Journal of Korea Game Society 2018 Apr; 18(2): 121-130
http://dx.doi.org/10.7583/JKGS.2018.18.2.121

데이터마이닝활용

빅데이터에 대한 Completeness를 이용한 빈발 패턴 마이닝

박인규
중부대학교 게임소프트웨어학과
fip2441g@gmail.com

Frequent Pattern Mining By using a Completeness for BigData

In-Kyu Park
Dept. of Game Software, College of Engineering Joongbu University

요 약

대부분의 빈발 패턴은 패턴이 트랜잭션 데이터베이스에 나타나는 support를 패턴 interestingness의 핵심 척도로 다루어 왔으나 패턴의 횟수는 패턴의 completeness가 가지는 정보를 최대치로 가정하고 있다. 그러나 실제적으로는 임의의 패턴 X의 completeness는 트랜잭션에서 서로 다르게 나타나기 마련이다. 따라서 패턴이 가지는 정보의 손실을 줄이기 위해서는 가중치에 의한 support와 completeness에 의한 유용한 패턴 마이닝을 고려하여야 한다. 즉, 높은 completeness율을 갖는 패턴은 더 높은 recall로 이어질 수 있고 높은 빈도수를 갖는 패턴은 보다 높은 정밀도로 이어진다. 본 논문에서는 동적인 항목들의 가중치에 따른 적응된 support와 completeness를 고려하는 WSCFPM 패턴 마이닝 알고리즘을 제안한다. 제안한 방법은 모노톤 또는 반 모노톤 속성이 가중치에 의한 support와 completeness에 영향을 미치지 않기 때문에 탐색과정을 줄일 수 있다. 실험결과를 통하여 제안된 알고리즘이 효과적이며 확장성이 좋은 것임을 보인다.

ABSTRACT

Most of those studies use frequency, the number of times a pattern appears in a transaction database, as the key measure for pattern interestingness. It prerequisites that any interesting pattern should occupy a maximum portion of the transactions it appears. But in our real world scenarios the completeness of any pattern is more likely to become various in transactions. Hence, we should also consider the problem of finding the qualified patterns with the significant values of the weighted support by completeness in order to reduce the loss of information within any pattern in transaction. In these pattern recommendation applications, patterns with higher completeness may lead to higher recall while patterns with higher completeness may lead to higher recall while patterns with higher frequency lead to higher precision. In this paper, we propose a measure of weighted support and completeness and an algorithm WSCFPM(weigted support and completeness frequent pattern mining). Our algorithm handles the invalidation of the monotone or anti-monotone property which does not hold on completeness. Extensive performance analysis show that our algorithm is very efficient and scalable for word pattern mining.

Keywords : Frequent Pattern Mining(빈발패턴 마이닝), Weighted Support(가중치 지지도), Completeness(점유), Anti-monotone Property(비단조성)

Received: Dec. 18. 2018 Revised: Feb. 10. 2018
Accepted: Feb. 27. 2018
Corresponding Author: In-Kyu Park(Joongbu University)
E-mail: fip2441g@gmail.com

ISSN: 1598-4540 / eISSN: 2287-8211

Ⓒ The Korea Game Society. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.otg/licenses/by-nc/3.0), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. 서 론

데이터 마이닝은 데이터베이스의 트랜잭션(Transaction)등의 데이터에서 빈번하게 발생하는 아이템들의 집합을 찾아내는 기법이다[1,2]. 초기의 빈발 패턴알고리즘은 FP-Growth방법을 활용하여 여러 가지 문제점을 극복하여 성능면에서 많은 개선을 가져오게 되었다. 그러나 항목간의 중요도가 서로 상이한 환경서 이들은 각 항목의 중요도를 모두 같은 값으로 고려하고 있고 트랜잭션내에서의 항목의 비율은 고려하지 않고 있다[2]. 따라서 빈발패턴의 추출에서 패턴의 중요도에 대한 시변성(Time-Variance)과 더불어 항목이 각각의 트랜잭션에서의 비율에 관한 많은 연구가 진행되어 왔다[3,4,5]. 일반적으로 임의의 항목이 가지는 의미는 시간에 따라서 변하기 마련이고 각 트랜잭션에서의 임의의 각 항목이 차지하는 비율은 빈발패턴의 결정에 중요한 정보를 제공한다. 실제적으로 많은 사례에서 보면 항목이 가지는 의미는 시간에 종속적인 경우가 존재한다[6,7,8,9,10]. 또한 Anti-monotone성질을 만족하기 위해서는 전역임계값이나 국부적임계값을 최대치 임계값으로 설정하고 있기 때문에 항목간의 가중치를 고려할 경우에 정보의 손실이 발생할 수 있다.

