DOI : 10.5392/JKCA.2011.11.1.056

스트림 데이터 환경에서 배치 가중치를 이용하여 사용자 특성을 반영한 빈발항목 집합 탐사

Discovering Frequent Itemsets Reflected User Characteristics Using Weighted Batch based on Data Stream

서복일, 김재인, 황부현
전남대학교 전자컴퓨터공학부

Bok-Il Seo(seo.boxer@gmail.com), Jae-In Kim(sereno3@naver.com),
Bu-Hyun Hwang(bhhwang@chonnam.ac.kr)

요약

스트림데이터는 무한하고 연속적인 특성을 지니고 있기 때문에 전체 데이터를 기반으로 빈발 항목 집합을 탐사하는 것은 어렵다. 이 때문에 데이터의 특성과 사용자의 특성을 반영한 특수한 데이터마이닝 방법이 필요하다. 이 논문에서는 사용자가 최근에 발생한 데이터에 더 많은 관심이 있다는 특성을 반영하여 빈발 항목을 탐사하는 FIMWB 방법을 제안한다. FIMWB는 과거 데이터의 발생 시점과 현재 시점과의 시간 간격에 따라 가변적인 가중치를 배치에 부여하여 최신 데이터에 더 많은 관심과 중요성을 반영한다. FP-Digraph는 FIMWB를 통해 탐사된 빈발 항목으로 그래프를 구성하여 빈발 항목 집합을 탐사한다. 실험 결과로 FIMWB 방법이 불필요한 항목의 생성을 감소시키고 트리기반(FP-Tree)의 빈발 항목 집합 탐사에 비해 제안하는 FP-Digraph 방법이 스트림 데이터 환경에 더 적합함을 알 수 있다.

■ 중심어 : | 스트림 데이터 | 가중치 | 빈발항목 집합 | 인터벌 | 배치 |

Abstract

It is difficult to discover frequent itemsets based on whole data from data stream since data stream has the characteristics of infinity and continuity. Therefore, a specialized data mining method, which reflects the properties of data and the requirement of users, is required. In this paper, we propose the method of FIMWB discovering the frequent itemsets which are reflecting the property that the recent events are more important than old events. Data stream is splitted into batches according to the given time interval. Our method gives a weighted value to each batch. It reflects user's interestedness for recent events. FP-Digraph discovers the frequent itemsets by using the result of FIMWB. Experimental result shows that FIMWB can reduce the generation of useless items and FP-Digraph method shows that it is suitable for real-time environment in comparison to a method based on a tree(FP-Tree).

■ keyword : | Data Stream | Weighted Value | Frequent Itemsets | Interval | Batch |

* 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2010-0005647)

접수번호 : #101206-010
접수일자 : 2010년 12월 06일
심사완료일 : 2011년 01월 03일
교신저자 : 황부현, e-mail : bhhwang@chonnam.ac.kr

I. 서 론

스트림 데이터는 실시간 시스템, 통신 네트워크, 인터넷 트래픽, 금융시장의 온라인 트랜잭션, 원격 센서 등의 동적인 환경(dynamic environment)에서 주로 생성된다. 전통적인 데이터인 정적 데이터(static data)와는 다르게 스트림 데이터는 크기가 무한하고 연속적이며 데이터의 경계가 없다. 이러한 데이터 스트림을 모두 저장하거나 여러 번 읽어 들인다는 것은 엄청난 데이터 크기 때문에 불가능하므로 이러한 환경에 적합한 마이닝 기법이 필요하다[3-7].

빈발 패턴(frequent pattern)은 데이터 집합에서 빈번하게 발생하는 패턴(항목집합, 부분순차, 부분구조)들이다. 빈발 패턴을 발견하는 일은 데이터 사이의 연관성, 상관성, 그리고 많은 흥미로운 관계를 탐사하는데 필수적인 역할을 한다. 빈발 패턴 마이닝(frequent patterns mining)은 스트림 데이터 마이닝 분야에서 가장 폭 넓게 연구되어지고 있는 분야로써 데이터베이스에서 빈번하게 발생하는 패턴들을 찾기 위한 다양한 알고리즘들이 제안되었다[1][2][4-7].

