韓國IT서비스學會誌
第16卷 第1號
2017年 3月, pp.73-82

Journal of Information Technology Services
https://doi.org/10.9716/KITS.2017.16.1.073

다중 해시함수 기반 데이터 스트림에서의 아이템 의사 주기 탐사 기법*

이학주** · 김재완*** · 이원석****

Finding Pseudo Periods over Data Streams based on Multiple Hash Functions*

Hak-Joo Lee** · Jae-Wan Kim*** · Won-Suk Lee****

■ Abstract ■

Recently in-memory data stream processing has been actively applied to various subjects such as query processing, OLAP, data mining, i.e., frequent item sets, association rules, clustering. However, finding regular periodic patterns of events in an infinite data stream gets less attention. Most researches about finding periods use autocorrelation functions to find certain changes in periodic patterns, not period itself. And they usually find periodic patterns in time-series databases, not in data streams. Literally a period means the length or era of time that some phenomenon recur in a certain time interval. However in real applications a data set indeed evolves with tiny differences as time elapses. This kind of a period is called as a pseudo-period. This paper proposes a new scheme called FPMH (Finding Periods using Multiple Hash functions) algorithm to find such a set of pseudo-periods over a data stream based on multiple hash functions. According to the type of pseudo period, this paper categorizes FPMH into three, FPMH-E, FPMH-PC, FPMH-PP. To maximize the performance of the algorithm in the data stream environment and to keep most recent periodic patterns in memory, we applied decay mechanism to FPMH algorithms. FPMH algorithm minimizes the usage of memory as well as processing time with acceptable accuracy.

Keyword : Data Stream, Period, Multiple Hash Function

Submitted : September 12, 2016 1st Revision : January 26, 2017 Accepted : February 6, 2017

* 이 논문은 2015년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.R0190-15-2019, 빅데이터 환경에서 비식별화 기법을 이용한 개인정보보호 기술 개발).

** 연세대학교 일반대학원 컴퓨터과학과 석사

*** 연세대학교 일반대학원 컴퓨터과학과 석사과정

**** 연세대학교 컴퓨터과학과 교수, 교신저자

1. 서 론

데이터 스트림이란 센서, 로그, 주식, 날씨, 의료 데이터와 같이 데이터가 아주 빠르고 지속적으로 무한하게 발생하는 환경을 의미한다. 이러한 데이터 스트림 환경에서는 데이터베이스 환경과는 다르게 데이터를 여러 번 읽어 처리하는 것이 아니라, 메모리 에서 단 한 번만 읽고(1-scan) 처리를 해야 한다. 데이터를 여러 번 읽게 되면 처리 시간이 길어져 실시간으로 정보를 제공하지 못하기 때문이다. 또한, 하드디스크와 달리 물리적인 메모리 공간에 제약이 있기 때문에 한정된 메모리 공간에서 데이터를 처리해야 한다는 제약조건이 있다(Cho, 2012).

데이터가 주기를 갖는다는 것은 같은 현상이나 특성이 한 번 나타나고부터 다음 번 되풀이되기까지의 시간이 일정하다는 것을 의미한다. 데이터의 주기를 파악할 수 있다면 미래를 예측할 수 있기 때문에 의료 분야, 날씨 예보, 주식, 보안 등 여러 분야에서 필수적인 부분이다. 만약 데이터의 주기가 정확하게 나타나는 상황이라면, 주기를 찾아 활용하는 것은 크게 어려운 작업이 아닐 것이다. 주기가 정확하게 나타난다면 다음 주기도 정확하게 예측할 수 있기 때문이다. 하지만 실제 세계에서는 주기는 정확하게 나타나지 않고, 오차를 포함하게 된다(Guan et al., 2015). 이러한 주기를 의사 주기(pseudo period)라고 한다(Tang et al., 2007). 오차를 포함한다는 의미는 데이터의 주기가 정확하지 않고, 약간의 변화를 가지며 발생한다는 의미다. 예를 들면 주기가 3초인 데이터는 다음 번 데이터가 나오기 까지 정확히 3초가 지나야 발생하는 것이 아니라, 3.1초, 혹은 2.9초 등 약간의 오차를 수반하여 발생한다는 의미이다.

