오차를 허용하는 주기적 연관규칙 탐사를 통한 오차의 경향성에 관한 연구

배수균*, 남도원**, 이동하***, 이전영**

Discovery of Cyclic Association Rule with Loose Cycle and Error Cycle over Loose Cycle

Soo-Kyoon Bae*, Do-Won Nam**, Dong-Ha Lee***, Jeon-Young Lee**

요 약

주기적인 연관규칙은 타겟데이터베이스를 일정 단위시간으로 나누었을 때 연관규칙이 만족하는 구간이 일정한 주기마다 발생하는 패턴을 탐색하는 방법이다. 하지만, 이 방법은 엄격한 주기를 가지도록 하여 실제 데이터에 그대로 적용하기가 어려웠다. 예를 들어 편의점 데이터에서 매일 오전 7시-8시 사이에 주기적으로 발생하는 연관규칙을 발견할 때, 이러한 연관규칙을 주기적인 연관규칙이라고 한다. 하지만, 실제 데이터에서는 날씨와 같이 사람의 행동에 영향을 미치는 다른 요인 때문에 항상 일정한 주기를 가지는 연관규칙을 찾기는 어렵다. 본 논문에서는 주기가 일정하지 않은 연관규칙을 찾기 위해서 연관규칙의 주기성을 허용 오차를 포함하여 재정의하고, 오차를 허용하기 위한 탐색 알고리즘을 보완하였다. 반면에, 오차를 허용함으로써 오차를 허용하지 않는 경우보다 더 많은 주기성을 찾을 수 있을 뿐만 아니라, 동일한 주기를 가지지만 오프셋이 다른 여러 개의 비슷한 주기까지 찾게 되어 사용자가 의미 있는 연관규칙을 찾는데 방해가 된다. 본 논문에서는 이를 해결하기 위해서 오차를 허용하는 주기적 연관규칙의 오차의 정도를 측정하기 위한 단위로 집중도(intensity)와 경향성(tendency)을 제안한다. 주기적 연관규칙이 매 주기마다 정확한 세그먼트에 나타나는 정도를 나타내는 집중도와, 최소 평균오차를 의미하는 경향성을 이용하여 유사한 주기들 중에서 대표주기만을 찾을 수 있도록 한다. 또한, 오차를 허용하는 주기적 연관규칙에서 오차가 주로 발생하는 패턴을 분석함으로써 고객들의 수요 경향성을 더 잘 파악할 수 있다. 예를 들어, 평소에는 매일 오전 7시~8시에 나타나던 연관성이 지각하는 사람들이 많은 월요일에는 1시간 늦은 8시~9시에 나타난다는 오차 정보까지 파악할 수 있다. 이러한 월요일마다 1시간 늦게 나타나는 오차의 경향성을 나타내는 오차 주기(error cycle)를 이용함으로써 고객들의 수요의 경향성을 좀 더 세밀한 부분까지 파악할 수 있게 해 준다.

Key words : data mining, association rule, cyclic association rule, loose cyclic association rule, intensity, tendency,error cycle

---

* 포항공과대학교 정보통신학과

** 포항공과대학교 전자컴퓨터공학부

*** 포항공과대학교 정보통신연구소

1. 서론

연관규칙은 1993 년에 처음 소개되어진 데이터 마이닝 기법중의 하나이다.[1] 연관규칙은 항목들의 집합으로 표현된 트랜잭션들로 구성된 데이터베이스에서 X→Y 의 형태로 표현되는 규칙이다. 예를 들어, '빵→우유(지지도:5%, 신뢰도:75%)'라는 연관규칙이 있을 때, 우리는 전체 판매 데이터의 트랜잭션 중에서 5%에서 빵과 우유를 동시에 팔았다는 정보와 빵을 판 트랙잭션 중에서 우유를 함께 팔 확률이 75%라는 정보를 얻을 수 있다.