이 논문에서는 스트림 데이터 환경에서 사용자가 과거에 발생한 이벤트보다 최근에 발생한 이벤트에 대한 관심이 더 높다는 특성을 반영하여 빈발 항목을 탐사하는 FIMWB(Frequent Item Mining using Weighted Batch) 방법을 제안한다. FIMWB를 방법으로 탐사된 빈발 항목들은 FP-Digraph(Frequent Pattern Mining using Directed Graph)를 이용하여 빈발항목 집합으로 탐사된다. 제안하는 방법은 일정한 시간 구간을 배치로 정의하고 스트림데이터를 배치로 분할하여 각 배치에 가변적인 가중치를 부여한다. 배치에 대한 가중치 부여는 배치가 현재와 멀수록 작은 가중치를 부여하고 가까울수록 높은 가중치를 부여한다. 제안하는 마이닝 방법은 최근 데이터에 더 많은 중요성을 부여하는 응용에 적합하다.

이 논문의 구성은 다음과 같다. 2장은 스트림데이터 환경에서 빈발항목집합 탐사에 관한 관련연구에 대해서 기술하고, 3장은 사용자의 특성을 반영한 스트림 데이터 마이닝 기법에 관해서 기술한다. 4장에서는 실험을 통하여 제안 방법의 효율성을 분석하고, 끝으로 5장에서는 본 연구의 결론을 기술한다.

2. 관련연구

스트림 데이터는 무한하고 연속적이며 경계가 없기 때문에 전통적인 데이터베이스 시스템에서의 빈발항목 집합 탐사와는 다른 접근 방법이 필요하다. 첫째, 스트림데이터 환경에서 수집되는 데이터는 연속적이다. 따라서 연속적으로 발생하는 동일한 이벤트를 하나의 이벤트로 요약할 수 있는 기법이 필요하다. 둘째, 수집되는 스트림 데이터의 크기는 무한하므로 빈발항목 집합 탐사 시 생성되는 후보 항목의 수 역시 무한하다. 이러한 특성을 고려하여 일정 시간 구간으로 정의된 윈도우 단위로 분할하여 윈도우 내의 이벤트에 대해서 탐사가 이루어져야 한다[9][10].

[4]에서는 FP-Tree에 이동 윈도우 기법을 적용한 WFPMDS(Weighted Frequent Pattern Mining over Data Streams)방법을 제안하였다. WFPMDS 는 윈도우의 변화에 따라 트리의 형태를 변화시킴으로써 스트림 데이터의 무한하고 연속적인 특성에도 적용 시킬 수 있도록 했다. 그러나 WFPMDS 는 윈도우의 이동에 따라서 트리의 형태를 변형시켜야 한다는 부담이 존재하며 사용자의 특성을 반영하는 점에서 한계가 있었다.

[8]에서는 FP-Tree의 구조적인 단점을 극복하기 위해 빈발항목 집합 탐사의 또 다른 방법으로 FP-Graph를 제안하였다. FP-Graph는 그래프 기반의 빈발항목 집합 탐사 알고리즘으로써 FP-Tree에 비해 공간적으로는 효율적임을 보였으나 처리속도에서는 최소 지지도에 따라 FP-Tree에 비해 효율적이지 못하였다. 또한 전통적인 데이터베이스 환경에서 모든 데이터 항목을 대상으로 빈발항목 집합을 탐사하므로 스트림 데이터 시스템의 특성을 고려하지 않고 있다.

3. 사용자 특성을 반영한 빈발항목 집합

3.1 문제 정의

스트림 데이터는 전통적인 데이터마이닝과는 달리 전체 데이터를 대상으로 마이닝을 수행할 수 없다. 이 때문에 데이터와 사용자의 특성을 고려한 특수한 마이닝기법이 필요하다.