주기를 찾는 모델은 크게 세 가지로 분류할 수 있다. 첫째, 횟수와 주기가 오차를 갖지 않고 정확할 때 주기를 찾는 모델. 둘째, 횟수에 대해 오차를 허용하는 횟수 의사 주기 모델, 마지막으로 주기에 대해 오차를 허용하는 주기 의사 주기 모델이 있다. 기존의 방법들은 의사 주기 패턴보다는 그 변화에 초점을 맞추었거나(Tang et al., 2007), 데이터 스트림 환경에 적합하지 않는 방법이었다. 또한, 새로 주기 패턴이 생기거나, 사라지는 것들은 포착하지 못하는 단점이 있다.

본 논문에서는 위의 세 가지 주기 패턴 모델을 마이닝하기 위한 알고리즘을 제시한다. 세부 알고리즘은 각각의 장점을 갖고 있으며, 상황에 맞게 더 적합한 알고리즘을 적용시킬 수 있다. 본 논문에서 제시하는 FPHM 기법은 다중 해시함수를 사용하여 주기를 찾는다. 본 기법에서 구축하는 해시 테이블은 물리적으로 고정된 크기의 공간이고, 해시 함수의 특성상 레코드에 대한 접근 속도가 동일하기 때문에 공간 복잡도 및 시간 복잡도 측면에서 효율적이다.

주기가 새로 생기거나 사리지는 등 현재 주기에 초점을 맞추기 위해 각각의 기법에 디케이 메커니즘(decay mechanism) (Chang and Lee, 2006)을 적용시켰다. 이를 이용해 최근에 검색된 주기를 강조함으로써, 과거에는 주기가 있었으나 현재는 주기가 없어진 아이템이나, 과거에는 주기가 있었으나 현재는 새로 주기가 생긴 아이템을 찾아낼 수 있다.

2. 관련 연구

데이터 마이닝이란 방대한 데이터 가운데 숨겨진 패턴을 찾거나 미래에 어떤 데이터가 생성될 것인지 예측하여 의사 결정에 이용하는 기술이다(Ding et al., 2015). 예를 들면, 미국 월마트에서는 남성이 주말에 기저귀를 사면 맥주도 살 확률이 높다는 연관관계를 연관 규칙(association rules)이라는 데이터 마이닝 기법을 이용해 밝혀냄으로써 주말과 평일에 상품진열을 다르게 하여 수익을 창출해 냈다.

데이터 스트림 마이닝이란 무한히 발생되는 데이터 스트림으로부터 빈발 항목 집합(frequent itemsets) 패턴, 클러스터링(clustering), 연관 규칙(association rules) 등을 찾아내는 작업을 의미한다. 데이터 마이닝은 과거부터 특정 시점까지의 데이터를

분석하여 패턴을 찾아내는데 초점을 맞추는데 반해, 데이터 스트림 마이닝은 과거부터 지속적으로 찾은 패턴의 변화를 다룬다. 즉, 데이터 마이닝은 시작 시점과 종료 시점이 있는 반면 데이터 스트림 마이닝은 종료 시점이 무한히 확장된다. 그러므로 패턴을 찾기 위해서 전통적인 데이터 마이닝 기법을 사용하려면 전체 데이터가 누적되어 있어야 하지만, 데이터 스트림 마이닝은 규칙 집합(rule set)만 누적하여 관리함으로써 종료시점이 정해지지 않은 방대한 양의 데이터를 처리할 수 있다는 장점이 있다.

빈발항목집합(Frequent itemsets) 연구는 다양한 방법을 사용하여, 한정된 메모리와 시간 안에서 각 아이템의 빈발 정도를 찾아내는 연구이다. 그 중 다중해시함수를 사용하여 각 아이템의 빈발 정도를 찾아내는 방법을 제시한 연구가 있었다(Jin et al., 2003). 알고리즘 동작 방식은 다음과 같다. n개의 각각 다른 홈 주소(home address)를 갖는 해시 함수를 생성한다. 아이템이 발생하면 각 해시 함수에 해당하는 홈 주소에 카운트를 누적한다. 아이템의 개수가 늘어나면 늘어날수록 해시 함수의 홈 주소가 겹쳐 카운트가 잘못 누적되는 현상이 많이 발생하게 되지만, 해시 함수가 여러 개이기 때문에 한 아이템에 대해 각 해시함수의 카운트 값 중 가장 작은 값은 실제 카운트와 거의 차이가 없게 됨이 실험에서 증명되었다(Jin et al., 2003). 그러나 한정된 메모리 공간을 고려하여, 주어진 데이터에 맞춘 해시 함수와 그에 대한 홈 주소 조절이 필요하다. 본 논문에서는 다중해시함수 테이블을 새롭게 변형시켜 주기를 찾는 알고리즘에 적용하였다.

