

항목 발생 간격을 고려한 Temporal 연관규칙

이경원 · 김재련

한양대학교 산업공학과

Temporal Association Rules Based on Item Time Interval

Kyong-Won Lee · Jae-Yeon Kim

Dept. of Industrial Engineering, Hanyang University

In this paper, we present a temporal association rule based on item time intervals. A temporal association rule is an association rule that holds specific time intervals. If we consider itemset in the frequently purchased period, we can discover more significant itemset satisfying minimum support. Because the previous study did not consider the time interval between purchased item, it could find itemset that did not satisfy the minimum support in case some item was frequently purchased in a specific period and rarely or not purchased in other period. Our approach uses interval support which is counted by period with support and confidence in the association rule to discovery large itemset.

Keywords : temporal, association rules, period, data mining

1. 서 론

1.1 연구 배경

오늘날은 많은 양의 데이터가 저장됨에 따라 데이터베이스의 용량이 커지고 있으며, 이를 데이터간의 상호연관성을 잠재적 사용가치가 있는 패턴이나 추세 등의 규칙들을 발견하여 시장전략 수립, 수요예측, 의료진단, 상품진열 등 광범위한 분야에 응용하고 있으며[3][8], 가장 큰 장점은 느낌을 사실로 옮길 수 있다는 것이다.

데이터마이닝의 기법에는 연관규칙(association rules), 순차패턴 (sequential pattern), 군집화(clustering), 신경망(neural networks), 분류(classification), 유전 알고리즘(genetic algorithms)등이 있으며[4], 이중 연관규칙은 대용량의 데이터베이스에서 어떤 사건들이 함께 발생하거나, 또는 하나의 사건이 다른 사건을 암시하는 것과 같은 사건간의 상호관계를 나타내는 문제를 다루고 있다.[7]

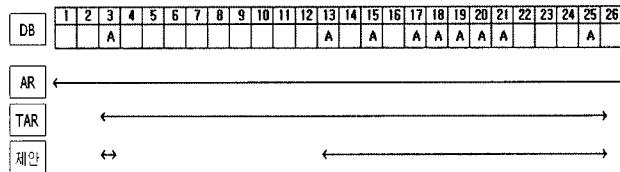
연관규칙은 “ $X \Rightarrow Y$ ”라는 형태로 표시되고 이 의미는 ‘ X ’라는 항목을 포함하는 트랜잭션은 Y 라는 항목도 포

함한다’라는 의미이다. 이러한 연관규칙은 항목이 동시에 발생하는 것을 뜻하는 지지도(support)와 항목간의 조건부확률로 표현되는 신뢰도(confidence)를 기본조건으로 하고 있다. 최소지지도와 최소신뢰도를 만족하는 연관규칙은 전체 데이터베이스를 대상으로 하고 있기 때문에 각각의 규칙이 시간에 대한 정보를 포함하지 않는다. 이에 반해 $X \Rightarrow Y[t_1, t_2]$ 로 표현되는 temporal 연관규칙은 연관규칙과는 달리 규칙에 시간에 대한정보가 들어가 있다. 즉 항목이 처음으로 발생한 시간 t_1 과 마지막으로 발생한 시간 t_2 라는 정보의 표현을 규칙 안에 포함하고 있다. 이는 사용자로 하여금 연관규칙에 대한 시간정보를 갖도록 하기 때문에 항목의 진열 등 마케팅부분에서 유리하게 사용할 수 있다.

1.2 연구 목적

Temporal 연관규칙은 항목이 처음으로 발생한 시점과 마지막으로 발생한 시점을 구분하여 규칙을 생성[2]하기 때문에 항목이 특정 기간 동안 집중적으로 발생한 부분

과 그렇지 않은 부분이 섞여있는 경우에는 연관규칙과 비슷한 결과를 가져올 수 있다. 본 연구에서는 temporal 연관규칙이 갖는 장점인 시간에 대한 정보를 극대화할 수 있도록 항목간의 발생간격을 고려하여 연관규칙을 구하는 방법을 제안하고자 한다.