주기적인 연관규칙[3]은 전체 트랜잭션을 일정 시간 단위로 나누었을 때 연관규칙이 성립하는 시간 구간이 일정 주기를 가지고 반복적으로 나타남을 의미하며 이러한 반복 패턴을 찾아내는 것이다. 이러한 주기적 연관규칙을 찾기 위해서는 타겟데이터베이스를 사용자의 요구에 맞게 일정한 시간 단위로 구분하고, 단위시간별 연관규칙들 사이에 나타나는 주기성을 찾는 것이다. 주기적인 연관규칙에서의 주기는 $c=(l,o)$와 같은 형식으로 표기하며, 단위시간으로 구분된 타겟데이터베이스의 o 번째 시간 세그먼트를 시작 지점으로해서 매 l 번째 시간 세그먼트에서 주기적으로 연관규칙을 만족함을 의미한다. 예를 들어, 편의점의 판매 데이터베이스에서 매일 오전 7~8 시 사이에 '빵→우유(지지도:5%, 신뢰도:75%)'인 연관규칙을 찾았다고 하자. 이 때 찾아진 연관규칙의 주기는 $c=(24,7)$로써 단위시간이 1 시간이고, 주기길이(l)는 하루이며, 이 주기의 시작위치(o)는 자정을 기준으로 하여 8 번째 단위시간인 오전 7-8 시를 나타낸다.

기존의 연구에서는 매 주기길이마다 엄격하게 주기성을 요구하기 때문에 주기성이 엄격히 지켜지지 않는 실제 데이터에 이를 그대로 적용하기에는 한계가 있다. 이 문제를 해결하기 위해서 주기에 해당하는 시간 세그먼트에서 연관규칙이 매번 만족되지는 못하더라도 일정한 미스매치 한계(mismatch threshold)의 오차를 허용하는 연구가 있었다.[4] 예를 들어, 빵→우유인 주기적 연관규칙의 주기가 (24,7)이고, 미스매치 한계가 5 라고 한다면, 매일 7-8 시 사이의 시간 세그먼트 중에서 빵→우유인 연관규칙이 만족되지 않는 세그먼트가 5 번 정도 발생하더라도 이를 허용하는 것이다.

그러나 미스매치 한계를 허용하는 연구는 주기의 엄격성으로부터 매 주기마다 반드시 연관규칙이 성립하지 않아도 됨을 허용한 것일 뿐이고, 연관규칙의 발생 패턴이 가지는 주기 자체에 오차를 허용한 것이 아니다. 예를 들어, 일반적으로는 '빵→우유, $c=(24,7)$'인 주기가 발견되어 7-8 시에 연관규칙이 만족되나 눈이나 비가 오는 날에는 해당 연관규칙이 1 시간 늦은 8-9 시에 나타난다면 주기의 길이를 엄격히 제한하고 있는 기존의 방식으로는 이와 같이 오차를 포함한 주기성을 찾을 수 없다. 본 논문에서는 이러한 문제점을 극복하기 위하여 연관규칙 발생 패턴의 주기의 길이에 오차를 허용하는 주기적 연관규칙을 탐색하는 방법을 제안한다. 이 방법은 연관규칙이 항상 일정한 주기마다 정확히 발생되지 않더라도 주기길이가 오차범위 이내에서 발생하는 경우를 허용하도록 해 준다.

오차를 허용하는 주기적 연관규칙을 탐색하게 되면 기존의 주기적 연관규칙을 찾는 방법에 비하여 많은 연관규칙이 검색될 뿐만 아니라 동일한 주기길이를 가지는 오프셋이 다른 유사한 주기가 여러 개 발견되어 사용자가 의미있는 연관규칙을 탐사하는데 방해가 된다. 예를 들어, 기존의 주기적 연관규칙에서는 $c=(24,7)$이라는 주기만 발견된 경우를 오차범위가 1 인 오차를 허용하는 주기적 연관규칙으로 탐사하면 주기가 동일하고 오프셋이 다른 $2(2*e_b)$개의 유사 주기인 $c_1=(24,6,1)$, $c_1=(24,8,1)$까지 찾게 된다. 이렇게 많은 유사한 주기들이 탐색되는 문제점을 제거하기 위해서 본 논문에서는 오차를 허용하는 주기의 오차의 정도를 측정할 수 있는 단위인 집중도(intensity)와 경향성(tendency)를 제안하고, 이 두 측정단위를 통해서 주기성을 가장 잘 표

현하는 주기만을 선택하고, 나머지 주기를 가지치기함으로써 사용자가 의미있는 연관규칙을 탐사하는데 도움을 주고자 한다.

