ISSN(Print): 2287-5905

http://dx.doi.org/10.3745/KTSDE.2013.2.2.137

A Sequential Pattern Mining based on Dynamic Weight in Data Stream

Pilsun Choi† · Hwan Kim† · Daein Kim†† · Buhyun Hwang†††

ABSTRACT

A sequential pattern mining is finding out frequent patterns from the data set in time order. In this field, a dynamic weighted sequential pattern mining is applied to a computing environment that changes depending on the time and it can be utilized in a variety of environments applying changes of dynamic weight. In this paper, we propose a new sequence data mining method to explore the stream data by applying the dynamic weight. This method reduces the candidate patterns that must be navigated by using the dynamic weight according to the relative time sequence, and it can find out frequent sequence patterns quickly as the data input and output using a hash structure. Using this method reduces the memory usage and processing time more than applying the existing methods. We show the importance of dynamic weighted mining through the comparison of different weighting sequential pattern mining techniques.

Keywords : Weight Pattern Mining, Dynamic Weight, Sequential Pattern Mining

스트림 데이터에서 동적 가중치를 이용한 순차 패턴 탐사 기법

최 필 선† · 김 환† · 김 대 인†† · 황 부 현†††

요 약

순차 패턴 탐사 기법은 순서를 갖는 패턴들의 집합 중에 빈발하게 발생하는 패턴을 탐사하는 기법이다. 순차 패턴 탐사 분야 중에 동적 가중치 순차 패턴 탐사는 가중치가 시간에 따라 변화하는 컴퓨팅 환경에 적용 가능한 탐사 기법으로 동적인 가중치 변화를 탐색 과정에 적용하여 다양한 환경에서 활용 가능하다. 이 논문에서는 다양한 순차 데이터가 들어오는 스트림 환경에서 동적 가중치를 적용하여 빈발한 이벤트들을 탐사하는 새로운 순차 패턴 탐사 기법을 제안한다. 제안하는 기법은 시간 순서에 의한 상대적인 동적 가중치를 사용하여 탐색해야 하는 후보 패턴을 줄여주고 해시 구조를 통한 데이터 입출력으로 빈발한 순차 패턴을 빠르게 탐사할 수 있다. 이 기법을 사용하면 기존 가중치를 적용하는 방식보다 메모리 사용과 처리 시간을 줄여줘 매우 효율적이다. 제안하는 기법은 다른 가중치 순차 패턴 탐사 기법과의 비교를 통해 동적 가중치 탐사 기법의 중요성을 보인다.

키워드 : 가중치 패턴 마이닝, 동적 가중치, 순차 패턴 마이닝

1. 서 론

순차 패턴 탐사는 시간 속성을 갖는 이벤트들의 집합에서 빈발하게 발생하는 이벤트들의 부분집합을 탐사하여 이벤트들의 순차적인 발생 정보를 탐사하는 기법이다[1]. 대표적 기법인 *PrefixSpan*[4]은 후보 순차 패턴들의 집합을 생성하지 않고 패턴을 탐사하는 효율적인 기법이다. 이는 후보 순차 집합을 생성하지 않고 데이터베이스를 한번만 검색하기 때문에 우수한 성능을 나타낸다. 그러나 *PrefixSpan* 기법은 각 항목들의 가중치를 모두 같은 값으로 가정한 것이기 때문에 각 항목들의 가중치가 다르게 고려되어야 하는 현실에서는 활용되기 어려운 문제점을 가지고 있다.

가중치 패턴 탐사(weighted pattern mining)[6, 7]는 항목들이 다른 가중치를 가질 경우 가중치를 고려한 높은 빈발도를 나타내는 패턴을 탐사하는 기법을 의미한다. 예를 들면, 고객들의 상품 구매 패턴은 시간, 환경에 따라 다르게 나타날 수 있고, 상품 가격 등과 같이 항목마다 다른 특성을 가지고 있기 때문에 모든 항목마다 같은 가중치를 적용하는 기존의 기법은 적용할 수 없다. 또한 사용자가 관심있게 보는 이벤트와 그와 연관된 이벤트를 탐사하기 위해서는 가중치 패턴 탐사가 필요하다.

최근에는 한정된 데이터베이스를 가지고 분석을 하기보다 시간속성을 갖는 스트림 환경의 데이터베이스를 가지고 유용한 자료를 탐사해 내는 방식이 필요하다. 특히, 각각의 시간 순서에 따라 발생하는 항목들에 대해 다른 가중치를 적

※ 본 연구는 교육과학기술부와 한국연구재단의 지역혁신인력양성사업으로 수행된 연구결과임.

† 준 회 원 : 전남대학교 전자컴퓨터공학부 석사과정
†† 정 회 원 : 전남대학교 전자컴퓨터공학부 시간강사
††† 종신회원 : 전남대학교 전자컴퓨터공학부 교수

논문접수 : 2012년 9월 12일
수 정 일 : 1차 2012년 11월 19일
심사완료 : 2012년 12월 6일

* Corresponding Author : Buhyun Hwang(bhhwang@jnu.ac.kr)

용하는 가중치 패턴 탐사 기법이 필요하다. 왜냐하면, 현재 중요한 이벤트가 미래에는 다른 가중치가 적용되어 덜 중요한 이벤트가 될 수 있기 때문이다. 예를 들어, 주식데이터에서 한 종목의 가격이 상승하거나 하락할 때, 변동 폭이나 변동 가격에 따라서 같은 이벤트에 대해 다른 가중치를 부여할 수 있다. 또한, 현재 유가 변동률에 따른 변화폭이 큰 종목은 또 다른 가중치를 부여해야 한다. 즉, 같은 가격상승·하락이라도 시장 환경과 가격 변화에 따라 가중치가 달라지는 것이다.