<그림 1>각 규칙들의 시간기간

위 <그림 1>은 각 방법들의 시간기간을 보여준다. AR은 연관규칙을, TAR은 Temporal 연관규칙을 의미한다. 그림은 데이터베이스에서 항목 A가 발생한 것을 표현한 것이고 화살표는 각 규칙들의 시간기간을 의미한다. 연관규칙은 1부터 26까지의 모든 기간을 대상으로 하여 지지도와 신뢰도를 구하는 반면에 temporal 연관규칙은 항목 A가 처음으로 발생한 곳에서 마지막으로 발생한 곳까지를 대상으로 하고 있다. 세 번째 것은 제안하고자 하는 방법으로 항목 A의 발생이 일정간격이상 떨어져 있는 경우 기간을 별도로 구분하여 2개의 시간기간을 가지고 있는 것을 표현한 것이다.

Temporal 연관규칙에서 지지도는 시간기간에 따라 영향을 많이 받으므로 시간기간은 후보항목집합을 생성시키는 중요한 요소이다. 따라서 항목별로 의미 있는 시간간격만을 사용하여 후보항목집합과 temporal 연관규칙을 생성하는 것이 더 많은 정보와 의미를 갖는 규칙이라 할 수 있다.

1장에서는 연구 배경과 연구 목적을 설명하였고 2장에서는 제안하는 알고리즘을 설명하기 위한 Apriori 알고리즘, Temporal 연관규칙, 순차패턴 등에 대해 설명한다. 3장에서는 제안하는 알고리즘에서 사용되는 정의, 용어와 절차 등을 설명하고 4장에서는 3장에서 설명했던 방법과 절차를 이용하여 예제를 통해 이해를 돋고 5장에서는 본 논문의 결론을 내린다.

2. 기존연구 고찰

2.1 Apriori 알고리즘을 이용한 연관규칙[5]

$I = \{i_1, i_2, \dots\}$ 는 항목(item)이라고 하는 문자들의 집합이라 하고 D 를 트랜잭션들의 집합(set)이라고 한다. 각

트랜잭션 T 는 $T \subseteq I$ 를 만족하는 항목들의 집합이다. 각 T 는 TID 라고 하는 유일한 식별자가 있다. 만약 $T \subseteq I$ 가 성립하면 I 의 부분집합으로 구성되어 있는 X 를 트랜잭션 T 가 포함한다는 것을 뜻한다. 연관규칙은 $X \subseteq I$, $Y \subseteq I$, and $X \cap Y = \emptyset$ 인 상황에서 $X \Rightarrow Y$ 의 형태로 나타낸다. X 를 포함하고 있는 D 의 트랜잭션들의 최소신뢰도 $c\%$ 가 또한 Y 를 포함하고 있다면 $X \Rightarrow Y$ 는 c 의 신뢰도(confidence)를 가지고 있다고 말한다. D 에 있는 트랜잭션의 최소지지도 $s\%$ 가 $X \cup Y$ 를 포함하고 있다면 $X \Rightarrow Y$ 는 s 의 지지도(support)를 가지고 있다고 말한다. 트랜잭션 집합 D 에서 연관규칙을 찾아내는 문제는 사용자가 정의한 최소지지도(minimum support : minsup)와 최소신뢰도(minimum confidence : minconf)보다 큰 지지도와 신뢰도를 갖는 모든 연관규칙을 찾는 것이다. 최소지지도를 만족하는 항목집합을 빈발 항목집합(frequent or large itemset)이라고 부른다.

k 개의 항목으로 이루어진 빈발 항목집합을 빈발 k -항목집합(large k -itemset)이라고 한다. 빈발 k -항목집합들의 집합을 L_k 라 하고 이를 생성하기 위한 후보 항목집합들의 집합을 C_k 라 한다. 알고리즘의 첫 번째 시행에서는 빈발 1-항목집합을 결정하기 위해 데이터베이스를 검색하여 각 항목별로 빈도수를 계산한다. $k(k \geq 2)$ 번째 시행부터는 두 단계로 분할하여 알고리즘이 진행된다. 먼저, $(k-1)$ 번째 검색에서 발견된 빈발항목집합 L_{k-1} 으로 후보항목집합 C_k 를 만든다. 다음으로, 데이터베이스를 검색하여 C_k 에 있는 후보 항목집합의 지지도를 계산한다. C_k 에 있는 후보 항목집합 중에 최소 지지도를 만족시키는 항목만 L_k 에 진입한다. 이러한 시행은 L_k 가 더 이상 발견되지 않을 때까지 반복한다. Apriori에서 가장 중요한 부분인 apriori-gen 함수[1]는 join단계와 prune단계로 구성되어 있다. <표 1>에는 연관규칙을 찾는 예제 데이터베이스가 주어져 있고, Apriori 알고리즘을 이용해 모든 후보 항목집합과 빈발 항목집합을 찾는 전 과정이 <그림 2>에 나타나 있다.