사용자는 오래된 정보보다는 새로운 정보에 더 많은 관심을 가진다. 인터넷 트래픽이나 주식시장의 경우 과거에는 빈발하게 발생하지 않았더라도 현재 시점에 빈발하게 발생하는 이벤트가 오히려 더 중요한 정보가 된다. 그러나 이러한 정보는 전체 데이터를 대상으로 마이닝을 수행했을 시에는 발견하기 어렵다. 따라서 과거에 발생한 데이터보다 현재시점에서 발생한 데이터를 중심으로 마이닝을 할 수 있는 기법이 필요하다[11].

기존의 연구에서는 사용자의 관심을 반영하지 않고 데이터의 특수성만을 고려하여 데이터항목의 중요도에 따라 데이터항목에 가변적인 가중치를 부여하는 마이닝에 대한 연구가 주로 이루어졌다. 따라서 마이닝 수행 시 대상이 되는 데이터항목이 과거에 발생한 데이터항목 이라 할지라도 현재 발생한 데이터항목과 동일한 가중치를 가지고 처리되는 문제점이 있었다[5][6].

이 논문에서는 사용자가 오래된 정보 보다는 새로운 정보에 더 관심이 있는 특성을 반영하여 발생 순서에 따라 배치에 가변적인 가중치를 부여하여 빈발항목 집합을 탐사하는 FIMWB 방법을 제안한다. 제안하는 방법은 정적 데이터에서와는 달리 모든 데이터를 마이닝 대상으로 할 수 없는 스트림 데이터 환경에서 사용자가 실제로 원하는 정보를 탐사하는데 유용하다. 또한 기존의 스트림 데이터 환경에서 빈발항목 집합을 탐사하기 위해 주로 이용되었던 트리 기반(FP-Tree)의 방법이 아닌 그래프 기반의 FP-Digaph 방법을 제안한다. 제안하는 FP-Digraph가 FP-Tree 보다 빠른 처리 속도를 보여 스트림 데이터 환경에서 빈발항목 집합을 탐사하는데 유용함을 보인다.

제안하는 방법은 두 가지 단계로 분류된다. 첫 번째 단계에서는 FIMWB 방법으로 빈발항목을 탐사한다. 두 번째 단계에서는 FIMWB를 통해 탐사된 빈발 항목을 이용하여 FP-Digraph를 통해 빈발항목 집합을 탐사한다.

3.2 FIMWB

이 절에서는 과거에 발생한 이벤트보다는 최근에 발생한 이벤트에 사용자가 더 많은 관심이 있다는 특성을 반영한 FIMWB 방법을 기술한다.

[그림 1]은 스트림 데이터를 배치와 윈도우 단위로 분할하여 표현한 그림이다. [그림 1]에서 현재시점을 t로 하고 현재시점과 배치n과의 시간 간격을 t-n (n≥0)으로 표기하며 n이 클수록 현재시점과 시간간격이 크다는 것을 의미한다. 발생한 이벤트들은 일정한 시간단위로 경계를 나누어 배치로 구분한다. 각 윈도우는 연속된 세 개의 배치들로 구성되며 마이닝은 윈도우 단위로 수행된다.

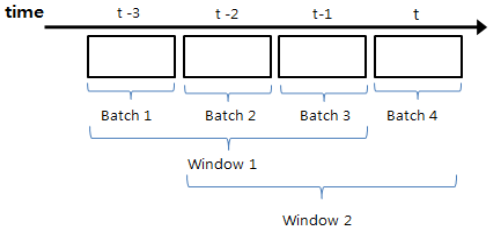


그림 1. 배치와 윈도우 단위로 분할된 스트림 데이터

표 1. 스트림 데이터의 트랜잭션

Batch	TID	이벤트(항목)
1	T_1	D, E, F
	T_2	C, D, E
2	T_3	B, C
	T_4	B, C, D
3	T_5	A, D
	T_6	A, B, C, E
4	T_7	B, C, D
	T_8	D, E

제안하는 FIMWB 방법은 연속된 세 개의 배치들을 하나의 윈도우로 구성하고 배치의 발생 순서에 따라서 사용자가 정의한 임의의 가중치를 부여한다. [표 1]은 데이터베이스에 포함된 트랜잭션들을 배치단위로 분할한 것이다. 윈도우 1 은 배치 1, 2, 3 을 포함하며 배치의 번호가 작을수록 과거에 발생한 배치임을 의미한다. 과

거에 발생한 순서대로 배치 1 에는 0.2, 배치 2 에는 0.3, 배치 3 에는 0.5 의 사용자 정의 배치 가중치를 부여한다. 윈도우 2 는 윈도우 1 에서 배치 1 을 제외하고 배치 4를 추가하여 배치 2, 3, 4를 포함하고 배치 2에는 0.2, 배치 3에는 0.3, 배치 4에는 0.5 의 사용자 정의 배치 가중치를 부여한다.