이 논문에서는 스트림 환경에 적용 가능하여 동적인 가중치를 고려할 수 있는 순차 패턴 탐사 기법(DWSPM: Dynamic Weighted Sequential Pattern Mining)을 소개한다. 제안하는 기법은 스트림 환경에서 시간 순서에 따라 변화하는 가중치를 고려하여 상대적 최대 가중치를 적용한 탐사 기법으로 동적 가중치를 적용한 빈발 순차 패턴을 탐사할 수 있고 검사해야 하는 빈발 후보 패턴의 수를 줄여준다.

이 논문의 구성은 다음과 같다. 2장에서는 순차 패턴 탐사 기본원리를 기술하고 동적 스트림 데이터에 기존 탐사 기법을 적용하는 것에 대한 문제점을 논의한다. 3장에서는 스트림 환경에서 동적 가중치를 계산하는 방법을 논의하고 순차 패턴 탐사를 위한 효율적인 알고리즘을 제안한다. 4장에서는 제안 기법의 효율성 및 유효성을 검증하기 위한 각종 실험 결과를 논의하고 5장에서는 결론 및 향후 연구를 기술한다.

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 탐사 기법이다.

순차 패턴 탐사의 초기 기법인 *GSP*[2]는 탐사 단계에서 후보 순차를 생성하고 데이터베이스로부터 각각의 후보 항목에 대한 지지도를 구한다. 그리고 미리 정의한 최소 지지도 보다 작은 지지도를 갖는 후보 항목들을 후보에서 제외하여 빈도가 높은 항목을 만드는 빈발 패턴 탐사의 *Apriori* 알고리즘[3]을 응용하였다. 그리고 이 과정은 더욱 빈도가 높은 어떤 항목도 발견되지 않을 때까지 반복된다. 즉, 가능성 있는 모든 예상 순차 패턴을 생성하여 빈발 패턴을 탐사하는 것이다. 그러나 *Apriori* 알고리즘을 사용한 이 GSP 기법은 많은 데이터베이스 스캔을 필요로 하고 긴 순차 패턴을 탐사할 때 짧은 순차패턴으로부터 탐색해야 하기 때문에 후보 항목의 수가 많아진다.

순차 패턴 탐사의 대표적 기법인 *PrefixSpan*[4]은 순차패턴이 될 수 있는 예상 순차 패턴들의 집합을 생성하지 않고 패턴을 탐사한다. *PrefixSpan*의 탐사 방법은 다음과 같다. 첫 번째로 "빈발 항목 집합의 공집합이 아닌 모든 부분집합은 반드시 빈발하다" 라는 데이터 탐사의 중요한 원리인 *Apriori* 성질(Apriori property)[3]을 이용하여 순차 데이터베이스를 검색하여 빈발하게 발생된 1항목 패턴을 탐색한다. 두 번째, 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 후두부(suffix)를 나타낸 것이다. 즉, 각 순차에서 공통으로 포함되어 있는 패턴을 제거한 나머지 순차들의 정보라 할 수 있다. 세 번째는 위와 같은 방법으로 접두부 항목에 의한 투영 데이터베이스에서 다시 빈발한 항목을 탐색하여 빈발한 항목을 접두부로 결정하고 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.

HAPT(HAsh based Pattern Tree)[5] 기법은 의미 기반 윈도우와 해시 구조를 사용하여 빠른 자료 탐색이 가능한 기법이다. 해시 구조는 중복되지 않는 데이터들을 리스트화 시켜서 검색 비용을 최소화하여 일반 자료형보다 빠르게 자료를 탐사할 수 있는 데이터 구조이다.

그러나 이러한 기존의 순차 패턴 탐사 기법들은 각 항목들의 가중치(weight)를 모두 같은 값을 적용하므로 각 항목들의 가중치가 다르게 고려되어야 하는 현실에서는 활용되기 어려운 문제점을 가지고 있다. 이를 고려하여 각 항목의 가중치를 반영한 가중치 패턴 탐사 기법이 연구되었다.

가중치 패턴 탐사(weighted pattern mining)는 항목들이 다른 가중치를 가질 경우 가중치를 고려한 높은 가중치 지지도를 나타내는 패턴을 탐사하는 기법이다. 상품에 대한 가중치는 상품가격 등과 같이 항목마다 다른 가중치를 설정해야하는 경우가 일반적이기 때문에 모든 항목상품에 대하여 같은 가중치를 부여하는 것은 적합하지 않다.

Table 1. Example of home appliances sales data

Product	Price (won)	Sale (frequency)	Normalized Weight
Computer	650,000	400	0.65
Printer	250,000	500	0.25
refrigerator	950,000	150	0.95
microwave	200,000	240	0.2
iron	60,000	290	0.06
cellphone	500,000	440	0.5
digital camera	600,000	250	0.6

Table 1은 가전 매장에서의 제품 판매 데이터이다. 이 데이터에서 가중치 값은 제품의 실제 가격을 사용하여 가격과 비례하는 정규화한 값으로 정의한다. 가중치 값은 실제 가격을 사용하는 것보다 정규화를 통하여 0~1 의 값을 갖는 가중치 값을 사용하는 것이 유용할 수 있다. 이 데이터에서 가중치 패턴 탐사의 목적은 판매 데이터에서 정해진 최소 매출(임계값) 이상의 판매를 보인 모든 상품들의 집합을 찾아내는 작업으로 생각할 수 있다. 예를 들어, 다리미의 판매 횟수는 290회로 냉장고 판매 횟수인 150회 보다 더 많이 팔렸다. 이는 "판매수" 라는 빈발횟수만 비교하면 다리미가 유용한 제품이지만, 제품별 매출량을 비교하면 다리미(6만 ×

290회 = 1,740만원)보다 냉장고(95만 × 150회 = 1억4,250만원)가 더 많은 매출을 기록하고 있다. 즉, 각 항목에 대해 가격이나 사용자가 흥미를 가지는 이벤트를 기준으로 다른 가중치를 설정하여 실제 응용 분야에 적합하도록 하는 것이 가중치 패턴 탐사의 목적이다.