<표 1> Apriori 예제 데이터베이스

TID	항 목
100	A C D
200	B C E
300	A B C E
400	B C

<표 1>은 4개의 트랜잭션과 5개의 항목으로 구성되어 있다. 최소지지도는 50%로 하여 Apriori 알고리즘을 적용한다.

C_1	L_1
항목	지지도
{A}	2
{B}	3
{C}	4
{D}	1
{E}	2

C_2	L_2
항목	지지도
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	3
{B, E}	2
{C, E}	2

C_3	L_3
항목	지지도
{B, C, E}	2

<그림 2> Apriori알고리즘 실행 예

위의 예제를 보면, 첫 번째 데이터베이스 검색 시에 각 항목의 지지도를 계산하고 그 중에서 최소지지도를 만족하는 항목만을 L_1 으로 진입시킨다. Join단계와 Prune 단계를 거쳐 C_2 를 만든다. 다시 데이터베이스를 검색하여 C_2 의 지지도를 계산하여 역시 최소지지도를 만족시키는 후보항목집합만을 가지고 L_2 를 만든다. C_3 를 보면 join방법을 이해할 수 있다. L_2 에서 만들 수 있는 모든 경우의 수대로 후보 항목집합을 만드는 것이 아니고 각 트랜잭션의 앞부분이 같은 {B, C}, {B, E}를 가지고 {B, C, E}라는 후보 항목을 만든 것을 볼 수 있다. 이런 과정을 계속 반복하여 L_3 , L_4 , L_5 ,...를 만든다. 이 예제의 경우 L_4 에서 공집합이 되므로 알고리즘을 종료한다. 최종적으로 사용자가 얻을 수 있는 빈발 항목집합은 $L_1 = \{ \{A\}, \{B\}, \{C\}, \{E\} \}$, $L_2 = \{ \{A,C\}, \{B,C\}, \{B,E\}, \{C,E\} \}$ 그리고 $L_3 = \{ \{B,C,E\} \}$ 이다.

Apriori알고리즘을 설명한 이유는 앞에서도 밝혔듯이 본 논문의 기초가 되기 때문이다. 또한 여기서 설명한 용어와 기호들은 본문에서도 똑같이 사용되며 같은 의미로 사용된다.

2.2 Temporal 연관규칙[2]

$T = \{ \dots t_0, t_1, t_2, \dots \}$ 은 시간의 집합이고, $R = \{ A_1, A_2, \dots \}$ 은 항목의 집합이다. 각각의 트랜잭션 s 는 timestamp t 를 갖고, 각각의 항목은 시간기간 $[t_a, t_b]$ 를 갖는다. 각각

의 트랜잭션 s 는 $s \subseteq R$ 이다. 이때 temporal 연관규칙은 $X \Rightarrow Y[t_1, t_2]$ 의 형태로 표시되고, 항목이 후보 항목집합이 되기 위해서는 항목의 최소 시간기간을 의미하는 temporal 지지도를 만족시켜야 한다. 항목 i 의 시간기간을 $[t_a, t_b]$ 항목 j 의 시간기간을 $[t_c, t_d]$ 이라고 k 가 2이상인 경우에는 항목 i 와 j 의 시간기간 $[t_l, t_2]$ 는 $t_l = \max[t_a, t_c]$, $t_2 = \min[t_b, t_d]$ 이다. 연관규칙에서와 마찬가지로 최소 지지도 이상을 갖는 항목집합을 빈발 항목집합이라 한다. 신뢰도와 지지도는 항목들이 가지고 있는 시간기간 내에서 구한다. k 개의 항목들로 이루어진 빈발 항목집합을 L_k 라 하고, 이를 생성하기 위한 후보 $k-1$ 항목집합들의 집합을 C_{k-1} (잠재적 빈발 항목집합)라 한다.

<표 2> temporal 연관규칙 데이터베이스

TID	항목	t
s1	A C F H I	1
s2	A B C G	2
s3	B C D G I	3
s4	A C I	4
s5	C D E H I	5
s6	A D F G	6

예를 들면 <표 2>에서 항목 A는 시간기간 [1, 6]을 항목 C는 시간기간 [1, 5]를 갖는다. 두 항목을 함께 고려하면 항목 [A, C]는 시간기간 [1, 5]를 갖는다.