정의 1. 지지도 횟수(Support Count)
Support Count(X) = 데이터항목 X를 포함하는 트랜잭션의 수

정의 2. 배치 가중치(Weighted Batch)
Bwin : 하나의 윈도우에 포함된 배치의 수

$$\sum_{i=1}^{Bwin} Weight(B_i) = 1$$

정의 3. 항목 가중치 값(Item Weighted Value)
IWV(X) : 윈도우에 있는 데이터항목 X에 대한 배치 별 지지도 횟수와 배치 가중치의 곱

$$IWV(X) = \sum_{i=1}^{Bwin} suport\,Count(X in B_i) \times Weight(B_i)$$

정의 4. 최소 항목 가중치 값
(Weighted Minimum Item Value)

최소 항목 가중치 값은 항목의 관심도를 판단하는 최소 임계값으로서 만약 항목 X의 가중치 값이 최소 가중치 항목 값보다 크거나 같으면 관심도가 높은 항목으로 분류되고 그렇지 않으면 관심도가 낮은 항목으로 분류되어 가지치기 된다.

표 2. 윈도우에 따라 배치에 부여되는 가중치의 변화

윈도우	포함된 배치	배치 가중치
1	B_1	0.2
	B_2	0.3
	B_3	0.5
2	B_2	0.2
	B_3	0.3
	B_4	0.5

표 3. 항목 가중치 값

윈도우	아이템	배치 별 지지도 횟수	IWV
1	A	B_3: 2	1.0
	B	B_2: 2, B_3: 1	1.1
	C	B_1: 1, B_2: 2, B_3:1	1.3
	D	B_1: 2, B_2: 1, B_3: 1	1.2
	E	B_1: 2, B_3: 1	0.9
	F	B_1: 1	0.2
2	A	B_3: 2	0.6
	B	B_2 :2, B_3: 1, B_4: 1	0.9
	C	B_2: 2, B_4: 1	0.9
	D	B_2: 1, B_3: 1, B_4: 2	1.5
	E	B_3: 1, B_4: 1	0.8

[표 3] 은 [표 1]의 각 윈도우에 포함된 항목의 가중치 값이다. 각 항목의 가중치 값은 하나의 윈도우에 포함된 각각의 배치에 대해서 각 배치에 발생한 동일한 항목의 지지도의 횟수와 그 배치 가중치를 곱한 후 항목 가중치 값을 합산하여 얻어진다[5][6]. 예를 들어 항목 D의 경우, 배치 1에서 2번, 배치 2에서 1번, 배치 3에서 1번으로 총 4번 발생한다. 이를 항목 가중치 값의 합으로 표현하면 (2 × 0.2) + (1 × 0.3) +(1 × 0.5) = 1.2 로 계산된다.

FIMWB 에서는 IWV 가 최소 항목 가중치 값보다 작으면 사용자의 관심이 낮은 항목으로 판별하여 가지치기 한다. [표 3]에서 항목 A는 윈도우 1과 윈도우 2 에서 나타나는 지지도 횟수가 2 로 동일하지만 IWV(A) 는 1.0 과 0.6 으로 다른 수치를 보인다. 따라서 최소 항목 가중치 값을 0.8 이라고 했을 때 윈도우 1 에서는 항목 A가 관심도가 높은 항목으로 탐사되지만 윈도우 2 에서는 관심도가 낮은 항목으로 판별되어 가지치기(Prune)가 된다. 이처럼 현재시점에 관심도가 높은 항목만을 탐사함으로써 사용자의 요구에 맞추어 실시간으로 빈발항목을 탐사해야 하는 스트림 데이터 환경에 적합하다.

