동적 가중치를 이용한 효율적인 순차 패턴 탐사 기법

최필선, 강동현, 김환, 김대인, 황부현
전남대학교 전자컴퓨터공학과
e-mail:pilddong@nate.com

Efficient Mining of Dynamic Weighted Sequential Patterns

Pilsun Choi, Donghyun Kang, Hwan Kim, Daein Kim, Buhyun Hwang
Dept of Electronic Computer Engineering, Chonnam University

요 약

순차 패턴 탐사 기법은 순서를 갖는 패턴들의 집합 중에 빈발하게 발생하는 패턴을 찾아내는 기법이다. 순차 패턴 탐사 분야 중에 동적 가중치 순차 패턴 탐사는 가중치가 시간에 따라 변화하는 컴퓨팅 환경에 적용하는 마이닝 기법으로 동적인 중요도 변화를 마이닝에 적용하여 다양한 환경에서 활용 가능하다. 이 논문에서는 다양한 순차 데이터에서 동적 가중치를 적용하여 순차 패턴을 탐사하는 새로운 시퀀스 데이터 마이닝 기법에 대하여 제안한다. 제안하는 기법은 시간 순서에 의한 상대적인 동적 가중치를 사용하여 탐색해야 하는 후보 패턴을 줄여줄 수 있어 빈발한 시퀀스 패턴을 빠르게 찾을 수 있다. 이 기법을 사용하면 기존 가중치를 적용하는 방식보다 메모리 사용과 처리 시간을 줄여줘 매우 효율적이다.

1. 서론

순차 패턴 마이닝은 데이터 분석의 한 분야로써 시간 속성을 갖는 이벤트들의 집합에서 빈발하게 발생되는 이벤트들의 부분집합을 탐사하여 어떤 이벤트의 순차적인 발생 정보를 탐사하는 기법이다[1]. 순차 패턴 탐사의 대표적인 알고리즘인 *PrefixSpan*[2] 알고리즘은 후보 순차 패턴들의 집합을 생성하지 않고 패턴을 탐사하는 효율적인 알고리즘이다. 이는 후보 순차 집합을 생성하지 않고 데이터베이스를 한번만 검색하기 때문에 우수한 성능을 나타낸다. 그러나 이러한 방법들은 각 항목들의 중요도를 모두 같은 값으로 가정한 것이기 때문에 각 항목들의 중요도가 다르게 고려되어야 하는 현실에서는 활용되기 어려운 문제점을 가지고 있다.

가중치 패턴 마이닝(Weighted Pattern Mining)[3][4]은 항목들이 다른 중요도(가중치)를 가질 경우 가중치를 고려한 높은 빈발도를 나타내는 패턴을 찾아내는 마이닝 기법을 의미한다. 예를 들면, 상품에 대한 고객들의 구매 패턴은 시간, 환경에 따라 다르게 나타날 수 있고, 상품가격 등과 같이 항목마다 다른 특성을 가지고 있기 때문에 모든 항목마다 같은 가중치를 적용하는 기존의 알고리즘은 적용할 수 없다. 또한 사용자가 관심 있게 보는 이벤트와, 그와 연관된 이벤트를 잘 찾아내기 위해서는 가중치 패턴 마이닝이 필요하다.

최근에는 이벤트의 가중치가 일정하지 않은, 동적인 가중치를 적용한 마이닝 기법이 필요하다[5]. 특히, 시간, 환경에 따라 같은 이벤트에 대한 중요도가 달라지는 환경에서는 동적인 가중치를 적용한 효율적인 기법이 더욱 요구되고 있다. 예를 들어, 날씨 및 기후 데이터를 분석할 때 같은 온도 변화라도 봄이나 가을보다는 여름과 겨울에 더욱 민감하게 반응하여 사람들의 스트레스 지수나 자외선 지수에 더욱 영향을 준다. 또한, 병원에서 사람들의 질병이나 질환에 대한 이벤트를 분석할 때, 일반적인 감기보다 천식, 고열 등의 이벤트가 여름보다 겨울에 더욱 잘 나타나므로, 이러한 질환들과 관련 있는 이벤트를 탐색할 때에는 이벤트가 더욱 빈번하게 발생할 수 있는 기간이나 환경에서 중요도, 즉 가중치를 변화시켜 주어야 한다.

이 논문에서는 동적인 가중치를 고려하여 빈발 후보 패턴의 수를 줄여주는 순차 패턴 마이닝 기법을 소개한다. 제안하는 기법은 시간 순서에 의해 변화하는 중요도를 고려하여 상대적 최대가중치를 적용한 마이닝 기법으로 동적 가중치를 적용한 빈발한 순차 패턴을 찾아낼 수 있고 탐색해야 하는 빈발 후보 패턴의 수를 줄여준다.