먼저 가중치에 대한 정의를 설명한다. 가중치(weight)는 트랜잭션 데이터베이스에서 항목의 중요성을 나타내는 지표[6, 7]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.

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

*Wsupport(P)*의 값이 최소 임계값보다 클 때 패턴 *P*를 가중치 빈발 패턴이라고 한다.

여기서 임계값은 시스템이 정의한 최소 가중치 지지도를 만족하는 값으로, 가중치 값이 상품 가격이면 판매 횟수라는 지지도를 적용하여 임계값은 상품 매출이 될 수 있다. 또한 주식 데이터에서 종목가격 변동 폭이 가중치 값이면 각각의 가중치에 따른 하락·상승 횟수를 적용하여 일정기간 동안 종목 가격의 변동 폭이 임계값이 된다.

가중치 패턴 탐사는 각각의 발생한 항목에 대해 다른 가중치를 적용하기 때문에 빈발한 패턴의 부분 패턴은 빈발하다는 *Apriori* 성질[3]을 만족하지 않는다. 그리고 가중치 패턴 탐사는 실제 가중치를 적용하여 빈발 후보 항목 중에 실제로 빈발한 항목을 탐사하는 방법이기 때문에 빈발 후보 항목을 적게 만드는 게 중요하다.

Table 2. Example of case that Apriori properties isn't satisfied

Item	Support	Frequency	Weighted Support
A	0.6	4	2.4
B	0.2	5	1.0
AB	(0.6+0.2) / 2= 0.4	3	1.2

항목 *"A"* 와 *"B"* 그리고 *"AB"* 라는 패턴이 있다. 패턴의 가중치와 빈도수를 다음 Table 2와 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. 가중치 빈발 패턴 탐사[6, 7]는 순차 패턴 탐사 환경에도 적용할 수 있는 전역적 최대 가중치(GMAXW: Global Maximum Weight)를 설정하여 탐색 과정에서 *Apriori* 성질을 고려할 수 있도록 했다. 이 방법은 항목 *"A"* 의 가중치인 0.6을 모든 항목에 대한 전역적 최대 가중치(GMAXW)로 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. 그러나 이 방식은 모든 항목들에 대해 최대 가중치 값을 적용함으로써 탐사해야 하는 항목들의 수를 불필요하게 늘릴 뿐만 아니라 빠른 처리를 요구하는 스트림 환경에서는 적용하기 어렵다.

순차 패턴 탐사에서 가중치를 적용한 *WSpan*[8]이 연구되었다. 이 기법은 가중치 범위(weight range)를 다양하게 정해두고 탐사를 실시할 때 전역적 최대 가중치(GMAXW) 대신 [8]의 *MaxW*(Maximum Weight)을 사용한다. *MaxW*는 다양한 가중치 목록 중에서 가장 많은 빈발 후보 항목을 만드는 최대 가중치 범위의 *MaxW*를 정하여 이를 최대 가중치로 정의하고 탐사를 실시한다. 이를 통해 전역적 최대 가중치(GMAXW)를 사용했을 때보다 빈발 후보 패턴이 적게 발생하고 탐색 소요 시간을 효과적으로 줄일 수 있다. 가중치 패턴 탐사에서 발생하는 항목들의 가중치 범위는 정해져있다. 이를 이용해서 가중치의 범위를 정하고 빈발 후보 패턴의 수를 줄일 수 있다. 그러나 [8]은 가중치가 시간에 따라 변화하는 동적인 가중치 환경에서는 적용할 수 없다. 가중치가 변화하면 각 항목의 가중치 범위가 달라지므로 *MaxW*의 값이 변화하기 때문이다. 가중치가 항목마다 정해져있는 정적 가중치 순차 패턴 탐사는 가중치의 범위를 정해놓고 이를 이용하여 빈발패턴을 탐사하는 것이 가능하지만 동적 가중치 순차 패턴 탐사에서는 가중치가 시간 순서에 따라 바뀔 수 있으므로 가중치의 범위를 정해놓을 수 없게 된다. 그러므로 동적 가중치 순차 패턴에는 새로운 탐사 기법이 필요하다.

Fig. 1은 주식 데이터에서 시간별로 등락폭을 도식화한 것이다. 주식에서 가격이 오르는 이벤트는 같은 이벤트로 볼 수 있다. 그러나 이것은 가격 변동의 폭을 고려하지 않고 오른 사실 자체를 이벤트로 간주하기 때문에 효율적인 분석을 할 수 없다. 왜냐하면 등락폭에 따라 다른 이벤트에 미치는 영향이 다르기 때문이다. 예를 들어 A라는 종목이 3% 오른 후 B 종목이 30% 오른다는 정보와, A 종목이 30% 오른 후 B 종목이 30% 오른다는 정보는 같은 정보로 볼 수 없다. 이는 어떠한 시점에서 발생한 이벤트가 B 종목의 상승에 큰 영향을 미쳤는지 구분해야 한다. 즉, 등락폭이

Fig. 1. Example of stock data

라는 동적으로 변화하는 가중치를 고려하지 않고서는 유용한 정보를 획득할 수 없는 것이다.