3.3 FP-Digraph

스트림 데이터 환경에서 빈발항목 집합을 탐사하는 방법으로는 FP-Tree 기반의 여러 가지 방법이 많이 연구 되었다. FP-Tree는 두 번의 데이터베이스 스캔만으

로 후보 항목을 생성하지 않고 빈발패턴을 탐사한다. 이 때문에 후보 항목이 생성 될 때마다 데이터베이스를 스캔하는 Apiori 기법에 비해 비용 소모가 적어 실시간 환경에 적합하다. 그러나 FP-Tree는 트랜잭션에 항목이 많아지면 트리의 복잡도가 급격히 증가하여 비용 소모가 커지는 한계를 지니고 있다[8].

이 절에서는 기존의 트리방식의 단점을 보완하기 위해 그래프를 이용한 FP-Digraph를 기술한다. FP-Digraph는 두 단계로 구성된다. 첫 번째 단계에서는 FIMWB 방법을 이용하여 최소 항목 가중치 값보다 크거나 같은 항목만을 대상으로 그래프를 구축한다. 두 번째 단계에서는 구축된 그래프를 통해 빈발 항목 집합을 탐사한다.

3.3.1 FP-Digraph 구축 단계

트랜잭션은 항목들이 알파벳순으로 되어있다고 가정한다. FP-Digraph는 노드와 방향이 있는 에지로 구성된다. 노드는 하나의 항목을 표현하고, 방향이 있는 에지는 트랜잭션을 표현하는 항목집합에서 인접한 두 항목들의 관계를 표현한다. 에지의 레이블은 두 속성을 표현한다. 첫 번째 속성은 트랜잭션의 항목집합에서 첫 항목부터 에지의 소스노드가 표현하는 항목을 포함하는 서브항목집합이다. 두 번째 속성은 서브항목집합의 빈발 횟수를 의미한다. 최초 에지 생성 시 1로 초기화되고 경로가 일치하는 트랜잭션이 존재할 때마다 1씩 증가한다.

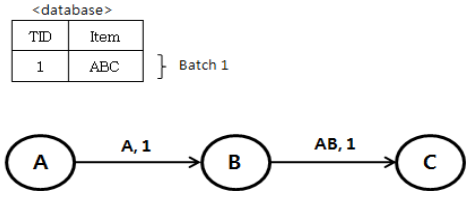


그림 2. Subgraph of FP-Digraph

항목 A, B, C를 포함하는 하나의 트랜잭션이 존재하는 데이터베이스를 FP-Digraph로 표현하면 [그림 2]와 같다. A, B, C 로 표현되는 세 개의 노드가 생성되고 각 노드사이에는 두 노드의 관계를 의미하는 에지가 생성된다. A와 B 노드 사이의 에지에 대한 값은 목적지 노드 B까지의 경로 A와 경로 빈발 횟수 1이 저장된다. B와 C 노드 사이의 에지는 에지의 목적지 노드 C까지의 경로 AB 와 경로 횟수 1 이 저장된다. FP-Digraph에서는 각 노드로 들어가는 에지의 정보를 통해 빈발항목 집합을 탐사한다.

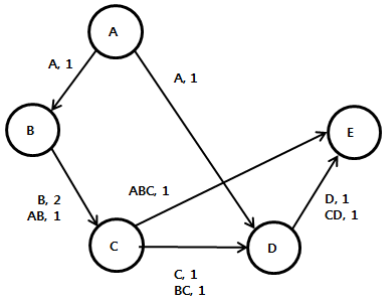


그림 3. 윈도우 1에 대한 FP-Digraph

3.3.3 빈발항목 집합 탐사 단계

구축된 FP-Digraph에서 빈발항목 집합을 탐사하는 것은 간단한 알고리즘을 통해 수행된다. [그림 3]에서는 다섯 개의 노드(A, B, C, D, E)가 존재한다. 각 노드는 들어오는 에지와 나가는 에지를 가지고 있다. 예를 들어 노드 D는 들어오는 에지의 레이블들 <A, 1>과 <C, 1>, <BC, 1>이 존재하며, 나가는 에지의 레이블은 <D, 1>과 <C D ,1>이 존재한다. 빈발항목 집합 탐사는 각 노드에서 들어오는 에지만을 대상으로 수행된다.