또한, 오차를 허용하는 경우에는 추가적으로 오차의 경향성을 분석함으로써 실제 데이터가 발생되는 패턴을 좀 더 정확하게 분석할 수 있다. 예를 들어, 평소에는 매일 오전 7시~8시에 나타나던 연관성이 지각하는 사람들이 많은 월요일에는 1시간 늦은 8시~9시에 나타난다는 오차 정보를 파악할 수 있다고 하자. 여기에서 우리는 월요일마다 1시간 늦게 나타나는 오차의 경향성을 오차 주기(error cycle)라고 하고, 이러한 오차의 경향성을 분석함으로써 고객들의 수요의 경향성을 좀 더 세밀한 부분까지 파악하고자 한다.

2. 문제정의

기존의 주기적 연관규칙은 매 주기마다 해당 시간 세그먼트에서 엄격하게 연관규칙을 만족해야만 한다. 하지만 실제 데이터에 주기성을 엄격하게 적용하기에는 문제점이 있다. 이를 보완하기 위해서 주기길이에 일정한 오차범위를 허용하는 방법을 제시한다.

오차를 허용하는 주기적 연관규칙은 기존의 방법에서 오차범위라는 척도를 추가적으로 사용한다. 이렇게 함으로써 오차를 허용하는 주기적 연관규칙에서는 엄격하게 주기가 지켜지지 않더라도 일정한 오차범위 이내에 주기성이 존재하면 주기적 연관규칙이 인정된다. 정의 1에서는 오차범위를 덧붙여 새롭게 정의한 오차를 허용하는 주기적 연관규칙에 대해서 정의한다.

정의 1. 항목집합 X에 대한 시간 세그먼트별로 빈번항목집합인지 여부를 표시한 비트 스트림 s의 크기가 n일 때, 오차를 허용하는 주기적 연관규칙(loose cycle)의 주기가 $c_l=(l,o,e_b)$를 가진다면 다음을 만족하여야 한다. (주기길이 l, 오프셋 o, 오차범위 e_b)

→ 모든 가능한 비트 스트림 구간 $[l*i+o-e_b, l*i+o+e_b]$에서 반드시 하나 이상의 값이 빈번항목집합임을 표시하는 1인 값이 존재하여야 하고, 이러한 구간을 오차구간(error interval)이라고 한다. (i는 정수, $l*i+o-e_b \geq 0$, $l*i+o+e_b < n$)

→ 상호 인접한 오차구간이 겹쳐지지 않도록 하기 위하여 오차범위는 $0 \leq e_b \leq (l-1)/2$ 이어야 한다.

3. 오차를 허용하는 주기적 연관규칙 찾기

주기적 연관규칙을 찾기 위해 기존의 연구에서는 순차적(Sequential)인 알고리즘과 순환적인(Interleaved)인 알고리즘을 제안하였다.[3] 본 논문에서는 기존의 두 가지 방법에 대해서 간단히 언급하고, 기존의 방법을 보완하여 오차를 허용하는 주기적 연관규칙을 찾는 방법에 대해서 논한다.

3.1 주기적 연관규칙을 찾는 방법

순차적인 방법으로 주기적 연관규칙을 찾기 위해서는 크게 두 단계의 작업이 순차적으로 진행된다. 첫번째 단계의 작업은 각 시간 세그먼트별로 Aprori와 같이 기존에 제시된 연관규칙을 찾는 방법들을 사용하여 모든 연관규칙을 찾는다. 두번째 단계에는 각 시간 세그먼트별로 찾아진 연관규칙을 주기탐색(cycle detection) 방법을 이용하여 주기성이 존재하는 연관규칙이 있는지를 찾게 된다.

순차적인 방법에서는 각 시간 세그먼트별로 모든 항목집합에 대한 지지도를 검사한다. 여기서는 주기성이 결여된 항목집합에 대해서도 지지도를 검사해야 하기 때문에 성능이 떨어진다. 이를 보완하기 위해서 가능한 빨리 항목집합별로 주기성을 파악하여 지지도를 검사할 때에 이를 이용하여 주기가 존재할 수 있는 항목집합에 대해서만 조사하고, 그렇지 않은 항목집합에 대해서는 해당 시간 세그먼트를 건너뛰고 지지도를 조사하는 순환적인 방법이 제안되었다.