이 논문의 구성은 다음과 같다. 2장에서는 순차 패턴 탐사 기본원리를 기술하고 기존 가중치 마이닝 방법을 적용하는 것에 대한 문제점을 논의한다. 3장에서는 동적 가중치 순차 패턴 탐사를 위해 빈발 후보 패턴의 수를 줄여주는 효율적인 마이닝 기법을 제안한다.

2. 관련 연구

순차 패턴 탐사는 실제적인 시간 변화에 따라 저장된 일련의 순서화된 요소 또는 사건으로 이루어진 순차 데이터베이스에서 공통적으로 빈발하게 발생하는 순차(요소, 사건, 패턴)를 탐사하는 데이터 마이닝 기법이다. 순차 패턴 마이닝의 대표적인 알고리즘인 *PrefixSpan*은 순차 패

※ 본 연구는 교육과학기술부와 한국연구재단의 지역혁신인력양성사업으로 수행된 연구결과임.

턴이 될 수 있는 후보 순차 패턴들의 집합을 생성하지 않고 패턴을 탐사하는 효율적인 알고리즘이다.

PrefixSpan 알고리즘의 탐사 방법은 다음과 같다. 먼저 "빈발 항목 집합의 공집합이 아닌 모든 부분 집합은 반드시 빈발하다"라는 데이터 마이닝의 중요한 원리인 *Apriori* 성질(Apriori property)[6]을 이용하여 순차 데이터베이스를 검색하여 빈발하게 발생된 1항목 패턴을 탐사한다. 다음으로, 발견된 각 빈발 1항목 패턴을 이용하여 1항목 패턴에 대한 투영 데이터베이스를 생성한다. 투영 데이터베이스란, 발견된 각 빈발 항목을 각 순차에서 접두부(prefix)로 정의하고 각 순차의 공통 접두부를 제외한 후 두부(suffix)를 나타낸 것이다. 즉, 각 순차에서 공통으로 포함되어 있는 패턴을 제거한 나머지 순차들의 정보라 할 수 있다. 이후에는 위와 같은 방법으로 접두부 항목에 의한 투영 데이터베이스에서 다시 빈발한 항목을 탐사하여 빈발한 항목을 접두부로 결정하고 재귀적으로 투영 데이터베이스를 생성해 나간다. 재귀적으로 투영 데이터베이스를 생성해 나가면서 패턴의 항목수를 늘려가고, 항목이 작은 패턴으로부터 항목이 많은 패턴으로 빈발한 순차 패턴을 탐사해 나간다. 탐사가 완료된 이후에는 완전한 빈발 순차 패턴들을 모두 탐사할 수 있다.

가중치 패턴 마이닝(Weighted Pattern Mining)은 항목들이 다른 중요도(가중치)를 가질 경우 가중치를 고려한 높은 빈발도를 나타내는 패턴을 찾아내는 마이닝 기법이다. 상품에 대한 고객들의 구매 패턴은 시간, 환경에 따라 다르게 나타날 수 있고, 상품가격 등과 같이 항목마다 다른 가중치를 설정해야하는 경우가 일반적이기 때문에 모든 항목에 대하여 다른 가중치를 적용하여 활용할 수 있다.

먼저 가중치에 대한 정의를 설명한다. 가중치(Weighted)는 트랜잭션 데이터베이스에서 항목의 중요성을 나타내는 지표[3][4]로 항목집합 $I = \{i_1, i_2, i_3 \cdots i_n\}$에 대하여 패턴 $P\{x_1, x_2, x_3 \cdots x_m\}$의 가중치 $Weight(P)$는 식 (1)과 같이 정의한다.

$$Weight(P) = \frac{\sum_{i=1}^{length(P)} Weight(x_i)}{length(P)} \quad (1)$$

패턴 P의 가중치 지지도(Weighted Support)는 식 (2)와 같다.

$$WeightSup(P) = Weight(P) \times Support(P) \quad (2)$$

$WeightSup(P)$ 의 값이 최소 임계값보다 클 때 패턴 P를 가중치 빈발 패턴이라고 한다.

하지만 가중치 패턴 마이닝은 각각의 발생한 항목에 대해 다른 가중치를 적용하기 때문에 빈발한 패턴의 부분 패턴은 빈발하다는 *Apriori* 성질을 만족하지 않는다.