주기 패턴을 찾기 위한 알고리즘은 기존에 여러 가지가 존재했다. 순환 자기 상관 방법을 사용하여 정확한 주기를 검색하는 방법(Parthasarathy et al., 2006) 및 시계열 데이터베이스에서 정확한 주기 패턴을 검색하는 방법(Elfeky et al., 2005)이 있었다. 또한 데이터스트림에서 확률분포 모델을 이용해 정확한 주기성을 관리하는 모델도 존재한다(Tao and Ozsu, 2009).

디케이란 현재 들어온 정보에 더 초점을 맞추기 위해 가중치(weight)를 반영하는 방법이다(Chang and Lee, 2006). 디케이 방법을 적용시키면 자연스레 과거의 중요도가 떨어지는 정보들이 사라지고, 현재 정보들에 더욱 초점이 맞춰지게 되어 랜드마크(Landmark) 모델 방식의 단점을 극복할 수 있다. 디케이 비율(decay rate) d는 두 개의 독립 변수로 구성되어 있으며 b는 디케이-베이스(decay-base), h는 디케이-베이스-라이프(decay-base-life)이다.

$$d = b^{-1/h}, (b > 1, h \geq 1).$$

디케이-베이스는 가중치 감소량을 결정하는 변수이다. 디케이-베이스가 커진다면 감소되는 양도 많아지게 된다. 디케이-베이스-라이프는 가장 최근 발생한 아이템의 가중치가 1이라 가정할 때, 동값이 1/b가 될 때까지 위 연산의 누적적용 횟수에 관계하는 변수이다.

3. 다중 해시함수 기반 주기 탐색 기법

본 논문에서는 다중 해시함수 기반 주기 탐색(Finding Period using Multiple Hash functions : FPMH) 기법을 제시한다. 이는 검색 대상 주기 패턴에 따라 정확한 주기 탐색(FPMH Exact Period : FPMH-E), 횟수 의사 주기 탐색(FPHM Pseudo Count Period : FPHM-PC), 주기 의사 주기 탐색(FPMH Period Pseudo Period : FPMH-PP)으로 분류한다. 각 기법의 특성은 <Table 1>과 같다.

〈Table 1〉 FPMH Algorithms

Algorithm	Finding Period	Finding Count sudo period	Finding sudo period	decay
FPMH-E(decay)	○	×	×	×(○)
FPMH-PC(decay)	○	○	×	×(○)
FPMH-PP(decay)	○	×	○	×(○)

1) 스트림(Stream) : S = {s_0, …, s_n, …}, |S|=∞. 0~n은 각 아이템이 발생한 시점
2) 아이템(Item) : s ∈ S, 각 아이템은 발생시점이 존재
3) 발생시간차(Time interval) : s_a, s_b∈S일 때 발생시간차는 b-a(b > a)
4) 시간차 관리 해시 테이블(Time Interval Management Hash Table : TIMHT) : 아이템 발생 횟수와 아이템의 타임스템프를 이용해 주기 패턴이 될 가능성이 있는 아이템을 검색 및 저장
5) 주기 관리 테이블(Period Management Table : PMT) : TIMHT로부터 주기 패턴으로 인정된 아이템을 수신하여 관리
6) 주기 신뢰값(Period confidence count) : TIMHT가 PMT로 아이템, 발생시간차를 전송할 때 PMT의 조건에 부합하면 카운트를 증가
7) 유일항목집합(ΣS) : 스트림 S에서 서로 다른(distinct) 아이템의 집합. 예를 들어 S = {a, b, c, a, b, c}이면 ΣS = {a, b, c}
8) 전체 아이템 발생횟수(Item total count) : ΣS = {a}일 때 S에서 a가 발생한 횟수
9) 주기 신뢰율(Period confidence ratio) : 주기 패턴이 얼마나 정확한 가에 대한 척도. 즉, 아이템의 주기가 얼마나 정확하게 시간을 갖고 발생하였는지에 대한 비율

$$period\ confidence\ ratio = \frac{period\ confidence\ count}{item\ \ total\ \ count}$$

10) 유일 아이템 비율(Identical count ratio) :

$$Identical\ count\ \ ratio(\%) = \frac{|\Sigma s|}{|S|} \times 100$$

11) 해시 밀도(Hash density) :

$$Hash\ \ Density = \frac{item\ \ total\ \ count}{number\ \ of\ \ home\ \ addresses}$$

12) 해시충첩밀도(Packing density) :