Temporal 연관규칙 문제는 사용자가 지정한 temporal 지지도를 후보 항목으로 하여 최소지지도(minsup)와 최소신뢰도(minconf) 이상의 지지도와 신뢰도를 갖는 모든 빈발 항목집합들과 연관규칙을 생성시키는 문제이다. 여기에서 설명된 기호와 방식은 본 논문에서도 같은 방식으로 사용된다.

2.3 순차패턴

순차패턴 문제는 데이터를 시간적으로 분석한다는 의미에서 연관규칙문제와는 차이가 있다. 순차패턴 문제에서는 시간에 따른 순차 데이터가 입력 데이터가 된다. 각각 순차 데이터는 항목들을 포함하는 트랜잭션들로 이루어지고 각 트랜잭션은 트랜잭션시간들을 포함하고 있다. 결론적으로 사용자가 정한최소 지지도(minimum support)를 만족하는 순차패턴을 찾아내는 문제라고 볼 수 있다. 여기에서 최소지지도는 패턴을 포함하는 순차 데이터의 백분율로 정의한다.

다음 <그림 2>와 <그림3>은 순차패턴의 예이다. <그림 2>는 데이터베이스에서 발생한 트랜잭션을 소비자별로 구매한 항목을 시간적으로 배열한 것이다. 소비자 2

의 경우 세 번에 걸쳐 각각 항목 (10 20), (30), (40 60 70) 을 구입한 것이다.

Customer ID	Customer Sequence
1	<(30) (90)>
2	<(10 20) (30) (40 60 70)>
3	<(30 50 70)>
4	<(30) (40 70) (90)>
5	<(90)>

<그림 3> 소비자의 순차 구매 데이터

<그림 3>은 최소지지도를 25%라고 했을 때의 순차적인 패턴을 보여준다.

Sequential Patterns with support > 25%
<(30) (90)>
<(30) (40 70)>

<그림 4> 순차패턴

순차패턴에서 말해주는 것은(30)이라는 항목을 사면 항목(90) 또는 항목 (40 70)을 다음에 산다는 규칙을 의미한다. 우리는 이 결과들을 의사결정에 활용해 (30)이라는 항목을 구입한 구매자들에게 (90), (40), (70)이라는 항목을 권유할 수도 있을 것이고, 또한 상품진열 시에도 활용할 수 있을 것이다.

순차패턴문제는 소매 산업과 우편을 이용한 마케팅(attached marketing), 부가 세일(add-on sale), 고객만족(customer satisfaction)등에서 시작되었지만 다른 과학분야나 경영분야에 적용되었다. 예를 들어 의료분야에서 순차 데이터를 환자의 정후(symptoms)나 질병(disease)으로 보면 트랜잭션은 의사를 방문하는 시점에서 진단된 정후나 질병으로 볼 수 있다. 이 데이터를 통해 발견된 패턴은 질병연구에 많은 도움을 줄 수 있다.

2장에서는 Apriori 알고리즘과 temporal 연관규칙, 순차패턴 등에 대해 살펴보았고 3장에서는 제안하고자 하는 알고리즘에 대해 자세하게 설명한다.

3. 제안하는 알고리즘

본 연구에서는 데이터베이스 상에서 항목간의 발생 간격을 고려하여 temporal 연관규칙을 마이닝하는 알고리즘을 제안한다. 데이터베이스는 현재 사용되고 있는

월(month), 주(week), 일(day)등과 같은 기간(period)으로 나누어져 있으며 이와 같은 기간은 temporal 연관규칙을 발견하는 데 있어서 항목의 발생여부를 구분 짓는 기준이 된다.

제안하는 알고리즘은 기존 temporal 연관규칙보다 빈발 항목이 나타난 기간에 대해 더 정확한 시간정보를 제공하여 기존보다 효과적인 마이닝을 가능하게 한다.

정의 1. 간격 지지도(interval support)란 한 항목이 발생한 후 그 다음 발생하기 전까지의 기간(period)의 크기를 말한다.

간격지지도는 사용자가 결정하며, 최소 간격지지도를 만족하는 경우에는 한 항목이 여러 개의 구간을 갖는다.