3.2 주기생략 / 주기삭제 / 주기제거

순환적인 방법에서는 성능향상을 위한 가지치기 기법으로 3 가지 개념을 사용하였다. 이는 각각 주기생략(cycle skipping), 주기가지치기(cycle pruning), 주기삭제(cycle elimination)로 구분된다. 여기에서는 오차를 허용하는 주기적 연관규칙까지 찾을 수 있도록 확장된 개념에 대해서 논한다.

주기생략은 시간 세그먼트 D[i]에서 항목집합 X의 지지도를 검사하지 않기 위해서는 항목집합 X가 시간 세그먼트 D[i]를 포함하는 어떠한 주기도 가지지 않아야 함을 나타낸다. 이 기법을 오차를 허용할 경우에 적용하면 항목집합 X가 시간 세그먼트 D[i]에서 뿐만 아니라 오차범위 이내인 D[i-e_b]에서부터 D[i+e_b]까지의 시간 세그먼트를 포함하는 어떠한 주기도 가지지 않아야만 한다. 이와 같은 주기생략을 적용하기 위해서는 해당 항목집합이 가지는 모든 주기를 먼저 정확하게 파악해야만 가능한데 이를 사전에 미리 완전히 알기는 불가능하다. 다음에 살펴볼 주기가지치기와 주기삭제 기법을 이용하여 주기가 될 수 없는 것을 미리 가지치기 하는 방법을 이용하여 주기생략 기법을 활용할 수 있도록 해준다.

주기가지치기는 k-빈번항목집합의 주기를 이용하여 (k+1)-후보빈번항목집합의 후보주기(candidate cycle)를 찾는 방법으로 후보주기가 될 수 없는 주기를 가지치기함으로써 순환적인 방법의 성능을 향상시키는 기법이다. (k+1)-후보빈번항목집합의 후보주기를 찾기 위해서는 우선 (k+1)-후보빈번항목집합을 Apriori 에서 사용하였던 가지치기 기법을 이용하여 생성하여야 한다. 그 이후에 (k+1)-후보빈번항목집합을 만들기 위해 사용된 k-빈번항목집합에서 공통적으로 만족하는 주기만을 검색하여 (k+1)-후보빈번항목집합의 후보주기를 만든다. 이 기법은 오차를 허용할 경우에도 기본적으로는 동일하게 적용되나 추가적인 작업이 필요하다. 기존의 주기가지치기 기법에서 찾은 후보주기 중에서 (k+1)-후보빈번항목집합의 주기가 결코 될 수 없는 것이 존재하기 때문이다. 아래의 예제에서는 그러한 예를 설명하고 있다. 이 문제를 해결하기 위해서는 두 개의 k-빈번항목집합인지 여부를 표시한 비트 스트림을 비트 AND 연산한 결과를 주기탐색(cycle detection) 기법을 사용하여야 정확한 후보주기를 생성할 수 있다.

예제 1. k-빈번항목집합 X와 Y의 시간 세그먼트별 빈번항목집합 여부를 표시한 비트 스트림이 bx, by 일 때, (k+1)후보빈번항목집합의 후보주기를 찾고자 한다. 이 때, l=4, e_b=1 인 후보주기를 찾고자 한다.

bx = 0100 0010 0001 --- c_l(4,2,1)

by = 0100 0010 0010 --- c_l(4,2,1), c_l(4,1,1)

bx 와 by 에서 공통적으로 검색되는 주기인 c_l=(4,2,1)이 기존의 주기적 연관규칙에서는 항목집합 X∪Y 의 후보주기로 검색된다. 하지만, 해당 주기 c_l=(4,2,1)는 오차를 허용할 경우에는 결코 후보주기로써 존재할 수 없는 주기이다.

bx∪by = 0100 0010 0000 --- no cycle

이러한 문제점을 없애기 위해서는 후보항목집합의 후보 비트 스트림(bx∪by)을 생성한 후에, 이 값을 주기탐색 기법을 이용하여 후보주기를 검색하게 되면 정확한 후보주기만을 검색할 수 있다.

