것을 알 수 있었다. 이것은 트리에 생성되는 노드 수가 그래프의 노드 수보다 많기 때문에의 트리 노드의 생성과 유지에 소요되는 시간이 더 많이 증가하는 것으로 판단된다.

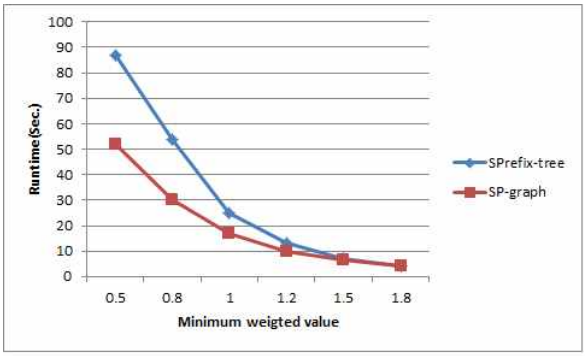


그림 6. 수행시간
Fig. 6. Running time

Ⅴ. 결 론

이 논문에서는 사용자의 상황이나 관심도의 변화를 고려하기 위하여 서비스에 동적 가중치를 부여하여 최적의 서비스를 탐색하는 방법을 제안하였다. 기존의 서비스 이력 데이터를 기준으로 시기나 연령에 따른 서비스 규칙을 미리 저장하고, 실시간으로 변화하는 서비스의 관심도를 고려한 유용한 서비스를 제공하기 위하여, 최신의 서비스 규칙을 지속적으로 탐사하여 새로운 규칙을 발견하여 추가하는 것이다. 이를 위해 빈발 패턴 마이닝에서 많이 사용되는 트리구조가 아닌 그래프를 이용하여 생성되는 노드 수를 줄이고 수행속도를 향상시키는 방법을 제안하였다. 제안한 방법은 트리구조와의 마이닝 수행시간의 비교 실험에서 우수한 결과를 보였다.

감사의 글

이 논문은 2017년도 남서울대학교 학술연구비 지원에 의해 연구되었음.

참고문헌

[1] C. F. Ahmed, S. K. Tanbeer and B. S. Jeong, “Efficient Mining of Weighted Frequent Patterns Over Data Streams,” International Conference on High Performance Computing and Communications, pp.400-406, 2009.

[2] Y. Kim, W. Kim, U. Kim, “Mining Frequent Itemsets with Normalized Weight in Continuous Data Streams,” Journal of Information Processing Systems, Vol.6, No.1, 2010.

[3] G.Chen, X. Wu and X. Zhu, “Mining Sequential Patterns Across Data Streams,” Computer Science Technical Report(CS-05-04), 2005.

[4] C. K. S. Leung Q. I. Khan, T. Hoque, "CanTree:A Tree Structure for Efficient Incremental Mining of Frequent Pattern Sets," In proc. ICDM 2005.

[5] C. K. S. Leung Q. I. Khan, "DSTree:A Tree Structure for the Mining of Frequent Sets from Data Streams," In proc. ICDM 2006.

[6] J. Han, J. Pei, Y. Yin, "Mining Frequent Patterns without Candidate Generation," In Proc. of ACM SIGMOD International Conference on the Management of Data, 2000.

[7] J. Pei, J. Han, B. M. Asl, 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, 2004.

[8] P. Chouksey, R.S. Thakur, R.C. Jain, “Exploring Interesting Multi-level patterns using graph based approach, In Proc. 4th Indian International Conference on Artificial Intelligence, pp 377-387, 2009.

[9] V. Tiwari, V. Tiwari, S. Gupta and R. Tiwari, “Association for Mining: A Graph Based Approach for Mining Frequent Itemsets,” IEEE International Conference on Networking

and Information Technology, 2010.
[10] A. Choubey, R. Patel, J.L. Rana, "Graph based new approach for frequent pattern mining," International Journal of Computer Science & Information Technology ,Vol 4, No 1, pp221-235, 2012.
[11] D. Han, D. Kim, J. Kim, C. Na and B.Hwang, “A Method for Mining Interval Event Association Rules from a Set of Events Having Time Property,” Journal of Korea Information Processing Society, Vol.16-D, No.2, pp.185-190, 2009.
[12] Y.Lee, J. Lee, D. Chai, B. Hwang and K. Ryu, “Mining Temporal Interval Relational Rules from Temporal Data,” The Journal of Systems and Software, Vol. 82, No.1, pp.155-167, 2009.
[13] S. Laxman, P. S. Sastry and K. Unnikrishnan, “Discovering Frequent Generalized Episodes where Events Persist for Different Durations,” IEEE Transactions on Knowledge and Data Engineering, Vol.19, No.9, pp.1188-1201, 2007.
[14] Unil Yun, “Efficient mining of weighted interesting patterns with a strong weight and/or support affinity,” Journal of Information Sciences, Vol.177(17), pp3477-3499, 2007.
[15] Zhongliang Li1, Tengfei Zhou1, Haoran Zhang1, Guocai Yang, "IWFPM: Interested Weighted Frequent Pattern Mining with Multiple Supports," Journal of Software, Vol.10(1) pp.9-19, 2015.
[16] Jeong Hee Hwang, Mining Association Rule on Service Data using Frequency and Weight," Journal of Digital Contents Society, Vol.17(2), 2016.

황정희(Jeong-Hee Hwang)

2001년 :충북대학교 전자계산학과 (이학석사)
2005년 :충북대학교 전자계산학과 (이학박사)

2001년～2006년: 정우시스템(주) 연구소장
2006년～현 재 : 남서울대학교 컴퓨터학과 교수
※관심분야 : 유비쿼터스 컴퓨팅(Ubiquitous Computing), 데이터 마이닝(Data Mining), 빅 데이터(Big Data) 등