<표 3>을 보면 1부터 12까지의 기간이 있다. 앞에서도 언급한 바와 같이 기간(period)은 사용자가 임의대로 구분 지은 것이다. 항목 A를 예를 들어 설명하면 항목 A는 기간3과 기간 9, 10, 11, 12에서 나타나는 것을 볼 수 있다. 간격지지도를 2라고 가정할 때 항목 A는 A[3, 3]과 A[9, 12]라는 두 개의 구간을 갖는다. 최소지지도 만족여부는 각 구간별로 계산되어 빈발여부를 가린다.

3.1 용어 정리

본 절에서는 제안하는 알고리즘과 예제에서 사용하는 용어들에 대해 설명한다.

기간(period) : 일, 주, 월과 같이 일정하게 나누어 놓은 것.

LS_{ix} : 항목 X 가 갖는 i번째 시간기간. 시간기간은 $[t_a, t_b]$ 으로 표현, 이때 $t_a = t_b$.

sup_{ix} : 시간기간이 LS_{ix} 인 항목 X의 지지도. 시간기간 내에서 구해진다.

$conf_i(X \Rightarrow Y[t_a, t_b])$: 시간기간이 LS_{ixY} 인 규칙 $X \Rightarrow Y[t_a, t_b]$ 의 i번째 신뢰도. $t_a \leq t_b$

$$conf_i(X \Rightarrow Y[t_a, t_b]) = sup_{ixY}[t_a, t_b] / sup_{ix}[t_c, t_d]$$

최소간격지지도(minimum interval support) : min interval sup.

최소지지도(minimum support) : minsup.

최소신뢰도(minimum confidence) : minconf.

[예제 1] 다음은 12개의 period를 같은 데이터베이스의 일부이다. 각각의 기간(period)은 4개의 트랜잭션으로 이루어져 있다고 가정한다. 최소간격지지도는 2로 하고 각각의 숫자는 해당

기간에 속한 트랜잭션에서 발생한 항목의 개수이다

<표 3> 예제 1의 데이터베이스

기간 \ 항목	1	2	3	4	5	6	7	8	9	10	11	12
A			4						3	2	1	3
B			3	2	3		1	2				

최소 간격지지도가 2이므로 항목 A는 2개의 구간을 갖고, 항목 B는 1개의 구간을 갖는다.

각각의 구간은 $LS_{1A}=[3, 3]$, $LS_{2A}=[9, 12]$, $LS_{1B}=[3, 8]$ 이고, 항목 A의 첫 번째 구간의 지지도 $sup_{1A}=4/4=1$, 항목 A의 두 번째 구간의 지지도 $sup_{2A}=9/16=0.56$, 항목 B의 지지도 $sup_{1B}=11/24=0.45$ 이다.

2항목을 구해보면 AB의 시간 범위 $LS_{2AB}=[3, 3]$ 이고, $sup_{1AB}=3/4=0.75$ 이다

3.2 제안하는 알고리즘

제안하는 알고리즘은 기존의 Apriori 알고리즘[1]을 근간으로 수정하였다. L_k 는 빈발 k -항목집합을 의미하고, C_k 는 후보 k -항목집합, count는 해당 항목의 수, Tcount는 시간기간 내에서의 전체 트랜잭션의 수이다. 주요 절차를 정리하면 다음과 같다.

1. minimum interval support 적용
2. $L_1 = \{ \text{large 1-itemsets} \}; /* \text{하나의 항목은 minimum interval support에 따라 2개 이상의 } L_1 \text{을 가질 수 있다.*/}$
3. for ($k = 2; L_{k-1} \neq \emptyset; k++$) do begin
4. $C_k = \text{apriori_gen}(L_{k-1}); /* \text{새로운 후보 항목집합 생성. } L_1 \text{과 같이 } C_k \text{도 서로 다른 시간기간을 갖는 같은 항목이 2개 이상 있을 수 있다.*/}$
5. foreach transactions $T \in D$ do begin
6. $C_T = \text{subset}(C_k, T); /* T에서의 후보 항목집합 } c. timestamp \text{은 } c \text{의 시간기간 } [t_1, t_2] */$
7. foreach candidates $c \in C_T$ do
8. $c. count++;$
9. foreach candidates $c \in C_k$ do
10. update $c. Tcount;$
11. }
12. $L_k = \{ c \in C_k | c. count \geq . \} /* \text{새로운 빈발 항목집합 생성. } L_1 \text{과 같이 } L_k \text{도 서로 다른 시간기간을 갖는 같은 항목이 2개 이상 있을 수 있다.*/$
13. }
14. return $L = \cup_k L_k;$