이 논문은 스트림 환경에서 동적인 가중치를 계산하여 동적 가중치 탐사를 하는데 목적이 있다. 3장에서는 스트림 환경에서 사용 가능한 상대적 최대 가중치(RMAX: Relative Maximum Weight) 기준을 적용하여 빈발 후보 패턴 수를 줄인 효율적인 알고리즘을 제안한다. 이 기법은 데이터의 유입이 무한하게 생성되는 빅데이터 환경에서도 적용 가능하다.

3. 스트림 환경에서의 동적 가중치 탐사

3.1 트랜잭션 생성 기법

스트림 환경은 어떤 현상에 대한 데이터를 획득하는 센서가 존재한다. 이 센서는 최대, 최소와 같은 범위를 가지고 있으며 발생하는 데이터의 범위를 예측할 수 있는데 이와 같은 데이터의 범위를 일정 단위로 나누어서 문자화 하는 것을 특성화(characterizing)라고 한다. 예를 들어 습도를 나타낼 때에 퍼센트(%) 단위를 나타내는데 각 측정값마다 범위를 구분하여서 문자항목으로 변환할 수 있다. 0~30%는 *A*, 30~70%는 *B*, 70~100%는 *C*와 같은 문자로 나타낼 수 있는데 이를 스트림 환경에서는 한 항목으로 본다. 즉, 이와 같은 항목들이 시간 순서에 따라 처리장치에 입력되는 시스템을 스트림 시스템으로 정의하고 이 시스템에서 데이터를 탐사하는 기법을 제안한다.

[5]에서는 스트림 환경에서 트랜잭션을 생성할 때 의미(meaning) 있는 이벤트를 중심으로 그전에 발생했던 항목들의 집합을 한 트랜잭션으로 정의하였다. 이 트랜잭션은 길이가 일정하지 않아 가변적으로 데이터를 수용할 수 있다는 장점이 있다.

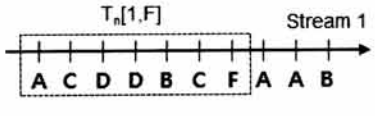

Fig. 2. Generation of transaction

[5]에서는 트랜잭션을 생성할 때 시스템이 관심을 갖는 중요한 이벤트를 타겟 이벤트(target event)로 설정하고, 이 이벤트가 발생했을 때 타겟 이벤트를 기준으로 사용자가 정의한 길이 안에 발생했던 항목들의 집합을 한 트랜잭션으로 생성한다. Fig. 2는 타겟 이벤트를 *F*로 정의하고 *F*를 기준으로 6 만큼의 길이 전에 발생한 항목의 집합을 한 트랜잭션으로 생성하였고 이는 T_n[1,F] : <*A C D D B C*> 로 표현한 것이다. 여기서 n은 트랜잭션 번호이고 1은 스트림이 발생하는 센서의 고유번호이다.

정의 1. 트랜잭션(transaction) : T[스트림번호, 타겟이벤트] 로 나타낸다. 각 항목들은 타겟 이벤트가 발생하기 전에 일어난 이벤트들의 집합이며, 항목 집합은 <A B C> 등과 같이 나타낸다. 스트림번호는 중복 가능하며, 무한한 트랜잭션 크기 증가를 막기 위해 일정한 시간 이후의 타겟 이벤트와 무관한 이벤트는 처리하지 않는다.

3.2 동적 가중치를 적용한 순차 패턴 탐사

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

정의 2. 패턴들은 시간 순서에 따라 정의된 항목들의 집합으로, 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.

$$DWS(P) = \sum_{i=1}^{N} Weight_i(P) \times Support_i(P) \quad (3)$$

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

[6, 7]에서는 가중치로 인한 부정확한 탐사를 막기 위해 전역적 최대 가중치(GMAXW)를 사용한다. 이는 빈발 패턴 탐사와 순차 패턴 탐사에 둘 다 적용 가능하다. 그러나 항목 개수가 2 이상인 패턴의 동적 가중치 지지도를 구할 때 전역적 최대 가중치(GMAXW)를 사용하면 불필요한 후보 패턴을 많이 만들어내게 된다. 이를 보안하기 위해서 가중치 빈발 패턴 탐사 분야에서는 국지적 최대 가중치(Local Maximum Weight)[9]를 사용한다. 그러나 국지적 최대 가중치는 빈발 패턴 탐사 기법에서 시간 순서가 없는 패턴을 탐사할 때 유용하지만 시간 순서가 있는 스트림 환경에서는 시간 순서에 따라 관련이 없는 항목들의 집합으로 이루어진 패턴을 제외하는 다른 규칙이 필요하다. 이 논문에서는 스트림 환경에 적합한 변화하는 가중치를 적용하는 동적 가중치 순차 패턴 탐사 기법(DWSPM: Dynamic Weighted Sequential Pattern Mining)을 제안한다.

먼저 1 항목 집합의 가중치 빈도수를 구하기 위해 전역적 최대 가중치(GMAXW)를 사용한다. Table 3에서 각각의 항목 중에 최대 가중치 값을 갖는 항목의 가중치는 *"D"* 항목의 가중치 0.9가 전역적 최대 가중치(GMAXW)가 된다. 이를 이용하여 최소 임계값이 1.6일 때 탐사하는 과정에서 각

Table 3. Example of database applied dynamic weight

TID	Transaction	Weight			
		A	B	C	D
T1[1,F] T2[1,F] T3[1,F]	A, B D A, B, C, A	0.4	0.6	0.1	0.2
T4[1,F] T5[1,F] T6[1,F]	B, A, C, A B, A, B, C, A B, B	0.5	0.7	0.7	0.3
T7[1,F] T8[1,F] T9[1,F]	B, A D, B A, B, A	0.2	0.2	0.2	0.9

각 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 1 항목 집합들은 가중치 빈발 패턴이 될 수 있다는 것을 의미하고, 그 후에 각 항목에 대하여 Projected DB 생성을 통한 탐색을 실행한다.