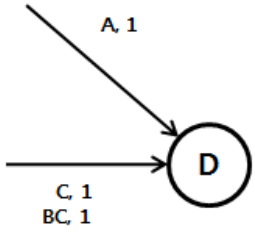


그림 4. 노드 D에 대한 빈발항목 집합 탐사

임의의 노드 D에 대한 빈발항목 집합 탐사는 다음과 같다. 먼저 D로 들어오는 각 에지의 레이블에 있는 정

보를 이용하여 서브항목집합과 그의 빈발 횟수를 조사하여 빈발 후보항목을 생성한다. 빈발 후보항목은 에지 레이블에서 서브항목 집합을 분할하여 얻어진다. [그림 4]에서는 D노드의 서브항목 집합 {<A: 1>, <C: 1>, <BC :1>} 이 존재한다. 이 서브항목 집합을 한 개의 항목으로 분할했을 때 <BC :1 >이 {<B: 1>, <C: 1>}로 분할된다. D 노드로 들어오는 에지에 대한 서브 항목집합 분할이 끝나면 같은 항목집합끼리 더하여 빈발 후보 항목집합 {<A: 1>, <C: 2>, <B: 1>}을 만든다. 최소 지지도 횟수를 2 라고 정의했을 때 생성된 후보 항목 중 A와 B는 가지치기 되고 C 는 빈발 항목이 된다. 빈발 항목에 대해서 노드 D와 접합 과정을 수행하여 빈발항목 집합(<CD :2>)을 탐사할 수 있다. D노드에 대한 탐사를 통해 항목 D를 접미부로 하는 모든 빈발 항목집합을 얻을 수 있다.

D에 대한 탐사가 끝나면 D노드와 D노드로 들어오는 에지는 더 이상 탐사 대상에서 제외된다. 이것은 노드 D에서 나가는 에지가 D에 대한 경로 정보를 가지고 있기 때문에 다른 노드의 탐사를 통해서 항목 D가 접미부가 아닌 빈발항목 집합의 탐사를 보장해준다. 예를 들어 E 노드의 탐사과정을 살펴보자. 노드 E 에서는 후보 항목집합{<A: 1>, <B: 1>, <C: 2>, <D: 2>} 가 생성이 되고 {<C: 2>, <D: 2>}가 빈발 후보 항목으로 탐사가 된다. 그리고 E 노드와의 접합을 통해 빈발항목 집합 {<CE: 2>, <DE: 2>, <CDE: 2>}을 탐사할 수 있다. 이처럼 D가 아닌 다른 노드의 탐사를 통해서 D가 접미부가 아닌 빈발 항목집합에 대해서 탐사를 할 수 있다. 이후 남아있는 다른 노드에 대해서 동일한 작업을 재귀적으로 수행하고 모든 노드에 대해서 한 번씩 탐사를 수행하면 탐사는 종료된다.

3.4 알고리즘

Input : 배치로 구성된 한 개의 윈도우
output : 빈발 항목으로 구성된 윈도우

```
FIMWB(window wi in database D)
//weight_batch = 배치에 부여된 가중치
//weighted_support= 최소 가중치 값
        For All batch bj in wi
        begin
          if(there exists an item in bj)
            add weight_batch to item's weight
          else
            item's weight set weight_batch
        end
        For(All Item in wi)
        begin
          if(item's weight < weighted support )
            remove item
        end
```

알고리즘 1. FIMWB 알고리즘

Input : 빈발 항목으로 구성된 윈도우
output : FP-Digraph

```
Make_Graph(FIMWB's output)
//path is initialized to first item in transaction
For From second item To last item in transaction
        begin
          add item to path
                If( exist path in graph)
                   add one to node_count
                else
                   add new path
                   new path's count set 1
        end
```