<그림 5> Interval support를 적용한 알고리즘 절차

Temporal 연관규칙을 발견하기 위해서는 기간별로 나뉘어진 데이터베이스상의 항목에 최소간격지지도를 적용하여 1항목집단에서 빈발항목을 찾는다. 각각의 빈발1항목은 Apriori 알고리즘방법으로 빈발항목끼리 join과 prune을 반복하고 시간기간 내에서 해당 항목의 수(count)와 전체 트랜잭션의 수(Tcount)를 계산하여 $k \geq 2$ 인 빈발항목집합을 찾는다. 위의 절차중 Apriori_gen(L_{k-1})의 절차는 다음과 같다.

1. foreach itemset $I_1 \in L_{k-1}$
2. foreach itemset $I_2 \in L_{k-1}$
3. if $(I_1[1] = I_2[1]) \wedge (I_1[2] = I_2[2]) \wedge \dots \wedge (I_1[k-2] = I_2[k-2]) \wedge (I_1[k-1] = I_2[k-1]) \wedge (I_1 \cap I_2) \neq \emptyset$ then
 $\{c = I_1 \bowtie I_2\}$
4. if has_infrequent_subset(c, L_{k-1}) then
5. delete c;
6. else add c to C_k
7. }
8. return C_k ;

<그림 6> Apriori_gen(L_{k-1})의 절차

apriori_gen(L_{k-1})에는 join시 시간기간에 대한 정보($I_1 \cap I_2$)가 추가되었다. $k \geq 2$ 이상의 항목집단에서 시간기간이 이라면 조인(Join)을 할 수 없기 때문이다.

temporal 연관규칙을 만들기 위해서 각각의 신뢰도를 계산하여 최소신뢰도를 만족시키는지를 계산 한다

3.3 기존연구와의 비교

본 연구에서 제안하는 알고리즘은 기존의 temporal 연관규칙에 최소간격지지도를 적용함으로써 결과가 비 빈발로 나올 수 있는 항목에 대해서 시간기간별로 빈발인지의 여부를 판단하기 때문에 특정 항목이 특정 기간에 집중적으로 발생한 경우에는 의미 있는 temporal 연관규칙으로 만들어 낼 수 있다.

그리고 기존의 temporal 연관규칙과는 다르게 한 항목이 여러 시간기간에서 최소지지도와 최소신뢰도가 만족된다면 의미 있는 규칙으로 여러 개가 존재할 수 있다는 것이다. 예를 들어 기존의 temporal 연관규칙은 항목 X와 Y에 대해 $X \Rightarrow Y[t_1, t_2]$ 라는 하나의 규칙[2]만을 생성하였으나, 제안하는 최소간격지지도를 적용하면 항목 X와 Y에 대해 $X \Rightarrow Y[t_1, t_2]$ 라는 temporal 연관규칙 이외에도 다른 시간기간을 갖는 $X \Rightarrow Y[t_1, t_2]$ 라는 temporal 연관규칙을 가질 수 있다.

즉, 같은 항목의 빈발정도가 어떠한 기간에서는 많고 어떠한 기간에서는 적은경우에는 간격지지도를 적용하여 별도의 시간기간을 적용하여 temporal 연관규칙을 찾

고자 하는 것이다.

3장에서는 제안하는 알고리즘에서 사용되는 용어와 temporal 연관규칙을 찾는 과정, 그리고 기존 temporal 연관규칙들과의 차이점에 대해서 설명하였다. 4장에서는 예제를 통해 제안하는 temporal 연관규칙을 구현해 보고자 한다.

4. Temporal 연관규칙 예제

본 장에서는 예제를 통하여 제안하려는 temporal 연관규칙이 생성되는 과정에 대해 설명한다. 4.1에서는 최소간격 지지도를 적용하는 방법을 4.2에서는 4.1에서 구한 1빈발을 적용하여 temporal 연관규칙 생성을 알아본다.

4.1 최소 간격 지지도 적용

다음 <표 4>는 예제에서 사용할 데이터베이스이다.