Table 4. Projected DB of item B

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 접두부(prefix) 항목에 대한 투영 데이터로써 접두부(prefix) 항목과 시간 순서에 따라 연관된 항목들을 열거한 것이다. projected DB 는 항목 *"B"* 이후에 발생한 항목(postfix)들을 의미하므로 불필요하게 처리할 항목들을 줄여준다. Table 4와 같이 투영된 데이터에서 최대 가중치 로 정한 *D*:0.9 대신 *B*와 시간순서에 따라 관련 있는 (postfix) 항목들 중에 최대 가중치인 *C*:0.7을 사용하여도 *Apriori* 성질[3]을 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*는 후보에서 제외된다. 만약, 초기 최대 가중치인 0.9로 계산했다면 *"BB"* 는 가중치 빈도수 0.9 × 2 = 1.8 로 불필요하게 빈발 후보패턴이 될 수 있다. 이같이 불필요한 항목들을 초기에 가지치기하기 위하여 사용하는, 순차 탐사 환경에 적용 가능한 상대적 최대 가중치(RMAXW: Relative Maximum Weight)를 정의한다.

정의 3. 상대적 최대 가중치(RMAXW: Relative Maximum Weight) : 순차 데이터에서 projected DB를 생성할 때 prefix 항목과 연관이 있는(relative) 항목, 즉 postfix 항목 중에 제일 큰 가중치 값을 가진 항목을 탐색해서 그의 가중치를 최대 가중치로 설정하고 이를 상대적 최대 가중치(RMAXW)로 정의한다.

이를 통하여 불필요한 후보항목을 생성하지 않고, 순차 패턴 탐사 환경에서 가중치 값을 적용할 때 *Apriori* 성질을 만족할 수 있다. 이와 같이 모든 후보 순차 패턴들을 탐사하면 실제 가중치를 이용하여 임계값과 비교하면 실제 빈발 순차 패턴이 나오게 된다.

Table 5에서 후보패턴 *"BC"*, *"BCA"* 는 빈발하지만 후보 패턴 *"C"* 는 빈발하지 않다. 이는 가중치 패턴 탐색에서 빈발한 항목의 부분 집합은 모두 빈발하다는 *Apriori* 성질을 만족하지 않는다는 것을 보여준다.

위에서 설명한 과정을 알고리즘으로 기술하면 다음과 같다.

순차 패턴 탐사는 빈발 패턴 탐사와는 다르게 트리를 구성하는 방식보다 *PrefixSpan*[4]의 투영 데이터(projected

Table 5. Frequent candidate patterns and checking weighted frequent pattern

Candidate Patterns	Dynamic weighted support calculation	Result (Y,N)
A	(0.4 × 2) + (0.5 × 2) + (0.2 × 2) = 2.2	Y
AA	((0.4 + 0.4)/2 × 1) + ((0.5 + 0.5)/2 × 2) + ((0.2 + 0.2)/2 × 1) = 1.6	Y
AB	((0.4 + 0.6)/2 × 2) + ((0.5 + 0.7)/2 × 1) + ((0.2 + 0.2)/2 × 1) = 1.8	Y
ABA	((0.4 + 0.6 + 0.4)/3 × 1) + ((0.5 +0.7 + 0.5)/3 × 1) + ((0.2 + 0.2 + 0.2)/3 × 1) = 1.23	N
AC	((0.4 + 0.1)/2 × 1) + ((0.5 + 0.7)/2 × 2) = 1.45	N
ACA	((0.4 + 0.1 + 0.4)/3 × 1) + ((0.5 + 0.7 + 0.5)/3 × 2) = 1.43	N
B	(0.6 × 2) + (0.7 × 3) + (0.2 × 3) = 3.9	Y
BA	((0.6 + 0.4)/2 × 1) + ((0.7 + 0.5)/2 × 2) + ((0.2 + 0.2)/2 × 2) = 2.1	Y
BC	((0.6 + 0.1)/2 × 1) + ((0.7 + 0.7)/2 × 2) = 1.75	Y
BCA	((0.6 + 0.1 + 0.4)/3 × 1) + ((0.7 + 0.7 + 0.5)/3 × 2) = 1.63	Y
C	(0.1 × 1) + (0.7 × 2) = 1.5	N
CA	((0.1 + 0.4)/2 × 1) + ((0.7 + 0.5)/2 × 2) = 1.45	N
D	(0.2 × 1) + (0.9 × 1) = 1.1	N

Input data: Transaction Database with Dynamic Weights, Minimum threshold (δ)
Output data: Frequent Sequence Pattern List

GMAXW : Global Maximum Weight
RMAXW : Relative Maximum Weight

Begin
For All transaction Ti in Database
Insert Ti list to Item List L
Calculate GMAXW
Update weight header table WH
End For
For All item α_i in Item List L
If (frequency(α_i) × GMAXW) > δ then
Calculate RMAXW
Create Projected_db(α_i) PDi
Call Mining_Process(α_i, PDi, RMAXW)
End If
End For
End
Procedure Mining_Process(α, PD, RMAXW)
Begin
Create Projected_db by deleting item γ
from PD having frequency(γ) × RMAXW < δ
For All item β_i in Item List of PD
Calculate dynamic weighted support DWS
for itemset $\alpha\beta_i$
If DWS $\geq$ δ then
Add $\alpha\beta_i$ in Frequent Sequence Pattern List
End If
End

DB) 방식이나 해시 구조 방식[5]을 자주 사용한다. 스트림 환경은 시간 순서에 의한 데이터 처리를 하기 때문에 빈발도나 지지도에 의한 정렬에 의해서 패턴을 탐사하는 빈발 패턴 탐사 방식에서는 적합하지 않다. 이 논문에서는 해시 구조를 사용하여 빈발 순차 패턴을 갱신하는 *HAPT*(HAsh based Pattern Tree)[5]를 이용한 동적 가중치 패턴 탐사 기법을 설명한다.