주기삭제는 순환적인 방법을 통한 주기가지치기 기법 중에서 가장 핵심이 되는 기법이다. 항목집합 X의 지지도가 최소지지도를 넘지 못하는 시간 세그먼트 D[i]를 검색했을 때, 항목집합 X는 D[i]를 포함하는 어떠한 주기도 가질 수 없고 해당 주기들은 후보주기에서 삭제하게 된다. 이 기법을 오차를 허용할 경우에 적용하면 다음과 같이 보완된다. 최소지지도를 넘지 못하는 시간 세그먼트 D[i]를 검색했을 때, 이 세그먼트를 포함하는 연속된 2*e_b+1 개의 세그먼트 D[i-e_b+x] ~ D[i+e_b+x] (단, |x|<=e_b)에서 각각 최소지지도를 모두 넘지 않을 경우에, 연속된 세그먼트의 중심인 D[i+x]를 포함하는 모든 주기를 삭제함으로써 주기생략 기법을 이용할 수 있게 된다. 아래의 예제는 주기삭제 기법을 통

해서 실제로 어떤 주기가 삭제되는지를 보여주고 있다.

예제 2. 각 시간 세그먼트별로 빈번항목집합 여부를 표시한 b를 가지고 있는 항목집합 X에 대한 D[5]에서 지지도를 검사할 때, (단, e_b=1, l_{min}=l_{max}=4)

b = 1110 0?01 1???

시간 세그먼트 D[4]까지 지지도를 조사한 싯점까지는 삭제되는 주기가 하나도 존재하지 않는다. 시간 세그먼트 D[5]에서 만약 지지도가 최소지지도를 넘지 않을 경우에는 D[3]~D[6]까지 연속해서 최소지지도를 넘지 않게 된다. 여기에서 3개의 연속된 시간 세그먼트에서 최소지지도를 넘지 않는 2(D[3]~D[5], D[4]~D[6]) 개의 부분을 추출하게 되고 각각의 중심 시간 세그먼트인 D[4]와 D[5]를 통과하는 c_f=(4,0,1), c_f=(4,1,1)이 주기삭제에 의해서 제거된다.

3.3 순환적인 방법으로 오차를 허용하는 주기적 연관규칙 찾기

순차적인 방법으로 오차를 허용하는 주기적 연관규칙을 찾는 것은 오차를 허용하지 않는 경우와 기본적인 접근방법에 있어서는 동일하다. 단지, 위에서 살펴본 주기 가지치기를 위한 세가지 방법을 이용할 때 오차를 고려하여 사용하여야만 한다. 아래의 그림에서는 순환적인 방법으로 오차를 허용하는 주기적 연관규칙을 찾기 위한 첫번째 단계로써 주기적으로 빈번한 항목집합을 찾는 방법을 간략하게 소개한 알고리즘을 제시하고 있다.

6번째 행에는 3가지 가지치기 방법중에서 주기생략을 행하는 부분이다. 새로운 시간 세그먼트에 대해서 지지도 조사하기 전에, 해당 시간 세그먼트에서 주기성이 결여된 항목집합이 있는지 여부를 검사하여 해당 항목집합에 대한 지지도 검사를 행하지 않게 하는 부분이다. 9-11 행에서는 주기삭제가 이루어 지는 부분으로 각 시간 세그먼트마다 지지도를 조사한 이후에 최소지지도를 넘지 못하는 항목집합에 대해서 주기삭제를 통해서 불필요한 주기를 삭제하는 부분이다. 13 행에서는 주기가지치기를 행하는 부분이다. 항목집합의 개수가 k개인 모든 시간 세그먼트에 대해서 지지도 조사와 주기성 조사가 끝난 이후에, 항목집합의 개수가 k+1 개인 후보 항목집합의 주기를 생성하는 부분이다.