이에 대한 일환으로 본 논문에서는 임계값의 최대치를 설정하지 않고 항목의 가중치에 의한 support와 트랜잭션에서의 항목비율을 이용하기 때문에 Anti-monotone 성질을 무효화하는 가중치에 따른 지지도와 트랜잭션의 완성도 기반 빈발패턴(ACWFP: Weighted Support and Completeness Frequent Pattern) 알고리즘을 제안한다. 트랜잭션에서의 항목이 가지는 고유한 정보를 추출할 수 있고 단 한 번의 데이터베이스 스캔을 통하여 처리되므로 스트림 데이터 환경과 같은 실시간 처리가 필요한 환경에서 적용이 가능하다.

2. 가중치 빈발 패턴 마이닝

I = {i1, i2, …, in}은 n개의 항목 집합으로 항목집합 X ⊂ I는 I의 항목의 부분집합이고 트랜잭션 Ti ⊂ I는 항목 집합이다. 데이터베이스 D = {T1, T2, …, Tn}은 일련의 트랜잭션으로 구성되어 있다. S(X)로 표시된 X의 지원은 항목 집합 X를 포함하는 트랜잭션 집합으로 S(X) = {Ti ∈ D, i = 1, …, m | X ⊆ Ti}. Freq(X)로 정의 된 X의 빈도는 X가 들어있는 트랜잭션의 수이다. 즉, Freq(X) = Card(S(X)). 빈도는 예를 들어 X 항목을 함께 구입 한 횟수에 해당한다. Length(X)로 표시된 X의 길이는 X에 있는 항목의 수이다. L(X) = Card(X). 즉, L(X)는 S(X)의 트랜잭션간에 공유되는 항목의 수이다. 트랜잭션간에 공유되는 항목이 많을수록 트랜잭션이 더 유사하다. 임의의 항목이 동시에 발생하는 패턴을 빈발 패턴이라고 한다. 자주 사용되는 항목 집합은 데이터베이스에서 자주 발생하는 항목 집합이다. 이러한 빈발패턴은 사용자가 정의한 최소 빈도 임계 값 Fmin에 따라 달라진다. 즉, Freq(X)> = Fmin이면 빈발패턴이다. Anti-monotone의 성질에 의하여 빈발항목의 집합을 구성하는 어떠한 집합의 부분집합도 빈발항목이어야 한다. 임계치를 이용하여 이러한 조건을 만족하는 빈발항목의 탐사여부를 결정한다. 데이터베이스에 존재하는 하나의 패턴에 대한 가중치는 해당패턴의 중요도를 나타낸다. 가중치를 이용하여 데이터베이스에 존재하는 빈발 항목의 마이닝은 항목의 가중치가 변경 될 때와 데이터베이스가 변경 될 때 더욱 유용하며 항목의 지지도와 가중치를 곱한 가중치 지지도(Weighted Support)가 사용되어 진다.

가중치 빈발패턴 알고리즘으로 초기에 Apriori 알고리즘 기반의 MINWAL, WARM와 WAR는 여러 번의 데이터베이스의 스캔으로 속도에 문제가 있다. MINWAL은 k-지지도를 이용하여 Anti-monotone성질을 유지하지만 빈발 항목의 정확도에서 효율성이 떨어진다. WARM에서 “ab” 항