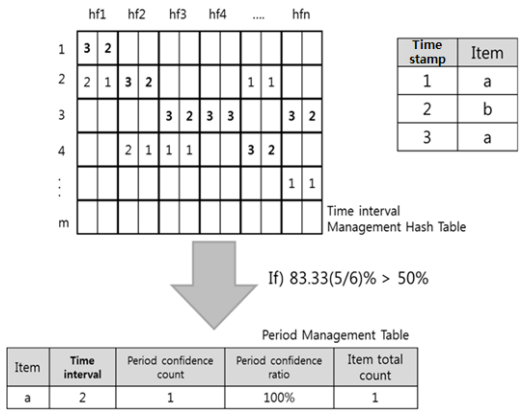
$$Packing\ \ Density = \frac{item\ \ total\ \ count}{number\ \ of\ \ hash\ \ functions}$$

13) 의사주기비율(Pseudo period coverage) :

$$Pseudo\ Period\ Coverage = \frac{\#\ of\ pseudo\ period\ patterns}{\#\ of\ period\ patterns}$$

FPMH는 크게 카운트와 타임스탬프(time stamp)를 이용하여 주기가 될 가능성이 있는 아이템을 찾는 TIMHT와 주기를 관리하는 PMT로 구분할 수 있다. TIMHT는 최근의 타임스탬프와 카운트를 저장하고 있으며, 아이템의 각 해시 함수의 카운트의 비율이 일정 임계치 이상이면 주기를 관리하는 PMT로 전송한다.

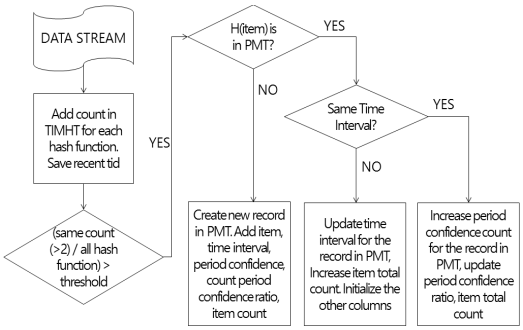
PMT는 TIMHT에서 아이템과 각 아이템이 얼마만큼 시간 간격으로 발생되었는지 알 수 있는 발생시간차 전송 받는다. 전송 받은 아이템, 발생시간차를 저장하고, 주기 신뢰값(Period confidence count)을 증가시키고, 주기 신뢰율(Period confidence ratio)을 계산한다. 주기는 발생시간차/전체 아이템 발생횟수 인데 본 알고리즘에서는 전체 아이템 발생횟수를 1로 고정하므로 발생시간차가 곧 주기이다.



〈Figure 1〉 FPMH Modules

FPMH-E는 FPMH 분류 중 가장 간단하고 기본적인 기능을 갖는다. FPMH-E의 목적은 정확한 주기 패턴을 찾는데 있으며 횟수, 주기가 모두 정확해야 주기 패턴으로 인정된다. 따라서 TIMHT가 PMT로 아이템, 발생시간차를 전송할 때 아이템과 발생시간차가 모두 동일해야만 주기 신뢰값이 증가

한다. 또한 횟수 의사 주기를 인정하지 않기 때문에 주기 신뢰율은 100%가 유지된다.



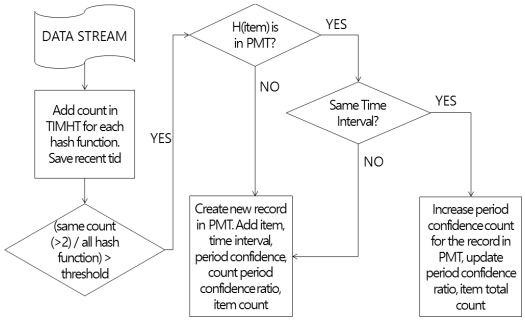
〈Figure 2〉 FPMH-E Work Flow

FPMH-PC는 횟수에 대한 의사 주기 패턴을 인정한다. 따라서 같은 아이템이라고 하더라도 여러 개의 주기 패턴을 갖는 것이 허용된다. TIMHT의 레코드 수보다 훨씬 더 많은 고유한 아이템이 존재하게 된다면 TIMHT의 오류 발생률은 증가한다.

FPMH-E에서는 한 개의 아이템이 여러 개의 주기 패턴을 갖는 것이 불가능 했으나 FPMH-PC에서는 여러 개의 주기 패턴을 갖는 것이 허용된다. 따라서 FPMH-E 알고리즘보다 유연하다. FPMH-E는 정확도가 100% 혹은 0%인 반면에 FPMH-PC는 그 사이의 값들을 가질 수 있다. 즉, FPMH-E는 99%라는 높은 수치의 정확도를 가진 패턴을 무시하지만 횟수 의사 주기를 인정하게 되면 99%라는 높은 수치의 정확도를 가진 주기 패턴을 마이닝 할 수 있게 된다.