다음은 *HAPT*를 이용하여 해시 구조로 자료를 입력하는 과정이다. 해시 구조 중 하나인 맵(Map) 구조는 빅데이터 처리 기법에서 유용하게 쓰이는 자료구조로써 하둡(Hadoop)[10], 소셜 네트워크 데이터 등과 같은 대용량의 자료 처리에 적합하다. 스트림 환경은 무한한 자료의 입력으로 인해 빅데이터 처리 방식을 활용해야 한다.

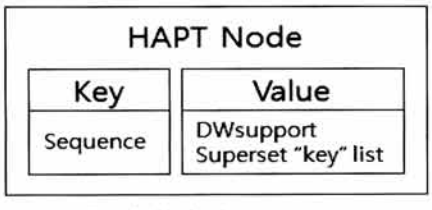

Fig. 3. Node data structure

*HAPT*에서 각각의 순차마다 생성되는 노드의 자료 구조는 Fig. 3과 같다. 맵 구조에서 자료탐색 시 검색값이 되는 키 값이 순차 이름이고, 보유하고 있는 값은 동적 가중치 지지도(DWsupport) 값과 후발생 노드를 가리키는 슈퍼노드 키 목록이다.

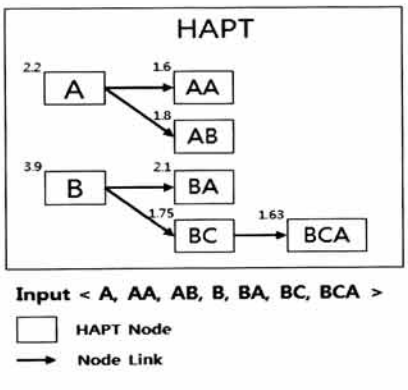

Fig. 4. Pattern matching results

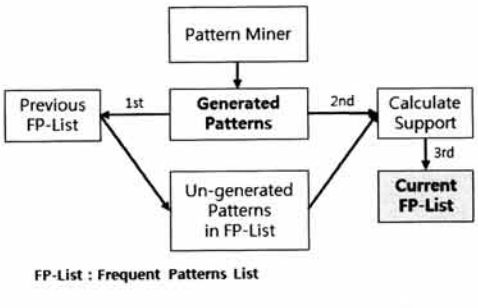

Fig. 5. Method to update frequent pattern

Fig. 4과 같이 각각의 순차 이름과 순차관계, 동적 가중치 지지도를 설정하고 *HAPT*의 빈발 순차 패턴 갱신 알고리즘을 실행하면 실시간으로 빈발하거나 발생 가능한 패턴을 노드 링크를 이용하여 예측 가능하다. 갱신 방법은 Fig. 5와 같이 임계값을 만족한 순차 패턴을 현재 생성된 패턴 목록에 넣고 과거에 빈발했던 패턴들과 비교를 통해, 현재에도 빈발하면 빈발 항목에, 현재에는 빈발하지 않다면 과거 빈발 패턴 목록에 추가한다.

빈발 순차 패턴을 갱신하는 방법을 알고리즘으로 기술하면 다음과 같다.

```
Input data: new Sequence(S)
Output data: Frequent Patterns List(FPL)

GPL : Generated Patterns List
UGPL : UnGenerated Patterns List
δ : Minimum threshold

begin
add S in GPL
for patterns Pn in FPL
    If GPL do not have Pn
        add Pn in UGPL
    End If
End For
clear FPL
Calculate WSupport Pn in GPL, UGPL
If Pn.support >= δ
    add Pn in FPL
end
```

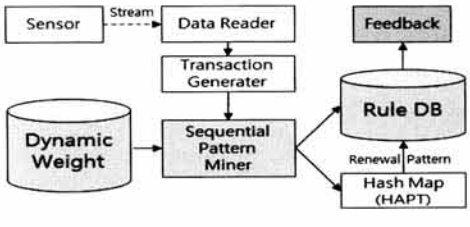

Fig. 6. Flowchart of DWSPM

4. 실험 결과 및 분석

본 절에서는 제안한 동적 가중치 순차 패턴 탐사 기법(DWSPM) 기법과 다른 순차 패턴 기법과의 성능평가를 수행한다. 기존의 순차 패턴 탐사 기법에는 가중치 적용을 위해 최대 가중치(GMAXW)를 이용하여 가중치를 계산한다. 이 논문 방법의 성능을 평가하기 위하여 실제 물건 판매 패턴을 기반으로 한 순차 데이터[1, 2, 4, 11]를 사용하였다. 실험 환경은 Intel Core 2 Duo E7300 2.66GHz, RAM 4GB, 이며 JAVA JDK 1.6 언어로 작성되어 수행한다.