목의 가중치 지지도는 모든 트랜잭션의 가중치에 대한 “a” 와 “b”를 포함하는 트랜잭션의 가중치의 비율로 적용하여 하향폐쇄에 위배되는 문제를 해결하였다. WFIM은 FP-트리 기반의 초기의 가중치 빈발 패턴 마이닝으로 최소 가중치와 가중치의 범위를 가지고 패턴을 탐색한다. WIP에서는 가중치 신뢰도(weight confidence)를 정의하고 가중치의 범위를 이용하여 가중치의 그룹을 결정하고 h-신뢰도를 이용하여 우수한 지지도를 가지는 패턴을 탐색되어 진다. 또한 WLPMiner에서는 가중치와 지지도에 대해 제약을 두어 최소의 우수한 빈발 패턴을 탐색하게 된다. 지금까지의 모든 가중치 빈발 마이닝에서의 지지도는 임의의 항목집합이 트랜잭션에 존재하는지에 대한 여부를 이분법적인 논리를 적용하였다. 본 논문에서는 하나의 트랜잭션내에서 항목집합이 가지는 Completeness에 따라서 지지도를 적응시키는 적응된 지지도(Adjusted Support)를 제안한다.

3. 적응 지지도기반 가중치 빈발 패턴 마이닝(WSCFPM)

3.1 제안된 트리의 생성

[정의 1] 패턴 X에 대한 동적 가중치 지지도(WS: Weighted Support)는 식(1)와 같이 정의된다[11].

$$WS(X) = \sum_{i=1}^{N} W(X, i) \times S(X, i) \qquad \text{(eq.1)}$$

여기서 N은 동적인 가중치를 위한 배치의 개수를 나타내고 Weight(X, i)는 X에 속하는 배치의 각 항목들의 평균 가중치에 해당하는 j번째 배치에 있는 X의 가중치이고, Support(X, j)는 j번째 배치에 있는 X의 지지도(빈도수)이다. 예를 들면 [Table 1]에서 첫 번째 배치의 패턴 “bd”의 WS(bd)는 ((0.9 + 0.3) / 2) × 1 = 0.6이고 두 번째, 세 번째 배치의 WS(bd)는 각각 ((0.7 + 0.5) / 2) × 1 = 0.6, ((0.7 + 0.4) / 2) × 0 = 0이 된다. 따라서 전체 WS(bd) = 0.6 + 0.6 + 0 = 1.2가 된다.

[정의 2] 패턴 x의 WS(x)의 값이 주어진 최소 임계값보다 크거나 같을 때 패턴 x를 적응 가중치 빈발 패턴이라 한다. 만일 최소 임계값이 1.2라면 [Table 1]에서 패턴 “bd”는 적응 가중치 빈발 패턴이다.

[정의 3] 본 논문에서는 각각의 t∈TX에 대하여 t의 크기에 대한 X의 크기의 비율을 X의 Completeness는 (eq.2)와 같이 정의한다.

$$Comp(X) = \frac{|X|}{|t|}, t \in T_X \qquad \text{(eq.2)}$$

패턴 p에 대한 Completeness에 의한 적응 가중치 지지도(WSC: Weighted Support and Completeness)는 (eq.3)과 같이 정의된다.

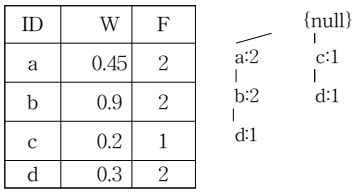
$$WSC(X) = \sum_{i=1}^{N} \frac{Weight_i(X) + Comp_i(X)}{|Sup_i(X)|} \qquad \text{(eq.3)}$$

첫 번째 배치에서 항목 “bd”의 WSC는 ((0.9 + 0.3) / 2 + 2 / 3) / (1 / 3) = 0.422이고, 두 번째, 세 번째 배치에서는 각각 ((0.7 + 0.5) / 2 + 2 / 3) × 1 / 3 = 0.422, (((0.3 + 0.4) / 2) × 0 × 0) = 0이 된다. 따라서 전체적인 WSC(bd) = 0.422 + 0.422 + 0 = 0.844 된다. 또한 항목 “b”에서 WSC는 첫 번째의 배치에서는 (0.9 + (0.33 × 0.5) / 2) × (2 / 3) = 0.877이고 두 번째 배치에서는 (0.7 + (1.0 + 0.33) / 2) × (2 / 3) = 0.91과 세 번째의 배치에서는 (0.3 + 0 × 0) = 0이 된다.