<Figure 3>은 FPMH-PC의 작업 흐름도이다. FPMH-E와의 차이점은 발생시간차가 다를 때 기존 레코드의 발생시간차를 갱신하는 대신, 새로운 레코드를 생성하고 그 레코드에 현재 값들을 할당하는데 있다.

FPMH-PP는 사용자로부터 주기 패턴으로 인정여부에 대한 임계치를 받는다. 예를 들어 PMT에서 아이템 a의 발생시간차가 2이고 한계치가 10%일 때 발생시간차가 2.1인 아이템 a가 들어오면 허용 편차는 2×0.1 = 0.2이다. 그러므로 1.8 이상



〈Figure 3〉 FPMH-PC Work Flow

2.2 이하의 발생시간차는 2와 같은 발생시간차로 취급한다. 전술한 FPMH 알고리즘들과 다른 점은 발생시간차가 같은지 검사할 때 임계치를 사용한다는 점이다.

디케이는 주기 관리 테이블의 주기 신뢰값을 이용해 구현한다. 주기 관리 테이블에 존재하는 아이템과 상이한 아이템이 발생한 경우 기존 아이템의 주기 신뢰값에 디케이를 적용한다. 예를 들어 주기 관리 테이블에 아이템 a가 존재하고 이 아이템의 주기 신뢰값이 1인 상황에서 아이템 b가 삽입되었다면 아이템 b의 디케이 비율은 1이다. 디케이-베이스가 1.1, 디케이-베이스-라이프가 1이라 가정할 때 디케이 비율은 0.91로 계산되므로, 아이템 a의 기존 주기 신뢰값에 0.91을 곱하여 이 아이템의 갱신된 주기 신뢰값은 0.91이 된다. 주기 신뢰값이 임계치 이하가 되면 주기 관리 테이블에서 해당 아이템을 삭제하고, 과거 주기 분석을 위해 필요할 경우 별도 저장장치에 옮겨 보관한다.

4. 성능 평가

실험은 조건부 생성 데이터와 실제 데이터를 사용하여 실험하였다. 실험 환경은 리눅스 커널 2.6.35에서 수행되었으며, 4GB의 메모리를 가진 듀얼코어 2.4GHz CPU로 구성하였다.

조건부 생성 데이터는 주어진 유일항목집합의 각 아이템에 대해 각각 주기 폭(P)을 할당하고, 주기 폭의 오차를 설정하여 생성하였다. 또한 아이템이

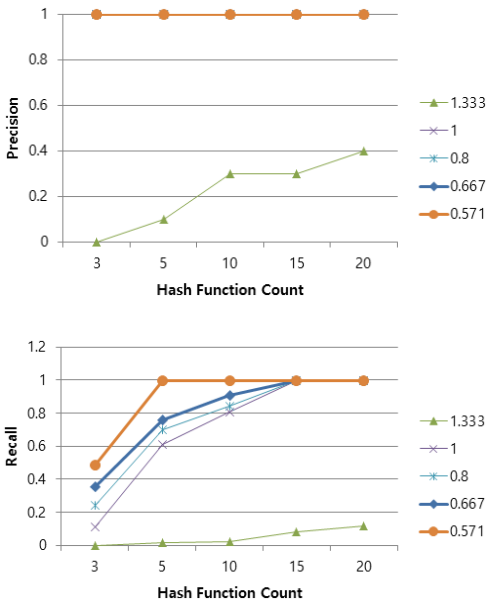
발생해야 할 시점에 미리 설정된 확률에 따라, 해당 아이템이 발생하지 않도록 하였다.