그림 1. 순환적인 알고리즘

```
1   k =1
2   create 1-cyclic candidate itemset
3   → set the possibility of all possible cycle
4   While ( exist k-cyclic candidate)
5     for TS(time_segment) = 0 to n-1
6       generate k-candidate itemset in current TS
7       calculate support
8       for (each itemset)
9         if (itemset sup less than min_sup in 2e_b +1
10            consecutive TS containing current one)
11            cycle elimination
12    store current k-cyclic large itemset
13    generate next (k+1)-cyclic candidate itemset
14    k = k+1
```

4. 오차를 허용하는 흥미로운 주기적 연관규칙 찾기

오차를 허용하는 주기적 연관규칙을 탐색하게 되면 오차를 허용하지 않는 경우에 비하여 많은 연관규칙이 검색될 뿐만 아니라 찾아진 연관규칙에서도 주기길이는 동일하나 오프셋이 다른 여러 개의 유사한 주기를 검색하게 되어 사용자가 흥미로운 주기적 연관규칙을 찾는 데 어려움을 겪게 된다. 이를 해결하기 위하여 본 논문에서는 집중도(intensity)와 경향성(tendency)이라는 척도를 추가하여 오차를 허용하는 주기적 연관규칙 중에서 흥미로운 주기만을 검색할 수 있도록 제안한다.

수식 1.

$$| \text{loose cycle} | \le (2*e_b+1) * | \text{cyclic} | + | \text{pure loose cycle} |$$

위의 수식에서는 오차를 허용하는 연관규칙으로 찾았을 때 발견되는 연관규칙의 개수를 기존의 주기적 연관규칙으로 찾을 때와 비교하여 표시한 것이다. 위의 수식을 통해서 알 수 있듯이 오차를 허용하는 경우가 오차를 허용하지 않는 경우보다는 항상 더 많은 연관규칙을 찾게 된다. 예를 들어, 기존의 주기적 연관규칙에서 c=(7,4)라는 하나의 주기만 검색되었을 경우에, 이 데이터를 오차범위를 1로 설정한 오차를 허용하는 주기적 연관규칙에서는 c_1=(7,4,1)뿐만 아니라, 오차범위 이내인 c_1=(7,3,1), c_1=(7,5,1)이라는 주기까지 탐색되어 최소한 3개의 주기성을 찾게 되고, 추가적으로 기존의 주기성에서는 추론 불가능하며 오차를 허용할 경우에만 추출되는 주기성(pure loose cycle)도 있기 때문에 탐색되는 주기가 더 많아진다. 아래의 예제에서는 오차를 허용할 경우와 허용하지 않을 경우에 추출되는 주기의 개수에 대해서 예를 들어 설명하고 있다.

예제 3. 항목집합 X에 대한 시간 세그먼트별로 빈번항목집합인지 여부를 표시한 비트 스트림 b를 통해서 기존의 주기적 연관규칙을 찾는 방법을 통해서는 단지 c=(4,1)이라는 주기만을 검색할 수 있다.

b = 0110 0110 0101 → c=(4,1)

b = 0110 0110 0101 → c_1=(4,3,1)

위와 동일한 데이터를 근거로 하여 오차범위가 1인 오차를 허용할 경우를 통해서 주기성을 검사하면 수식 1에서 살펴본 바와 같이 추가적으로 c_1=(4,0,1), c_1=(4,2,1)라는 주기를 찾을 수 있으며, 기존의 주기적 연관규칙에서 추론 불가능하며 오차를 허용할 경우에만 찾을 수 있는 주기인 c_1=(4,3,1)까지 찾을 수 있다. .

4.1 유사주기들 중에서 대표주기 찾기

오차를 허용함으로써 주기가 일정하지 않은 연관규칙까지 찾을 수 있게 되었지만, 주기길이는 동일하지만 오프셋이 다른 유사한 주기까지 검색하게 된다. 이와 같이 유사한 주기들 중에서 주기의 특성을 가장 잘 표현하는 주기성만을 찾기 위해서 오차를 허용하는 주기의 특징을 표현하는 측정단위로 집중도(intensity)와 경향성(tendency)을 제안한다.