<표 1> *Apriori* 성질을 만족하지 않는 예

항목	가중치	빈도수	가중치 지지도	GMAXW 적용
A	0.6	4	2.4	2.4
B	0.2	5	1.0	3.0
AB	(0.6+0.2)/2 = 0.4	3	1.2	1.8

항목 "*A*" 와 "*B*" 그리고 "*AB*" 라는 패턴이 있다. 패턴의 가중치와 빈도수를 다음 표 1와 같이 정의할 때, 가중치 지지도의 임계값을 1.2라 하면 항목 "*B*"는 가중치 빈발 패턴이 아니지만 "*AB*"는 가중치 빈발 패턴이 된다. 이는 *Apriori* 성질을 만족하지 않을 뿐만 아니라 정확한 패턴 탐사에도 부정확한 결과를 내보낼 수 있다. [3][4]에서는 시퀀스 패턴이 아닌 빈발 패턴 환경이지만 시퀀스 환경에도 적용할 수 있는 전역적 최대 가중치(GMAXW: Global Maximum Weight)를 설정하여 탐색 과정에서 *Apiori* 성질을 고려할 수 있도록 했다. 이 방법은 항목 "*A*" 의 가중치인 0.6을 모든 항목에 대한 최대 가중치로 설정하고 이를 이용해 다른 항목의 가중치 지지도를 구하면 항목 "*B*" 는 0.6 × 5 = 3.0, "*AB*"는 0.6 × 3 = 1.8 로 *Apriori* 성질을 만족하는 결과를 보여주게 된다.

하지만 GMAXW는 가중치가 변화하는 동적 가중치 환경에서는 각 아이템의 가중치 값이 트랜잭션 별로 다르므로 모든 트랜잭션에 같은 최대 가중치를 적용할 수 없다. 동적인 가중치 환경에서는 최대 가중치 값이 변화하여 GMAXW 값을 적용한 마이닝 결과가 달라질 수 있기 때문이다. 또한, GMAXW를 적용한 가중치 패턴 마이닝은 빈발 후보 패턴을 많이 만들어내 실제 빈발 패턴을 탐색하는데에 많은 시간을 소요한다. 즉, 기존 가중치 패턴 마이닝 기법[3][4]은 모든 데이터에 대해 정해진 최대가중치를 적용하기 때문에 가중치가 동적으로 변화하는 환경에서는 효율적인 데이터 처리를 할 수 없다. 장에서는 동적 가중치 순차 패턴 마이닝 기법에서 사용 가능한 상대적 최대 가중치(Relative Maximum Weight) 기준을 적용하여 빈발 후보 패턴 수를 줄인 효율적인 기법을 제안한다.

2. 동적 가중치 순차 패턴 마이닝 기법(DWSPM)

동적인 가중치 환경에서는 가중치의 값이 시간에 따라 변하기 때문에 실제 가중치를 구하기 위해서는 트랜잭션에서 각 아이템의 가중치와 지지도를 모두 구해야 한다. 이를 위해 정적 가중치 패턴 마이닝의 가중치 지지도 계산법과는 다른, 동적 가중치 패턴 마이닝의 계산법이 필요하다. 이를 위해 다음과 같은 식을 정의한다.

정의 1. 패턴들은 시간 순서에 의해 정의된 아이템셋으로, 발생한 순서대로 이루어진 항목집합 $I = \{i_1, i_2, i_3 \cdots i_n\}$에 대하여 패턴 P에 대한 동적인 가중치 지지도(Dynamic Weighted Support)는 다음과 같이 정의한다.

$$Dynamic\,WS(P) = \sum_{i=1}^{n} Weight_i(P) \times Support_i(P) \quad (3)$$

물론, 시퀀스 패턴은 시간 순서가 있는 패턴들의 조합이므로 "*AB*" 라는 패턴과 "*BA*"라는 패턴은 서로 다르다. 그러므로 "*AB*" 와 "*BA*" 의 가중치 지지도는 다를 수 있다.

<표 2> 동적 가중치를 적용한 데이터베이스의 예

TID	Transaction	Weight			
		A	*B*	*C*	*D*
T_1	*A, B*	0.4	0.6	0.1	0.2
T_2	*D*				
T_3	*A, B, C, A*				
T_4	*B, A, C, A*	0.5	0.7	0.7	0.3
T_5	*B, A, B, C, A*				
T_6	*B, B*				
T_7	*B, A*	0.2	0.2	0.2	0.9
T_8	*D, B*				
T_9	*A, B, A*				

동적인 가중치 순차 데이터베이스는 표 2 과 같다. 각각의 트랜잭션 데이터는 순차적 성질을 가지고 있으며, 시간 순서에 의해 나열된다. 가중치 값은 각각의 트랜잭션마다 다른 가중치를 가지며 표 2 의 데이터에서는 편의적으로 3개의 트랜잭션 마다 다른 가중치를 적용하였다.