실험에 사용한 데이터셋의 특징은 Table 6, Table 7과 같다. 사용하는 데이터 셋은 GSP[2], PrefixSpan[4] 등의 순차 패턴 탐사에 쓰인 대표적인 데이터 집합으로 실제 생활에서 물건 판매 데이터를 이용한 큰 용량의 합성 데이터(synthetic data)이다[4, 11]. C1kT8S8I8 데이터 집합은 트랜

Table 6. Dataset parameter

Parameters	Contents
C	Average number of transacitons
T	Average number of items
S	Average length of maximal potenially large Sequences
I	Average size of Itemsets in maximal potentially large sequences

Table 7. Parameter setting

Name	|C|	|T|	|S|	|I|	Size
C1kT8S8I8	1 million	8	8	8	281.0MB

잭션 수가 1백만개(C1k), 한 트랜잭션 안에 들어있는 항목 수는 평균 8개(T8)이며, 트랜잭션의 평균 길이는 8(S8)이다. 이 데이터셋의 설정 값은 긴 순차 패턴에서 낮은 지지도 임계값을 측정할 수 있는 좋은 조합이다[4]. 그러나 해당 데이터에는 가중치 값이 정해지지 않았기 때문에 확률 변수와 랜덤(Random) 함수를 이용하여 0.01~0.99까지의 값을 생성하여 추가하였다. 실험은 기존 순차 패턴 탐사 기법과 제안하는 기법의 성능 비교를 실시하고, 상대적 최대 가중치(RMAXW)의 성능 및 빈발 후보 패턴이 생성된 수를 비교하여 상대적 최대 가중치의 우수성을 보인다.

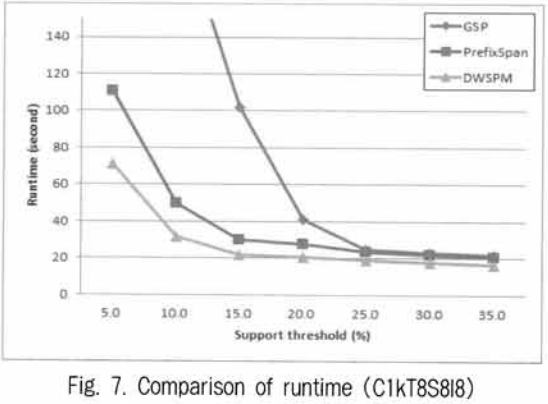

Fig. 7. Comparison of runtime (C1kT8S8I8)

Fig. 7은 기존의 순차패턴 기법에 C1kT8S8I8 데이터 셋을 탐사한 수행 시간을 이 논문에서 제안한 DWSPM 기법과 비교한 실험이다. 이 실험에서 DWSPM 기법은 GSP[2]과 PrefixSpan 기법[4]보다 수행시간에서 월등히 좋은 결과를 보여준다. 이는 DWSPM의 방식은 투영데이터를 이용하여 데이터를 탐색해 나가는 방식을 사용하여 PrefixSpan 탐사 기법과 유사하지만 기존 PrefixSpan 방식은 빈발 후보 패턴을 생성하여 실제 지지도 임계값을 계산하는 시간과, 불필요한 빈발 후보 패턴과 연관 있는 또 다른 후보 패턴을 탐사하기 때문에 월등한 시간 차이가 생기는 것으로 보인다. GSP 방식은 *Apriori* 탐색 기법[3]을 사용하기 때문에 빈발 후보 집합을 자주 만들어 내어 낮은 지지도의 가중치 패턴 탐사 과정에서 오랜 시간이 걸리는 것으로 나타났다.

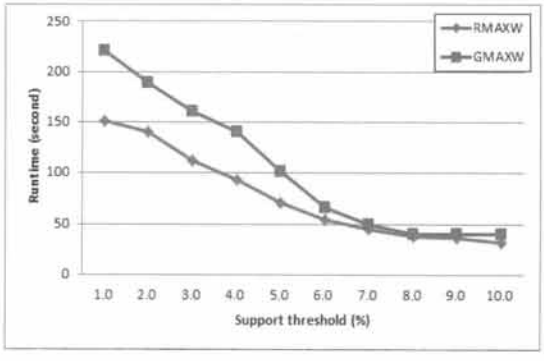

Fig. 8. Comparison of RMAXW and GMAXW runtime

Fig. 8은 DWSPM 기법에서 가중치를 계산하는 방식인 상대적 최대 가중치(RMAXW)와 기존의 전역적 최대 가중치(GMAXW)를 이용했을 때 속도 비교를 한 것이다. 실험은 DWSPM 방식에 가중치 계산 방식을 RMAXW와 GMAXW를 각각 적용하여 비교하였다. 실험 결과 낮은 지지도 임계값 상태일 때 상대적 최대 가중치(RMAXW) 방식이 월등히 높은 속도를 보였다. 이는 지지도가 낮을수록 빈발 후보 패턴을 적게 만들어내는 상대적 최대 가중치(RMAXW) 방식이 많은 빈발 후보 패턴을 생성하는 전역적 최대 가중치(GMAXW) 방식보다 실제 패턴을 분별하는 과정에서 짧은 시간이 걸린 것으로 보인다. 이를 통하여 빈발 후보 패턴에서 실제 빈발 패턴을 분별하는 과정이 많을수록 실제 알고리즘 성능에 큰 영향을 미치는 것을 확인할 수 있다.