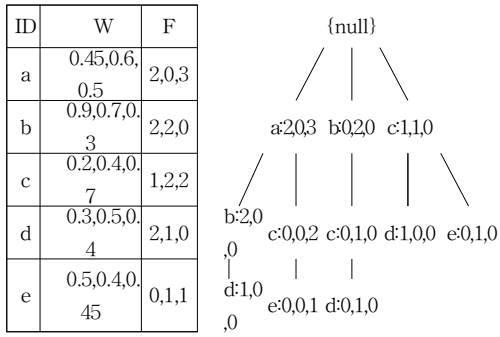
[Table 1] An Example of Weighted Transactions

Batch	TID	Trans.	Weight				
1st	T_1 T_2 T_3	a b d c d a b	a 0.45	b 0.9	c 0.2	d 0.3	e 0.5
2nd	T_4 T_5 T_6	b b c d c e	a 0.6	b 0.7	c 0.4	d 0.5	e 0.4
3rd	T_7 T_8 T9	a c e a a c	a 0.5	b 0.3	c 0.7	d 0.4	e 0.45

따라서 전체적인 WSC(b) = 0.877 + 0.91 + 0 = 1.787이 된다. 항목 “d”에서 WSC는 각각 (0.3 + (0.33 + 0.5) / 2) × (2 / 3) = 0.483이고 (0.5 + 2 / 3) × (1 / 3) = 0.388와 세 번째의 배치에서는 0이 된다. 따라서 전체적인 WSC(d) = 0.483 + 0.388 + 0 = 0.871이 된다.이 절에서는 적응 가중치를 갖는 항목들로 이루어진 트랜잭션들의 내용을 저장하는 WSC-트리의 구조 및 생성 방법을 설명한다. WSC-트리는 헤더 테이블을 포함하고 있으며 루트노드와 일련의 트리로 구성된다. 헤더 테이블에는 항목 ID, 가중치와 빈도수가 포함되어 있고 각 배치의 트랜잭션들이 삽입되어 WSC-트리가 사전식으로 구성되어 진다. 또한 항목들의 배치에 따른 빈도수 정보를 유지하기 위한 노드를 운영한다. 예를 들어 세 개의 배체에서 두 번째 배치에서 테일 노드 “b” 가 처음으로 존재하면 노드의 구성은 b:0,1,0 이 된다. [Fig. 1]은 [Table 2]의 데이터베이스를 이용하여 구성한 트리의 모습을 보여준다.