이번 실험은 FPMH-E의 정확도에 대한 실험이다. 정확도는 precision과 recall로 측정한다.

$$pescision = \frac{actual\ periods\ found\ by\ algorithm}{total\ periods\ found\ by\ algorithm}$$

$$recall = \frac{actual\ periods\ found\ by\ algorithm}{actual\ periods}$$

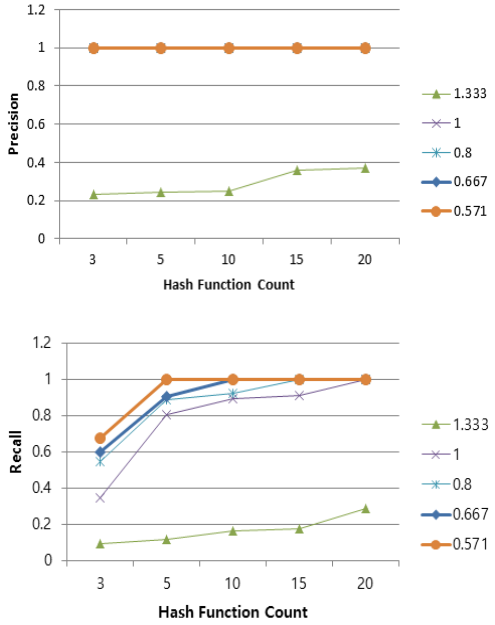
<Figure 4>~<Figure 7>에서 범례는 해시 밀도를 의미한다. 입력속도는 400tuple/ms이고, 유일 아이템 비율은 65%이다.



〈Figure 4〉 Accuracy According to Change of Hash Functions Hash Density for FPMH-E

<Figure 4>를 살펴보면 해시 함수의 개수가 늘어나면 TIMHT의 해시중첩밀도가 줄어들어 precision, recall의 값이 높이지게 된다. 해시 밀도도 해시 함수와 같이 해시중첩밀도가 줄어들기 때문에 같은 효과를 가진다. 하지만 해시 함수나 해시 밀도 중 하나만 늘어나거나 줄어든다고 해서 정확도가 높아지는 것은 아니다. 해시 함수의 개수가 늘어나면 TIMHT의 특성상 겹쳐 횟수가 잘못 증가되는 것을 방지할 수 있고, 해시 밀도가 줄어들면 해시 함수의 충돌을 피할 수 있는 확률이 커지게 된다.

다음 실험은 FPMH-PC에 대한 정확도를 측정한다. 파라미터 세팅은 입력속도 400tuple/ms, 유일 아이템 비율 65%, 최소지지도 0.01로 하였다.

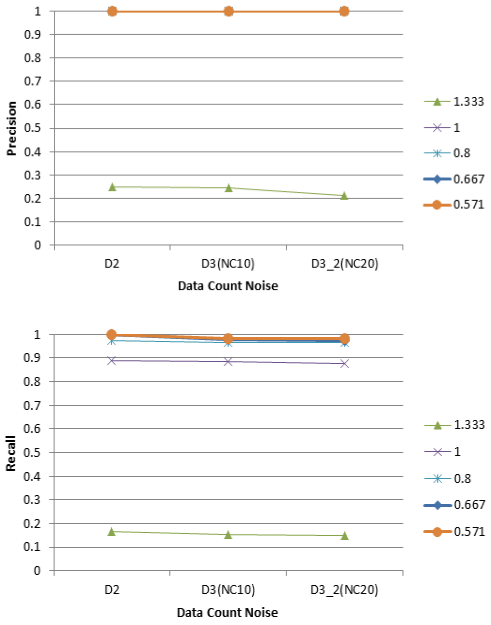


〈Figure 5〉 Accuracy According to Change of Hash Functions Hash Density for FPMH-PC

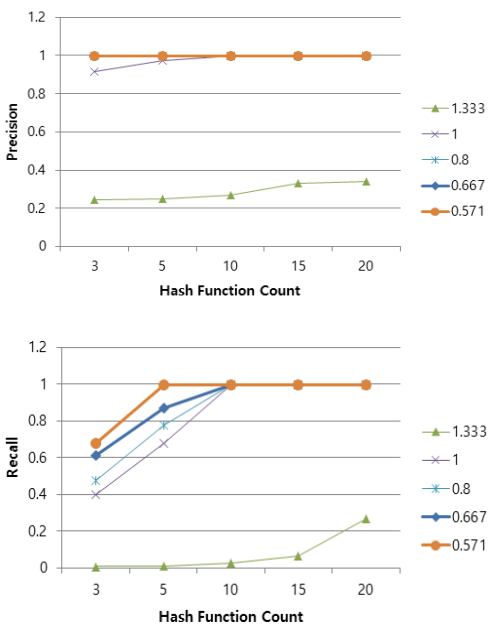
FPMH-E는 정확도가 0% 혹은 100%이기 때문에 한 아이템이 10번 중 9번이 주기를 갖는다고 하여도 정확도가 0%이다. <Figure 5>는 횟수 의사 주기를 인정함에 따라 FPMH-E보다 더 높은 정확도를 가지는 것을 볼 수 있다.