알고리즘 2. FP-Digraph 구축 알고리즘

Input : FP-Digraph
output : 빈발항목 집합

```
Mining_Graph(G)
```

```
//minsup_count = 최소 지지도 횟수
for exist All Node
   begin
      for All edge in Node ni
         begin
            for Path splitting in edge ej
               begin
                  if exist splitted node nk in path
                     add 1 to node_count
                  else
                     node_count set 1
               end
         end
   if( node_count > minsup_count)
      nk is frequent item with destination Node ni
   else
      remove nk
end
```

알고리즘 3. FP-Digraph 마이닝 알고리즘

4. 실험결과

실험 환경은 Intel Quad Core 2.40GHz, RAM 3GB, 이며 Java JDK1.6 언어로 작성되어 수행한다. 실험 데이터는 인공 데이터인 T10I4D100K dataset을 대상으로 한다. T10I4D100K dataset은 총 870개의 항목으로 구성되어있고 트랜잭션의 평균 크기는 10 이다. 실험을 위해 하나의 트랜잭션은 10개의 항목으로 구성된다. 또한 하나의 배치는 두 개의 트랜잭션으로 구성되고 하나의 윈도우는 세 개의 배치로 구성된다. 실험은 전체 데이터를 대상으로 하고 윈도우 단위로 수행된다.

T10I4D100K dataset를 대상으로 이동 윈도우 방식으로 FIMWB 방법을 적용한 결과는 [그림 5]와 같다. 최소 항목 가중치 값이 작으면 가지치기 비율이 낮고 클수록 가지치기 비율이 높아진다. 여기서는 최소 항목 가중치 값이 0.5 일 때 약 63% 정도의 항목이 가지치기 되는 것을 확인할 수 있다. 가지치기 되지 않은 항목은 FP-Digraph 방법을 이용하여 빈발항목 집합 탐사에 이용된다.

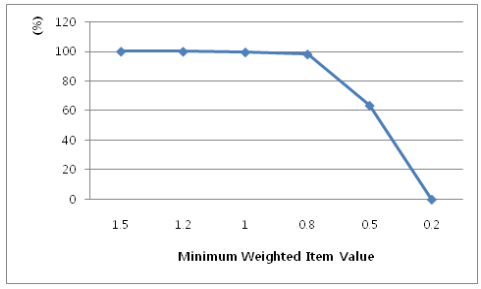


그림 5. FIMWB 적용 시 항목 개수의 감소 비율

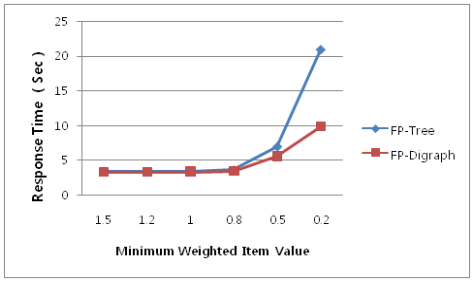


그림 6. 알고리즘의 수행 속도 비교

FIMWB 과정 후 FP-Tree와 FP-Digraph 로 수행했을 때의 결과는 [그림 6]과 같다. 최소 항목 가중치 값이 0.2 일 경우에는 약 54% 정도 연산속도가 감소하였고 최소 가중치 값이 0.5 일 경우에는 약 17% 정도 연산속도가 감소하였음을 확인할 수 있다. 전체적인 결과를 확인하면 빈발항목 집합 탐사의 대상이 되는 항목의 개수가 작을 때보다 많을수록 FP-Tree에 비해 더 빠른 처리 속도를 보이는 것을 알 수 있다.

5. 결론

스트림데이터는 무한하고 연속적인 특성을 지니고 있기 때문에 전체 데이터를 기반으로 빈발 항목집합을 탐사하는 것은 어렵다. 따라서 스트림데이터의 특성을 고려한 특수한 마이닝이 이루어져야 한다.

이 논문에서는 최신 데이터를 중요시 하는 사용자의 특성을 반영하여 빈발항목을 탐사하는 FIMWB 방법을 제안하였다. 제안한 방법은 최소 항목 가중치 값보다 작은 항목을 마이닝 수행 이전 시점에서 제거함으로써 마이닝 수행 시 불필요한 항목에 대한 연산을 줄일 수 있었다.