(a)After inserting 1st Batch



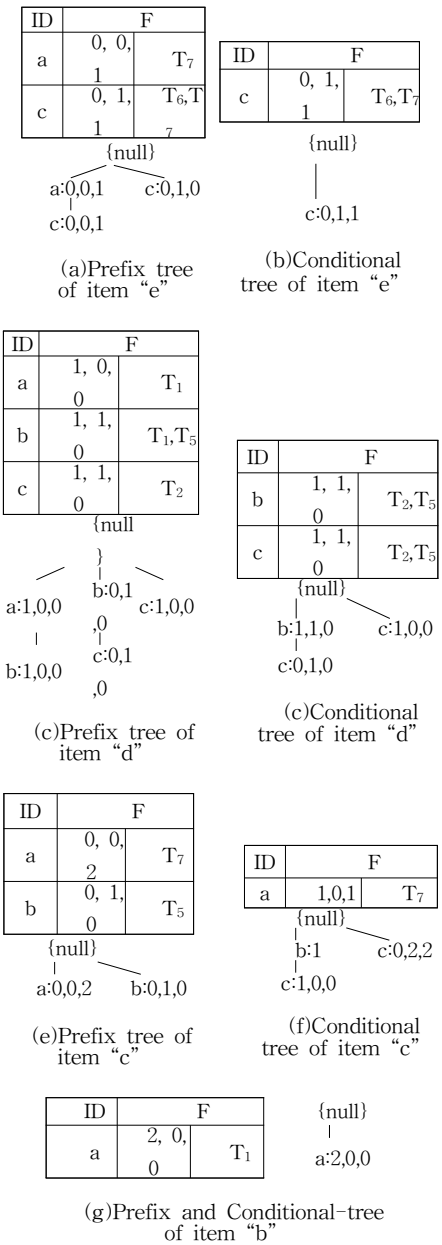
(b)After inserting 3rd Batch

[Fig. 1] Constructed WSC-Tree

3.2 제안된 트리 마이닝

WSCFP-growth 알고리즘은 FP-growth 알고리즘처럼 트리의 상향 탐색을 통하여 수행된다. [정의 3]에서 언급하였던 것처럼 항목집합의 동적 가중치와 적응된 지지도 빈도수를 이용하여 Anti-monotone 성질을 만족하지 않는다. 더욱이 Completeness를 고려할 경우에도 Anti-monotone 성질을 만족하지 않는다. [Table 2]의 데이터베이스와 이를 이용하여 구성한 [Fig. 1]의 트리를 가지고 최소 임계값이 1.4일 경우 마이닝하는 과정을 살펴보면 우선 Completeness를 고려한 적응 가중치 지지도는 <a:1.62, b:1.78, c:1.51, d:0.75, e:0.561>가 된다. 또한 제안된 방법에서는 최소 임계값을 고려하지 않기 때문에 모든 항목이 빈발 패턴의 가능성이 있다. 따라서 모든 항목에 대하여 트리의 상향탐색을 통하여 적응된 가중치 빈발 패턴을 찾게 된다. 먼저 헤더 테이블의 가장 밑에 있는 항목 “e”에 대한 Prefix 트리를 구성하기 위해서는 [Fig. 1]의 트리에서 “e” 항목을 포함하는 가지들을 빈도수를 고려하여 추출하면 <ac:001,

c:010>이 된다. 여기서 불필요한 항목을 추출하기 위하여 항목의 가중치 빈도수는 <a:001, c:011>가 된다. 따라서 이를 이용하여 적응 가중치 빈도수를 구하면 a는 세 번째 배치에 대하여 a = (0.5 + 1 /3) × 1 / 3 = 0.28이고 c는 두 번째와 세 번째 배치에 대하여 c = (0.4 + (1 / 2)) × 1 / 3 + (0.7 + (1 / 3)) × 1 / 3 = 0.64가 된다. 결국, <a:0.28, c:0.64> 이다. 따라서 최소 임계치 0.4인 경우에 항목 a는 Conditional 트리에서 제거된다. 이 과정에서 “e”와 “ec”가 후보 패턴으로 추출된다. 지금까지의 추출된 패턴은 <a, b, c, d, e, ce> 이다. 같은 방법으로 항목 “d”에 대한 Prefix 트리와 Conditional 트리를 구성하면 [Fig. 2]와 같다. “d” 항목을 포함하는 경로들의 빈도수를 고려하면 <ab:100, bc:010, c:100>이 된다. 단일 항목의 가중치 빈도수는 <a:100, b:110, c:110>가 되어 이를 이용하여 적응 가중치 빈도수를 구하면 <a:0.26, b: 0.75, c:0.47>이다. 따라서 a는 제거되고 “bd” 와 “cd”가 후보 패턴으로 추출된다. 지금까지의 추출된 패턴은 <a, b, c, d, e, ce, bd, cd> 이다.