집중도는 기존의 연구에서 제안한 미스매치(mismatch)의 반대개념이다. 기존의 주기적 연관규칙이 오프셋을 시작 시간 세그먼트로 하여 매 주기를 더한 시간 세그먼트마다 연관규칙이 만족하는 것인데, 오차를 허용함으로써 주기가 일정하게 나타나지 않는다. 이렇게 정확하게 나타나지 않는 횟수를 미스매치라고 하였다. 반면, 본 논문에서 제안한 집중도는 주기가 일정하게 나타나는 정도를 표시하는 값으로 전체 주기의 횟수 중에서 정확한 주기마다 나타난 횟수를 나눈 값으로 정확한 주기에 연관규칙이 만족하는 정도를 나타낸다.

경향성은 주기적 연관규칙이 정확한 주기마다 나타나는지 여부를 매 주기마다 오차를 측정하여 찾아지는 최소 평균오차를 의미한다. 각 주기마다 최소 오차를 찾을 때 오차의 절대값이 동일할 경우가 존재한다. 이러한 경우에는 좌우 최소 평균오차를 따로 관리하여 해결한다. 이 때 구해지는 좌우 최소 평균오차 범위 내에서 최소 평균오차를 경향성이라고 한다. 즉, 경향성은 0이거나, 좌우평균오차 중에서 절대값이 작은 것 중의 하나가 된다.

아래의 그림과 예제에서 집중도와 경향성을 이용하여 기울어진 주기를 가지치기하는 알고리즘과 실제 이 방법을 구체적으로 적용하여 설명하고 있다. 오차를 허용하는 주기 자체가 의미가 있는지 여부를 최소집중도를 통해서 구하는데, 최소집중도보다 작은 것은 오차가 발생하는 시간 세그먼트가 많음을 의미하여 이는 다른 주기에 의해서 가지치기 될 수 있음을 의미한다. 그리고, 경향성의 절대값이 0.5보다 크다는 것은 평균오차를 보정한 오프셋이 현재의 오프셋보다는 다른 오프셀 값에 더 가깝다는 것을 나타내서 가지치기 대상이 됨을 나타낸다.

그림 2. Pruning leaning cycle

if (intensity (c_1) < min_intensity)

if (|tendency(c_1)| ≥ 0.5 and |tendency(c_2)| ≤ 0.5)

re_off = o_1 + tendency(c_1)

if (re_off < -0.5 && l_2<>0) re_off += l

if (re_off > (l-0.5) && l_2<>(l -1)) re_off -= l

if (|o_2 – re_off| ≤ 0.5)

prune c_1

예제 4. 항목집합 X에 대한 시간 세그먼트별로 빈번항목집합인지 여부를 표시한 비트 스트림 b가 아래와 같을 경우에 c_1=(4,1,1)과 c_2=(4,2,1)을 발견할 수 있다.

b = 0100 0100 0100 0010

위와 같이 발견된 주기들 중에서 대표성이 결여된 주기를 검색하기 위해서 우선, 위에서 언급한 오차를 허용하는 주기의 측정단위인 집중도와 경향성을 분석해야만 한다.

Intensity(c_1) = 3/4, tendency(c_1) = 0.25

Intensity(c_2) = 1/4, tendency(c_2) = -0.75

c_1의 집중도는 주기성이 발견되는 4개의 시간 세그먼트 중에서 3군데에서 오차가 없이 발생되어 3/4이고, c_1의 경향성은 오차가 +1인 것이 하나만 있기 때문에 1/4이다.

reformed offsct(c_1) = 1.25

reformed offset(c_2) = 1.25

최소집중도(min_intensity)를 0.5라고 할 경우에 c_2는 최소집중도를 만족 못할 뿐만 아니라, 경향성의 절대값도 0.5보다 크기 때문에 가지치기 대상이 된다. 또한, 경향성을 보정한 오프셋 값(reformed offset)이 현재의 오프셋 값이 2보다는 c_1의 오프셋인 1에 더 가깝기 때문에 c_2는 c_1에 의해서 가지치기 된다.