먼저 1항목 아이템의 가중치 빈도수를 구하기 위해 *GMAXW*를 사용한다. 표 2에서 각각의 아이템 중에 최대 가중치 값을 갖는 항목의 가중치는 "*D*" 항목의 가중치 0.9가 *GMAXW*가 된다. 이를 이용하여 최소 임계값이 1.6 일 때 마이닝하는 과정에서 각각 계산한 가중치 빈도수는 *A*:0.9 × 6 = 5.4, *B*:0.9 × 8 = 7.2, *C*:0.9 × 3 = 2.7, *D*:0.9 × 2 = 1.8 이 된다. 이는 모든 1항목 아이템들은 가중치 빈발 패턴이 될 수 있다는 것을 의미하고, 그 후에 각 항목에 대하여 Projected DB 생성을 통한 탐색을 실행한다.

<표 3> 아이템 B에 대한 투영 데이터

prefix	projected DB	weight		
		A	*B*	*C*
B	*C, A*	0.4	0.6	0.1
	A, C, A *A, B, C, A* *B*	0.5	0.7	0.7
	A *A*	0.2	0.2	0.2

projected DB 란 prefix 항목에 대한 투영데이터로써 prefix 항목과 시간 순서에 의해 연관된 아이템들을 열거한 것이다. projected DB 는 항목 "*B*" 이후에 발생한 아이템(Postfix) 들을 의미하므로 불필요하게 처리할 항목들을 줄여준다. *B*와 관련 있는 아이템은, 표 3에서 투영된 세 번째 트랜잭션 < *A B C A* >에서 *A*, *C* 두 아이템인 것을 알 수 있다. 투영된 데이터에서 최대 가중치 *GMAXW*로 정한 *D*:0.9 대신 *B*와 시간순서에 의해 관련 있는(postfix) 아이템 *A*, *B*, *C* 의 가중치 중에 최대 가중치인 *C*:0.7을 사용하여도 *Apriori* 성질을 만족할 수 있다. 0.7의 가중치를 사용하면 *A*는 0.7 × 5 = 3.5, *B*는 0.7 × 2 = 1.4, *C*는 0.7 × 3 = 2.1이 된다. 그러면 임계값인 1.6를 만족하지 못하는 *B*, 즉 패턴 *BB*는 후보에서 제외된다. 만약, 초기 최대 가중치인 0.9로 계산했다면 "*BB*" 는 가중치 빈도수 0.9 × 2 = 1.8 로 불필요하게 빈발후보패턴이 될 수 있다. 이같이 불필요한 항목들을 초기에 가지치기하기 위하여 사용하는, 시퀀스 환경에 적용 가능한 상대적 최대 가중치(Relative Maximum Weight)를 정의한다.

정의 2. 상대적 최대 가중치(Relative Maximum Weight) : 시퀀스 데이터에서 Projected DB를 생성할 때 prefix 항목과 연관이 있는(relative) 항목, 즉 postfix 항목 중에 제일 큰 가중치 값을 가진 항목을 찾아서 그의 가중치 상대적 최대 가중치(RMAXW)로 정의한다.

이를 통하여 불필요한 후보항목을 생성하지 않고, 시퀀스 환경에서 가중치 값을 적용할 때 *Apriori* 성질을 만족할 수 있다.

<표 4> 빈발후보패턴과 가중치 빈발 패턴 확인

후보 패턴	동적 가중치 계산	결과
A	(0.4×2) + (0.5×2) + (0.2×2) = 2.2	빈발
AA	((0.4+0.4)/2×1) + ((0.5+0.5)/2×2) + ((0.2+0.2)/2×1) = 1.1	빈발하지 않음
AB	((0.4+0.6)/2×2) + ((0.5+0.7)/2×1) + ((0.2+0.2)/2×1) = 1.8	빈발
ABA	((0.4+0.5+0.4)/3×1) + ((0.5+0.7+0.5)/3×1) + ((0.2+0.2+0.2)/3×1) = 1.2	빈발하지 않음
AC	((0.4+0.1)/2×1) + ((0.5+0.7)/2×2) = 1.45	빈발하지 않음
ACA	((0.4+0.1+0.4)/3×1) + ((0.5+0.7+0.5)/3×2) = 1.43	빈발하지 않음
B	(0.6×2) + (0.7×3) + (0.2×3) = 3.9	빈발
BA	((0.6+0.4)/2×1) + ((0.7+0.5)/2×2) + ((0.2+0.2)/2×2) = 2.1	빈발
BC	((0.6+0.1)/2×1) + ((0.7+0.7)/2×2) = 1.75	빈발
BCA	((0.6+0.1+0.4)/3×1) + ((0.7+0.7+0.5)/3×2) = 1.63	빈발
C	(0.1×1) + (0.7×2) = 1.5	빈발하지 않음
CA	((0.1+0.4)/2×1) + ((0.7+0.5)/2×2) = 1.45	빈발하지 않음
D	(0.2×1) + (0.9×1) = 1.1	빈발하지 않음