<표 4> 예제 데이터베이스

기간	TID	항목	기간	TID	항목
P1	TID1	D	P6	TID21	A B
	TID2	E		TID22	A B
	TID3	B C D		TID23	A
	TID4	B D		TID24	A B
P2	TID5	B	P7	TID25	B
	TID6	F		TID26	B
	TID7	D		TID27	A
	TID8	D		TID28	
P3	TID9	A	P8	TID29	A C F
	TID10	A D F		TID30	A D F
	TID11	A B D E F		TID31	A C F
	TID12	A B D E		TID32	C E
P4	TID13	A D F	P9	TID33	A D F
	TID14	A F		TID34	A F
	TID15	B D E		TID35	C D F
	TID16	A B D		TID36	A E F
P5	TID17	B D F	P10	TID37	D F
	TID18	D		TID38	D F
	TID19	A B D		TID39	C E
	TID20	A		TID40	B

데이터베이스는 40개의 트랜잭션으로 이루어져 있고, 4개의 트랜잭션을 한 개의 기간(period)으로 하였다. TID는 고유한 트랜잭션번호를 나타내고, A, B, C, D, E, F는 규칙 생성에 사용되는 항목이다. 최소지지도는 30%이며, 최소간격지지도는 2로 하였다.

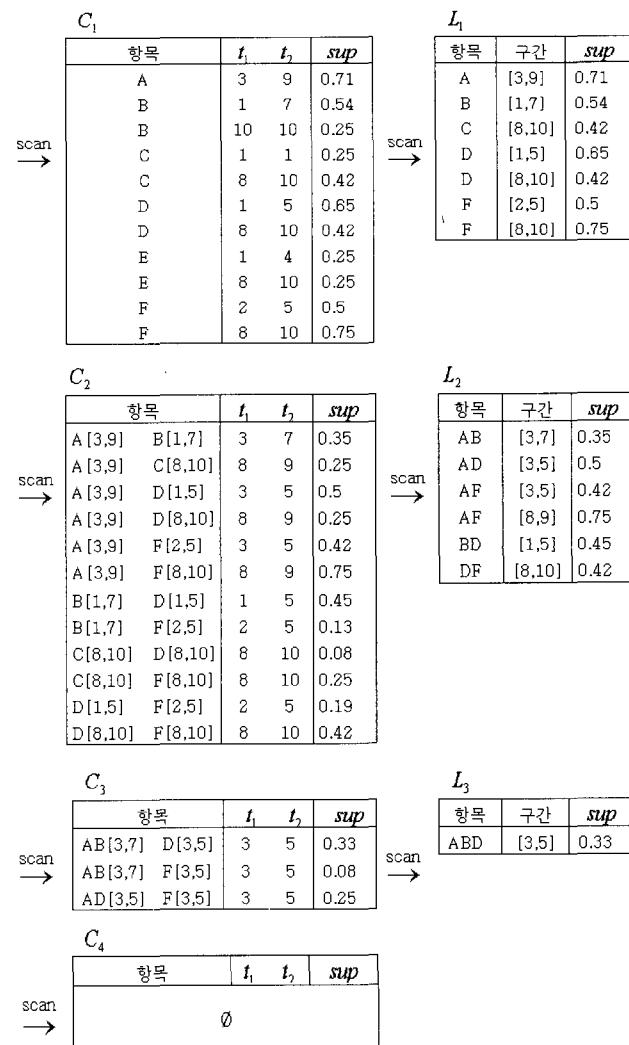
최소간격지지도를 적용하기 위해 위와 같은 표를 이용하였다. 기간과 항목으로 구성되어 있는 <표 5>는 어느 기간에 어떤 항목이 몇 번 발생했는가를 보여준다. 항목C의 경우기간 1에서 1번, 기간 8에서 10까지 5번 발생하였고, 간격지지도가 6이므로 최소간격지지도를 충족시켜 항목 C는 2개의 구간을 갖는다. 표현하면, $LS_{1C}=[1, 1]$ $LS_{2C}=[9, 10]$ 이 된다.

<표 5> 기간별 항목의 수

기간 \ 항목	1	2	3	4	5	6	7	8	9	10
A			4	3	2	4	1	3	3	
B	2	2	2	2	2	3	2			1
C	1							3	1	1
D	3	2	3	3	2			1	2	2
E	1		2	1				1	1	1
F		2	3	2	1			3	4	2

4.2 항목발생의 생성

본 절에서는 빈발항목의 생성과정을 살펴본다.