4.2 오차주기 탐색

오차를 허용함으로써 기존의 주기적 연관규칙에서는 보여지지 않는 오차가 발생하는 패턴을 분석함으로써 얻어지는 오차주기(error cycle)를 검색할 수 있다. 오차주기는 홀로 존재 가능한 주기가 아니라 오차를 허용하는 주기에서 발생하는 오차들이 발생하는 패턴을 분석함으로써 찾아지는 것이다. 이 때에 분석대상이 되는 오차는 경향성을 구할 때 사용되는 최소오차를 가지는 것을 대상으로 한다.

그림 3. 오차주기의 탐색 대상 시간 세그먼트

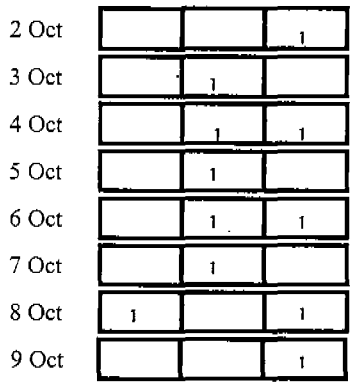

그림 3에서는 오차주기를 탐색하기 위해서 사용되는 시간 세그먼트를 예를 들고 있는 것으로써 c_f=(24,7,1)을 표시하고 있다. 각 날짜별로 세 개의 시간 세그먼트가 존재하는데, 가운데 시간 세그먼트는 오차가 없는 경우이고, 나머지는 각각 오차가 -1과 +1이 발생한 것을 나타낸다. 각 시간 세그먼트에서 1로 표시된 부분은 빈번(large)함을 의미한다. 그리고 배경이 회색으로 되어 있는 부분은 경향성을 계산하기 위해 사용된 시간 세그먼트로써 각 날짜별로 최소오차를 가지는 시간 세그먼트를 나타내는 것으로 오차주기를 찾기 위해서 사용되는 시간 세그먼트로 쓰인다.

5. 결론

본 논문에서는 연관규칙의 발생 패턴이 보이는 주기의 길이에 오차를 허용하는 새로운 주기적 연관규칙 탐색에 대해 정의하였고, 이를 탐색하기 위한 방법을 보였다. 제안한 방법은 기존 연구에서 이용하던 가지치기 기법들을 약간의 수정만으로 모두 적용할 수 있도록 해주기 때문에 큰 성능저하 없이 오차를 포함한 주기적 연관규칙을 탐색해낼 수 있게 한다. 주기의 길이에 대한 엄격성을 오차 범위까지 변화할 수 있게 함으로써 실제의 데이터

에 적용하였을 때 유용한 패턴을 놓치지 않고 찾아낼 수 있게 한다. 추가적으로 오차를 허용함으로써 주기길이는 동일하지만 오프셋이 다른 유사한 주기를 검색하게 되어 사용자가 의미있는 주기성을 검색하는데 문제점이 발생하는데, 이는 오차의 정도를 측정하는 단위인 집중도와 경향성을 통해서 찾아지는 주기의 개수를 줄일 수 있다. 이와 같이 유사한 주기가 발생하는 반면에 오차의 발생패턴을 분석함으로써 오차의 경향성을 파악할 수 있게 되어 데이터의 실제적인 분포에 대해서 보다 더 자세하게 분석할 수 있게 되었다.

참고문헌

[1] Rakesh Agrawal, Tomasz Imielinski, and Arun Sqami. "Mining Association Rules between Sets of Items in Large Databases". In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, May 1993.

[2] Juan M. Ale and Gustavo H. Rossi, "An approach to discovering temporal association rules", In Proceedings of the 2000 ACM symposium on Applied computing 2000 (volumn 1), 2000.

[3] Baun Özden, Shidhar Ramaswamy, and Abraham Silberschtaz. "Cyclic Association Rules", In Proceedings of the 14th International Conference on Data Engineering, February 1998.

[4] Shidhar Ramaswamy, Sameer Mahajan and Avi Silberschatz. "On the Discovery of Interesting Patterns in Association Rules", In Proceedings of the 24th International Conference on Very Large Data Bases, August 1998.

[5] Roberto J. Bayardo Jr. "Efficiently Mining Long Patterns from Databases", In Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, June 1998.

[6] 남도원, "연관규칙에서 신뢰도가 한시적인 특징을 보이는 부분구간의 탐사에 관한 연구", 포항공과대학교 석사학위논문, 1998.