다음 실험은 해시 함수 개수를 고정시키고 데이터 카운트 오차와 TIMHT의 해시중첩밀도를 변화시키며 측정한 실험이다. 해시 함수는 10개로 고정하였다.

FPMH-PP 역시 앞의 실험과 비슷한 형식으로 실험한다. 파라미터는 입력속도 400tuple/ms, 유일 아이템 비율 65%, 최소지지도 0.01, 의사주기 비율 15%이다.

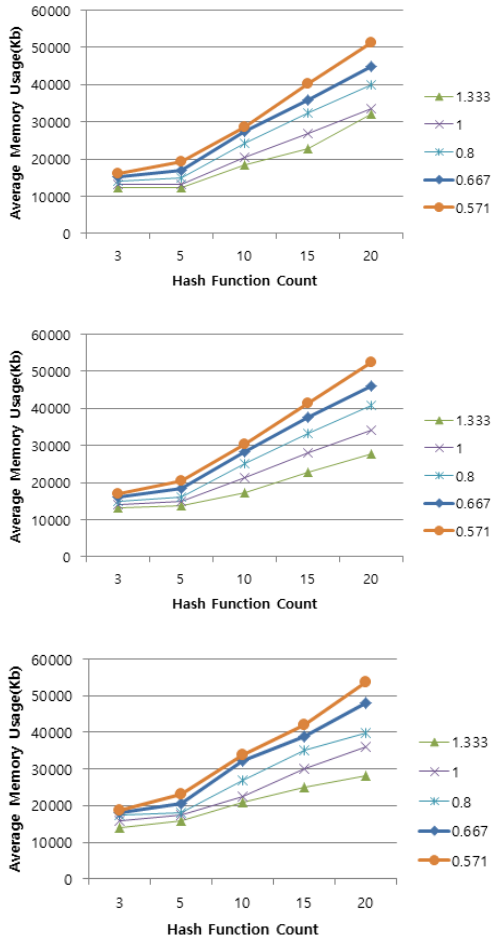


〈Figure 6〉 Average Accuracy According to Data Count Noise



〈Figure 7〉 Accuracy According to Change of Hash Functions Hash Density for FPMH-PP

다음은 메모리 사용량과 아이템 당 평균 처리속도에 관한 실험이다.

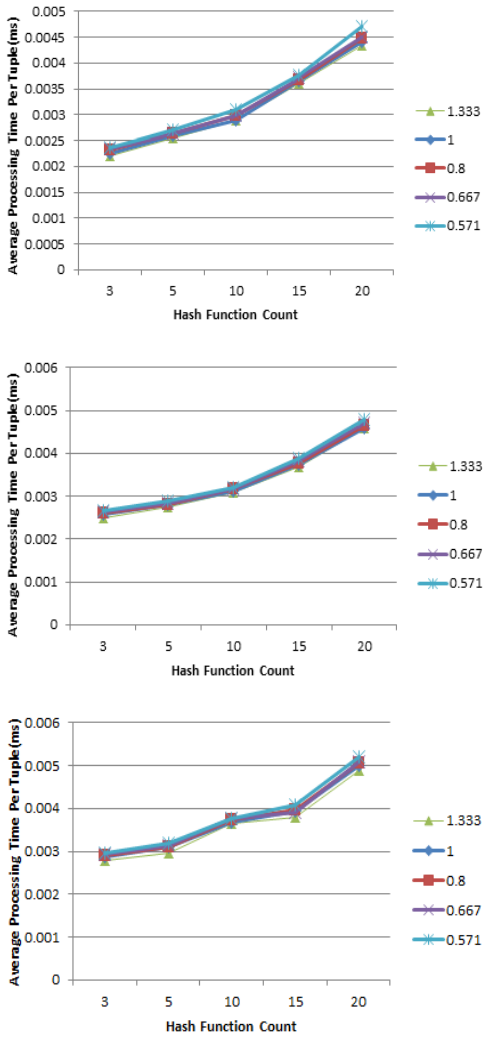


〈Figure 8〉 Memory usage of FPMH-E, PC, PP

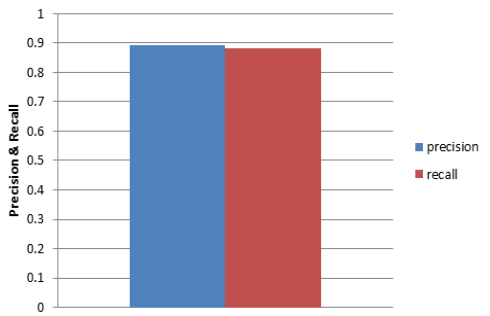
시간차 관리 해시 테이블의 메모리 크기는 고정값이기 때문에 메모리 사용량은 PMT의 크기에 비례한다.