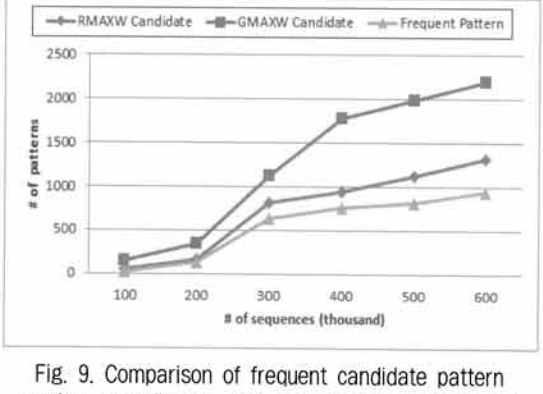

Fig. 9. Comparison of frequent candidate pattern number according to weight calculation method, and actual frequent pattern number

Fig. 9는 상대적 최대 가중치(RMAXW)와 전역적 최대 가중치(GMAXW)가 생성하는 빈발 후보 패턴의 수와 실제 빈발한 패턴의 수를 탐사하는 트랜잭션의 수에 따라 비교한 것이다. 실험은 DWSPM 방식에서 가중치 계산 방식을 각각 달리 적용한 것이다. 상대적 최대 가중치(GMAXW) 방식은 실제 빈발 패턴 수의 약 200%의 후보 패턴을 생성하고 상대적 최대 가중치(RMAXW)는 실제 빈발 패턴 수의 약 120%의 후보 패턴을 생성한다. 전역적 최대 가중치(GMAXW)는 알고리즘은 단순하지만 빈발 후보 패턴을 많이 만들어내고 메모리를 많이 사용한다. 그에

비해 상대적 최대 가중치(RMAXW)는 빈발 후보 패턴을 적게 만들어 메모리 사용량을 줄여주고 성능을 향상시키는 것을 볼 수 있다.

5. 결 론

이 논문에서는 시간에 따라 가중치가 변하는 동적 가중치 순차 패턴 탐사 기법으로 스트림 환경에 적용 가능한 동적 가중치 계산 알고리즘을 제안하였다. 기존 스트림 환경에서의 데이터 탐사는 모든 항목들의 가중치를 고려하지 않고 중요 이벤트와 중요하지 않은 이벤트를 같은 빈발함으로 계산하였지만 이는 효과적인 탐사 기법이 아니다. 그리고 시간이 지남에 따라 가중치의 값이 달라지는 것을 반영하지 않아 스트림 환경에 적용하기 어려웠다. 이 논문에서는 발생할 수 있는 이벤트에 각각의 가중치를 부여하여 탐사하고, 또한 스트림 환경에 적용 가능하도록 동적 가중치 계산 방식을 적용하였다. 그리고 상대적 최대 가중치를 이용한 빈발 후보 패턴 수를 획기적으로 줄일 수 있음을 알 수 있다. 이는 사용자가 정보를 얻고자 하는 중요 이벤트를 빠르게 탐사할 수 있으며 이를 실제 환경에 적용 가능하도록 할 수 있다. 향후 연구로는 스트림 환경에서의 동적 가중치 탐사 기법을 다른 환경의 패턴 탐사 기법에 적용하여 다양한 환경에 적용 가능한 효율적인 탐사를 할 수 있도록 한다. 그리고 이를 실제 적용 가능한 시스템을 개발하고 실제 수치를 통한 타 분야 연구에 도움을 줄 수 있도록 한다.

참 고 문 헌

[1] R. Agrawal and R. Srikant. "Mining sequential patterns", in *Proc. 1995 Int. Conf. Data Engineering (ICDE'95)*, pp.3-14, 1995. 4.

[2] R. Srikant and R.Agrawal, "Mining Sequential Patterns : Generalizations and Performance Inprovement", in *Fifth Int. Conference on Extending DataBase Technology (EDBT'96)*, Avignon, France, 1996.

[3] 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. 9.

[4] 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.

[5] J. I. Kim, "Real-time Sequential Pattern Mining for USN System", in *Proc. 2012 Int. Conf. on Ubiquitous Information Management and Communication (ICUIMC'12)*, Article No.36, 2012. 2.

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

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

[8] Unil Yun, and John J. Legget, "WSpan: Weighted Sequential pattern mining in large sequence database", in *International IEEE Conference Intelligent Systems*, pp.512-517, 2006. 9.

[9] B.S. Jeong, Ahmed Farhan, "Efficient Dynamic Weighted Frequent Pattern Mining by using a Prefix-Tree", *The KIPS Transactions: Part D*, Vol.17-D, No.4, pp.253-258, 2011.

[10] Apache Hadoop, http://hadoop.apache.org

[11] R. Agrawal and R. Srikant. Mining sequential patterns. Research Report RJ 9910, IBM Almaden Research Center, San Jose, California, October, 1994.

최 필 선

e-mail : pilddong@nate.com
2009년 전남대학교 전자컴퓨터공학부(학사)
2011년~현 재 전남대학교 전자컴퓨터 공학부 석사과정
관심분야 : Data Mining, Stream Data, Algorithm

김 환

e-mail : caeger@nate.com
2011년 전남대학교 전자컴퓨터공학부(학사)
2012년~현 재 전남대학교 전자컴퓨터 공학부 석사과정
관심분야 : Data Mining, Stream Data, Algorithm

김 대 인

e-mail : dikim@jnu.ac.kr
1998년 전남대학교 전산통계학과 (이학석사)
2006년 전남대학교 전산통계학과 (이학박사)
2004년~현 재 전남대학교 전자컴퓨터 공학부 시간강사
관심분야 : Stream Data, Data Mining, Digital Contents

황 부 현

e-mail : bhhwang@jnu.ac.kr
1978년 숭실대학교 전산통계학과(학사)
1980년 한국과학기술원 전산학과(공학석사)
1994년 한국과학기술원 전산학과(공학박사)
1980년~현 재 전남대학교 전자컴퓨터 공학부 교수
관심분야 : Stream Data Mining, Distrubuted System, Distributed Database