<그림 7> Apriori 알고리즘 실행예제

위 <그림7>은 최소간격지지도를 적용한 후에 최소지지도를 만족하는 빈발항목을 찾는 과정이다. 그림은 각각 시간 정보를 가지고 있는 항목, 구간 그리고 그 구간에서의 지지도로 구성되어 있다. C_1 은 <표 5>를 참고하여 모든 항목의 구간과 지지도를 계산한 것이고 L_1 은 최소지지도 30% 이상인 항목들을 빈발 항목집합으로 만든 것이다. L_1 을 보면 항목 C_1 가 있는 것을 알 수 있다. 간격지지도를 적용하지 않고 [2]처럼 하였다면 최소지지도를 만족시키지 못하기 때문에 빈발 1항목집합에 포함시키지 못할 뿐만 아니라 이후에도 특정 구간에서 다른 항목들과 연관성을 고려해 볼 수가 없다. 제안하는 알고리즘에서도 Apriori 알고리즘을 사용하였기 때문에 $k \geq 2$ 인 후보 항목을 조인(Join) 할 때 앞부분이 같은 빈발항목만을 대상으로 하였다. 후보 항목집합 C_2 를 보면 L_2 의 항목 중 앞부분이 A 로 같은 것만을 대상으로 후보 항목집합을 생성하고 생성된 구간에서의 지지도를 계산하였다. C_2 가 공집합인 이유는 L_1 에서 더 이상 조인할 빈발항목이 없기 때문이다.

5. 결 론

데이터 마이닝의 목표는 알려지지 않은 정보를 얻어 의사결정에 활용하는 것이다. 연관규칙은 데이터베이스 전체를 대상으로 항목간의 관련성을 고려하여 규칙을 생성하는 것이고, temporal 연관규칙은 규칙에 시간이라는 것을 포함시켜 규칙에 더 많은 정보를 표현하였다.

본 연구에서는 기간(period)으로 나누어져 있는 데이터베이스를 대상으로 항목의 발생 간격을 고려한 간격지지도를 적용하여 temporal 연관규칙을 살펴보았다. 연관규칙이나 간격을 고려하지 않은 temporal 연관규칙에서 발견할 수 없는 유용한 정보를 발견할 수 있도록 문제점을 보완하였다.

본 논문에서 제안한 알고리즘을 사용하면 지지도는 낮지만 신뢰도가 높은 항목에 대해 더 정확한 시간정보를 가지고 있는 연관규칙을 마이닝할 수 있다. 이는 사용자에게 상품변들링과 같은 마케팅 전략 수립은 물론이고 적절한 시기에 상품을 배치할 수 있게 함으로써 비즈니스 성과와 고객 만족을 증진시킬 수 있다.

참고문헌

- [1] Jiawei Han, Micheline Kamber, Data Mining : concepts and Techniques, Morgan Kaufmann publishers, 2000
- [2] Juan M. Ale, Gustavo H. Rossi, "AN APPROACH TO DISCOVERING TEMPORAL ASSOCIATION RULES". Proc. of the 2000 ACM symposium on Applied computing 2000
- [3] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo, "Finding interesting rules from large sets of discovered association rules," Proc. 3rd Int'l Conf. on Information and Knowledge Management, Gaithersberg, Maryland, pp. 401-408, Nov. 1994.
- [4] Pieter Adriaans and Dolf Zantinge, DATA MINING, Addison-Wesley, Harlow, U.K., 1996.
- [5] R. Agrawal, R. Srikant : "Fast Algorithms for Mining Association Rules" ,Proc. of the 20th Int'l Conference on Very Large Databases, Santiago, Chile, Sept. 1994. Expanded version available as IBM Research Report RJ9839, June 1994.
- [6] R. Agrawal, R. Srikant, "Mining Sequential Patterns." Proc. of the 11th International Conference on Data Mining Engineering, Taipei, Taiwan, March 1995
- [7] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," Proc. of the ACM SIGMOD Int'l Conference on Management of Data, Washington D.C., pp. 207-216, May 1993.
- [8] R. Srikant and R. Agrawal, "Mining Generalized Association Rules," Proc. of the 21st Int'l Conference on Very Large Databases, Zurich, Switzerland, Sep. 1995.