[Fig. 2] Mining Process

[Table 2] Frequent Pattern of Weighted Support and Completeness

Candidate Patterns	WASC calculation	Result
ce:0,1,1	((0.4+0.4) / 2) + 1) × 1/3 + ((0.7+0.45) / 2 + 2/3) × 1/3 = 0.87	Pass
e:0,1,1	(0.4 + 1/2) × 1/3 + (0.45 + 1/3) × 1/3 = 0.56	Pruned
cd:1,1,0	((0.2+0.3) / 2+1) × 1/3 + ((0.4+0.4) / 2 + 2/3) × 1/3 = 0.772	Pruned
bd:1,1,0	((0.9+0.3) / 2 + 2/3) × 1/3 + ((0.7+0.5) / 2 + 2/3) × 1/3 = 0.84	Pass
d:2,1,0	(0.3 + (1/3+1/2) / 2) × 2/3 + (0.5 + 1/3) × 1/3 = 0.74	Pruned
ac:0,0,2	((0.5+0.7) / 2 + (2/3+1) / 2) × 2/3 = 1.43	Pass
c:1,2,2	(0.2+1/2) × 1/3 + (0.4 + (1/3+1/2) / 2)× 2/3 + (0.7 + (1/3+1/2) / 2) × 2/3 = 1.51	Pass
ab:2,0,0	((0.45+0.9) / 2 + (2/3+1) / 2) × 2/3 = 1.01	Pass
b:2,2,0	(0.9 + (1/3+1/2) / 2) × 2/3 + (0.7 + (1+1/2) / 2) × 2/3 = 1.78	Pass
a:2,0,3	(0.45 + (1/3+1/2) / 2) × 2/3 + (0.5 + (1/3+1/2) / 2) × 1 = 1.62	Pass

항목 “c”에 대한 Prefix 트리와 Conditional 트리를 구성하면 [Fig. 3]과 같다. “c”항목을 포함하는 가지들을 빈도수를 고려하여 추출하면 <a:002, b:010>이 된다. 이에 대한 적응 가중치 빈도수를 구하면 <a:0.61, b:0.34>이다. 따라서 b는 제거되고 “ac” 가 후보 패턴으로 최종적인 적응 가중치 빈발패턴은 <a, b, c, d, e, ce, bd, cd, ac>가 된다. 항목 “b”에 대한 Prefix 트리와 Conditional 트리를 구성하면 [Fig. 3]과 같다. “b” 항목을 포함하는 가지들을 빈도수를 고려하여 추출하면 <a:2,0,0>이 된다. 이에 대한 적응 가중치 빈도수를 구하면 <a:0.87> 이다. 따라서 “ab”가 후보 패턴으로 최종적인 적응 가중치 빈발패턴은 <a, b, c, d, e, ce, bd, cd, ac, ab>가 된다. 이와 같은 방식으로 모든 후보 패턴은 [Table 3]의 첫 번째 열의 후보 패턴과 같고 실제 가중치와 Completeness를 패턴의 Support로 적응시켜 계산하여 임계값과 비교하면 [Table 3]의 오른쪽 열과 같이 최종적인 빈발 패턴을 얻게 된다. 지금까지 설명한 마이닝 과정을 알고리즘으로 기술하면 아래와 같다.

```
Procedure ASCFP-tree construction
Input: A transaction DB and a minimum threshold (∂)
Output: Its frequent pattern tree, ASCFP-tree
  scan transaction database DB once
  collect frequent items FI and their supports
  sort FI in support descending order as L
  for each transaction in DB
    select and sort FI in Trans by the order of L
    insert (p|P, T) into the root of Prefix tree, T
    update T.header
  end
  for each item a_i from the bottom of T.header
    if total_frequency(a_i)*GMAXW)> ∂
      create a conditional tree Tree_i for item a_i
      call mining(Tree_i, a_i)
    end
  end
```

```
Procedure ASCFPgrowth
Input: A conditional Tree(T) with weights and threshold
Output: The complete frequent patterns corresponding to T
  mining(T_i, a_i)
  {
    create the conditional tree, ASC of a
    by deleting each item d_i from T having
    adjusted support completeness(d_i)< ∂
    for each item β_i in the header table of ASC
      call Test_Candidate(aβ)
      create ASC-tree T_β for itemset aβ_i
      call mining(T_β, aβ)
  }
  Test_Candidate(X)
  {
    let adjusted support completeness of X is ASC_x
    set ASC_x= 0
    for each batch B_i
      ASC_x= ASC_x+adjusted support completeness(X_β_i)
    if ASC_x >= ∂
      add X in FP ist of adjusted support completeness
  }
```