이와 같이 모든 후보 시퀀스 패턴들을 찾으면 표 4와 같이 실제 가중치를 이용하여 임계값과 비교하면 실제 빈발 시퀀스 패턴이 나오게 된다. 표 4에서 후보패턴 "*BC*", "*BCA*" 는 빈발하지만 후보패턴 "*C*" 는 빈발하지 않다. 이는 가중치 패턴 탐색에서 빈발한 항목의 부분 집합은 모두 빈발하다는 *Apriori* 성질을 만족하지 않는다는 것을 보여준다.

<표 5> 기존 GMAXW를 적용시 만들어지는 불필요한 후보패턴

후보 패턴	개수
AA, ABA, ABC, ABCA, AC, ACA, BAA, BAC, BACA, BB, C, CA, D	13개

또한 최대 가중치 GMAXW 값을 0.9로 정하고 모든 패턴에 대해 상대적 가중치 값이 아닌 GMAXW 값을 적용하면 표 5 과 같은 총 13개의 불필요한 빈발 후보 패턴을 생성하고 표 4 의 빈발하지 않은 후보패턴의 수인 7개보다 6개 더 많은 후보 패턴을 생성한다. 그리고 각각의 트랜잭션 길이가 길어지고 아이템 수가 많아지면 이는 더 많은 수의 빈발 후보 패턴을 생성하여 처리 속도에 악영향을 미친다. RMAXW를 사용하면 불필요한 후보 패턴을 GMAXW를 사용할 때보다 40~50% 줄여주어 빠른 탐색 속도로 빈발한 동적 가중치 패턴을 찾아준다.

5. 결론

이 논문에서는 시간에 따라 가중치가 변하는 동적 가중치 빈발 패턴 탐사 기법으로 빈발 후보 패턴의 생성 및 탐색을 줄여주는 기법을 제안하였다. 기존 데이터 마이닝은 시간이 지남에 따라 가중치의 값이 달라지는 것을 반영하지 않아 현실 세계에서 일어날 수 있는 동적인 환경에 적용하기 어려웠다. 이 논문에서는 발생할 수 있는 이벤트에 각각의 가중치를 부여하여 효율적인 마이닝이 가능하도록 상대적인 동적 가중치 계산 방식을 적용하였다. 이는 사용자가 정보를 얻고자 하는 중요 이벤트를 용이하게 마이닝 할 수 있으며 이를 실제 환경에 적용 가능하도록 할 수 있다. 향후 연구로는 실제 스트림 데이터에서 동적 가중치 마이닝 기법을 적용하여 기존의 알고리즘과 비교할 수 있도록 한다. 그리고 알고리즘 설계 시 발생할 수 있는 문제점과 보안할 사항을 확인할 수 있도록 한다.

참고문헌

[1] R. Agrawal and R. Srikant. "Mining sequential patterns", In Proc. 1995 Int. Conf. Data Engineering(ICDE'95), pp.3-14, 1995. 04

[2] J. Pei, J. Han, B. M. Asl, H. Pinto, Q. chen U. Dayal, and M. Hus, "PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth", In Proc. Int. Conf. Data engineering(ICDE'01), pp.215-226, 2001.

[3] U. Yun, J.J. Leggett, "WFIM: weighted frequent itemset mining with a weight range and a minimum weight", Proc. of the Fourth SIAM Int. Conf. on Data Mining, USA, PP.636-640, 2005.

[4] C.F. Ahmed, S.K. Tanbeer, B.-S. Jeong and Y.-K. Lee, "Mining weighted Frequent Patterns in Incremental Databases", Proc. of the 10th Pacific Rim Int. Conf. on Artificial Intelligence, pp.933-938, Dec. 2008.

[5] 정병수, Ahmed Farhan, "Prefix-트리를 이용한 동적 가중치 빈발 패턴 탐색 기법", 정보처리학회 논문지D 제17-D권 제4호, pp253-258, 2010. 8

[6] R. Agrawal and R. Srikant. "Fast algorithms for mining association rules", In Proc. 1994 Int. Conf. Very Large Data Bases(VLDB'94), pp.487-499, 1994. 09