FP-Digraph는 FIMWB 방법을 통해 탐사된 빈발 항목만을 대상으로 빈발항목 집합을 탐사하였다. 이 방법은FP-Tree와 달리 항목에 대한 정렬 과정이 없어 더 빠르게 빈발항목 집합을 탐사할 수 있었다. 실험결과 기존의 FP-Tree 방법에 비해 항목의 개수가 많을수록 처리속도가 향상된 것을 확인할 수 있다.

참 고 문 헌

[1] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," pp.207-216, in Proc.ACM SIGMOD 1993.

[2] J. Chang and W. Lee, "A Sliding Window Method for Finding Recently Frequent Itemsets over Online Data Streams," Journal of Information Science and Engineering, Vol.20, No4, 2004(7).

[3] M. M. Gaber, "Mining data streams:a review," ACM SIGMOD record, Vol.34, No.2, pp.18-26, 2005.

[4] Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer, and B. S. Jeong, "Efficient Mining of Weighted Frequent Patterns Over Data Streams," 11th IEEE International Conference on High Performance Computing and Communications, 2009(6).

[5] C. F.ahmed, S. K. Tanbeer, and B. S. Jeong, "Efficient Mining of Weighted Frequent Patterns Over Data Streams," 2009 11th International Conference on High Performance Computing and Communications, pp.400-406, June, Seoul, Korea, 2009.

[6] Y. Kim, W. Kim, and U. Kim, "Mining Frequent Itemsets with Normalized Weight in Continuous Data Streams," Journal of Information Processing Systems, Vol.6, No.1, 2010(3).

[7] Carson Kai-Sang Leung, and Boyu Hao, "Mining of Frequent Itemsets from Streams of Uncertain Data," IEEE International Conference on Data Engineering, 2010(5).

[8] Vivek Tiwari, Vipin Tiwari, Shailendra Gupta, and Renu Tiwari, "Association Rule Mining: A Graph Based Approach for Mining Frequent Itemsets", IEEE International Conference on Networking and Information Technology, 2010(7).

[9] J. Pei, J. Han, B. M. Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, "Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach," IEEE Transactions on Knowledge and Data Engineering, Vol.16, No.11, 2004.

[10] G. Chen, X. Wu, and X. Zhu, "Mining Sequential Patterns Across Data Streams," Univ. of nd montComputerScience Technical Report(CS-05-04), 2005.

[11] LTC Bruce D.Caulkins, and J.Leem M.Wang, "A Dynamic Data Mining Technique for Intrusion Detection Systems," 43rd ACM Southeast Conference, March 18-20, 2005, Kennesaw, GA, USA.

저 자 소 개

서 복 일(Bok-Il Seo) 준회원

- 2010년 2월 : 전남대학교 전자컴퓨터공학과(공학사)
- 2010년 3월 ~ 현재 : 전남대학교 전자컴퓨터공학과 석사과정

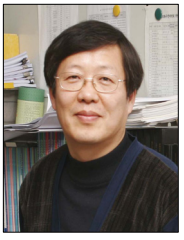
<관심분야> : 스트림 데이터 마이닝, 데이터베이스 시스템, 이동 컴퓨팅

김 재 인(Jae-In Kim) 정회원

- 2008년 2월 : 전남대학교 전자컴퓨터공학과(공학사)
- 2010년 2월 : 전남대학교 전자컴퓨터공학과(공학석사)
- 2010년 3월 ~ 현재 : 전남대학교 전자컴퓨터공학과 박사과정

<관심분야> : 스트림 데이터, U-Health, 스트림 데이터 마이닝

황 부 현(Bu-Hyun Hwang) 정회원



- 1978년 : 숭실대학교 전산학과(학사)
- 1980년 : 한국 과학기술원 전산학과(공학석사)
- 1994년 : 한국 과학기술원 전산학과(공학박사)
- 1980년 ~ 현재 : 전남대학교 전자컴퓨터공학부 교수

<관심분야> : 스트림 데이터 마이닝, 이동 컴퓨팅, 분산 시스템, 분산 데이터베이스, 전자 상거래