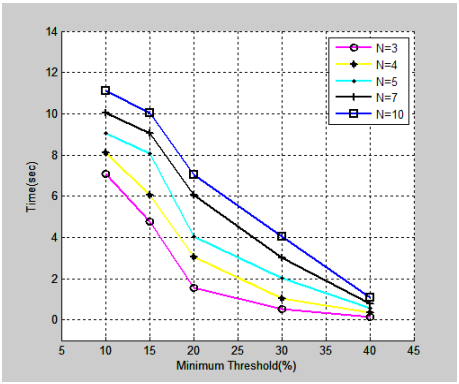
4. 실험 결과 및 분석

제안된 알고리즘의 성능적인 특성을 알아보기 위하여 몇 가지 실제 데이터 세트가 적용되었다. 모든 실험은 3.6GHz 8 Core CPU 및 16GB 메모리가 있는 Windows 10에서 Java로 구현하여 수행되었다. 이러한 데이터 세트와 관련된 정보가 [Table 3]에 나타나 있다. 실행시간은 CPU만의 시간이 아니고 데이터베이스에서 트리의 생성과 트리에 대한 마이닝 시간을 고려하여 측정한 시간이다.

[Table 3] The Characteristics of DataSets

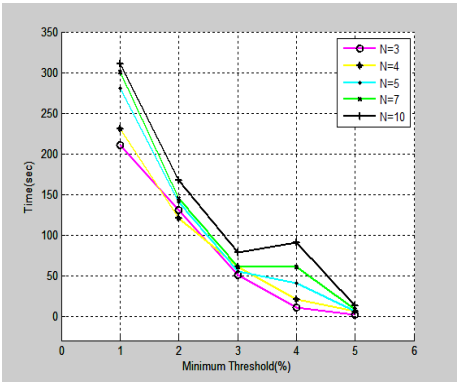
Dataset	#Trans	#Items	Avg. trans. size
Mushroom	8,124	119	23
T1024D100K	100,000	942	10.2

[Fig. 3]은 조밀한 데이터 집합인 Mushroom에 대하여 최소 지지도 값이 변화할 때의 WSCFP 알고리즘의 실행 시간을 나타낸다. Mushroom은 항목들의 약간의 조밀도를 가지는 데이터로서 트랜잭션의 길이는 긴 편이다. 이러한 경우에는 항목이 트랜잭션을 차지하는 비율이 줄어들게 된다.



[Fig 3] Mushroom's Mining Time

반면에 T1024D100K는 항목의 수가 많고 트랜잭션의 길이가 상대적으로 짧아 조밀하지 않은 특성을 가지고 있다. 따라서 트랜잭션을 차지하는 비율이 높아지게 된다. 결국 가중치에 의하여 조정된 Support의 정보와 항목의 트랜잭션에 대한 Completeness정보를 고려하여 빈발 패턴의 정확성에 기여하게 된다.



[Fig. 4] T1024D100K's Mining Time

조밀한 데이터의 동적 가중치에 의한 지지도만을 고려한 경우에는 임계값이 상대적으로 큰 구간에서 속도 변화를 보이고 있었지만[12,13] Completeness를 고려한 경우에는 트랜잭션의 길이가 길기 때문에 항목의 비율이 상대적으로 작아지기 때문에 기존의 경우보다 속도 변화에 약간의 상쇄를 가진다. 조밀하지 않은 데이터 집합의 경우에도 지지도만을 고려한 경우에는 임계값이 작은 변화에도 많은 속도 변화를 보이고 있는데 Completeness를 고려하면 속도변화에 선형성에 진동이 발생하고 있다. 그 이유는 임계값에 따른 후보 패턴의 수가 각기 다르고 트랜잭션의 길이에 따라서 항목이 차지하는 비율과 관련이 있는 것으로 보인다.

5. 결 론

본 논문에서는 빈발 패턴의 효율적인 탐색을 위한 척도의 구성에서 트랜잭션에 대하여 항목이 차지하는 비율을 고려하고 동적 가중치에 의한 항목의 지지도의 크기를 결합한 척도를 제안하였다. 제

안된 척도는 항목간의 Anti-monotone성질을 무효화하여 트리의 패턴 생성과정에서 발생하는 많은 불필요한 후보항목 생성 과정을 최소화하여 탐색시간을 줄이고 정확성을 기대할 수 있었다. 제안한 WSCFPM 알고리즘은 배치별로 항목들의 가중치에 의한 적응된 지지도와 트랜잭션을 차지하는 항목의 비율을 고려하여 적응 가중치 빈발 패턴을 찾아 낼 수 있었다. 조밀한 데이터 집합에 대해서는 배치의 수에 따라 약간의 속도의 차이를 보이고 있었다. 이는 데이터의 조밀한 특성과 배치에 따른 속도변화에서 약간의 차이가 있지만 후보패턴의 트랜잭션에 대한 구성 정보를 이용하여 패턴의 정확성에 일조할 수 있을 것으로 보인다. 또한 유저들이 게임을 플레이하며 만들어 내는 데이터에서 유저들의 유형과 행동 패턴을 추적하고 이러한 데이터들을 통하여 인사이트를 도출하는데 일조할 수가 있을 것으로 사료된다.

ACKNOWLEDGEMENTS

This paper was supported by Joongbu University Research & Development Fund, in 2017.

REFERENCES

[1] S. Y. Hong, “New Authentication Methods based on User’s Behavior Big Data Analysis on Cloud”, Convergence Society for SMB, Vol. 6, No. 4, pp.31-36, 2016.

[2] J. E. Shin, B. H. Jeong, D. H. Lim, “BigData Distribution System using RHadoop”, Society of Data Information Science, Vol. 36, No. 5, pp. 1155~ 1166, 2015.

[3] R. Agrawal, R. Srikant, “Fast Algorithm for Mining Association Rules”, In: 20 th Int. Conf. on Very Large Data Bases, pp. 487~499, 1994.

[4] C. H. Cai, A. W. C. Fu, C. H. Cheng, W. W. Kwong, “Mining Association rules with weighted items”, In Proceedings of Intl. Database Engineering and Applications Symposium (IDEAS 1988) , Cardiff, Wales, UK, July pp. 68~77, 1998.

[5] F. Tao, “Weighted association rule Mining using Weighted Support and Significant Framework”, In:9 th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining“, pp. 661~666, 2003.

[6] W. Wang, J. Yang, P. S. Yu, “WAR: Weighted Association Rules Item Intensities”, Knowledge Information and Systems, No. 6, pp. 203~229, 2003.

[7] U. Yun, J. J. Leggett, “WFIM: Weighted Frequent Itemset Mining with a wieght range and a minimum weight”, Society for Industrial and Applied Maathematics, Proceedings of the 2005 SIAM International Conference on Data Mining, pp.636~640, 2005.

[8] U. Yun, “Efficient Mining of Weighted Interesting Patterns with A Strong Weight and/or Support Affinity”, Information Sciences, Vol. 177, pp. 3477~ 3499, 2007.

[9] U. Yun, “An Efficient Mining of Weighted Frequent Patterns with Length Decreasing Support Constraints”, Knowlewdge-Based Systems, Vol. 21, Issue 8, Dec., pp. 741~752, 2008.

[10] S. Zhang, C. Zhang, X. Yan, “Post-Mining: Maintenance of Association Rules by Weighting”, Information Systems, Vol. 23, pp. 691~707, 2003.

[11] H. L. Nguyen, “An Efficient Algorithm for Mining Weighted Frequent Itemsets Using Adaptive Weights”, I.J. Intellogent Systems and Appillcations, Vol. 11, pp. 41-48, 2015.

[12] C. F. Ahmed, S. K. Tanbeer, B. S. Jeong, “Mining Weighted Frequent Patterns using Adaptive Weightes”, In: Fyfe et al. (Eds.): IDEAL 2008, LNCS 5326, pp. 258~265, 2008.

[13] S. W. Jin, B. C. Kim, I. K. Um and Y. I. Kim, “Prototype Development of a Mobile Baseball Pitching Prediction Game using Data Mining Techinque”, Journal of Advanced Information Technology and Convergence, Vol. 12, No. 02, pp.135~143, 2014.

박인규(Park, In Kyu)

약 력 : 1985 연세대학교 전기과 전자계산기 응용(공학석사)
1997 원광대학교 전자과 마이크로 프로세서 응용(공학박사)
1997- 중부대학교 게임소프트웨어학과 교수

관심분야 : 데이터 마이닝, 소프트웨어컴퓨팅