평균 처리시간은 해시함수의 개수에 비례한다. 이는 TIMHT에서 PMT로 전송시 연산량이 증가하기 때문이다. 또한 FPMH-E에 비해 FPMH-PC, FPMH-PP의 평균 처리시간이 약간 증가하는 것은 의사주기처리를 위한 연산량 증가에 기인한다.

마지막 실험은 실제(real) 데이터를 사용하여 측정하였다. 데이터는 태양의 흑점개수 주기에 대한 데이터이다. 1900년부터 2008년까지 태양의 흑점 개수이다. 태양의 흑점 개수 범위는 0개부터 200개 사이이다. 개수별로 이산화 했을 때 실제로 발생한



〈Figure 9〉 Average Processing time for FPMH-E, PC, PP



〈Figure 10〉 Average Accuracy for Periodic Patterns of Sunspot Count

흑점 개수는 157개이다. 일, 월 단위로는 특별한 주기 패턴이 나타나지 않지만, 약 10년 단위로 의사 주기 패턴이 나타나는 특징이 있다. 파라미터는 입력 속도 400tuple/ms, 유일 아이템 비율 65%, 최소지지도 0.01, 의사주기비율 30%이다. 또한 해시함수 10개, TIMHT 해시중첩밀도는 0.8로 고정하였다.

평균 정확도를 측정한 결과 precision, recall 모두 약 90%의 정확도를 갖는다. 이는 157개의 아이템 중 주기를 갖는 15개의 아이템을 추적하여 측정한 결과이다.

5. 결 론

본 논문에서는 다중해시함수 기반인 FPMH 기법을 제시하였다. 또한 의사 주기를 횟수 의사 주기와 주기 의사 주기로 분류하여 이에 따른 발전된 기법을 제시하였다.

TIMHT의 해시중첩밀도를 적절히 조절한다면, FPMH는 매우 작은 고정 메모리 공간을 사용하여 거의 정답과 유사한 정확도로 주기를 검색한다. 해시 함수를 사용하기 때문에 응답속도 또한 보장된다. 이러한 FPMH를 목적에 맞게 FPMH-E, FPMH-PC, FPMH-PP의 세부 기법으로 확장하고, 디케이를 적용하여 과거 주기와 현재 주기의 중요도를 다르게 구분할 수 있도록 개선하였다.

본 논문에서 제시한 기법은 조건부 생성 데이터와 실제 데이터를 사용하여 성능 평가를 하였다. 응답 속도, 메모리 사용량, 정확도 모두 효율적인 성능을 보이는 것을 확인하였다. 하지만 의사 주기를 좀 더 정밀하게 관리하는 방법을 찾을 수 있도록 보완되어야 할 것이다.

References

Chang, J.H. and W.S. Lee, “Finding Recently Frequent Itemsets Adaptively over Online Transactional Data Streams”, *Information Systems*, Vol.31, No.8, 2006, 849-869.

Cho, O.W., "Continuous Computing based OLAP Analysis for Multi-dimension Data Streams", Yonsei Grad. Univ., Dept. of CS, 2012.

(조오욱, "다차원 데이터 스트림을 위한 연속 처리 기반의 OLAP 분석 기법", 연세대학교 대학원 : 컴퓨터과학과, 2012.)

Ding, S., F. Wu, J. Qian, H. Jia, and F. Jin, "Research on Data Stream Clustering Algorithms", *Artificial Intelligence Review,* Vol. 43, No.4, 2015, 593-600.

Elfeky, M.G., W.G. Areg, and A.K. Elmagarmid, "Periodicity Detection in Time Series Databases", *IEEE Transactions on Knowledge and Data Engineering*, Vol.17, No.7, 2005, 875-887.

Guan, T., K.R. Wang, and S.P. Zhang, "A Robust Periodicity Mining Method from Incomplete and Noisy Observations based on Relative Entropy", *International Journal of Machine Learning and Cybernetics*, Vol.8, Issue.1, 2015, 283-293.

Jin, C., W. Qian, C. Sha, J.X. Yu, and A. Zhou, "Dynamically Maintaining Frequent Items over A Data Stream", *CIKM '03 Proceedings of the Twelfth International Conference on Information and Knowledge Management*, 2003, 287-294.

Parthasarathy, S., S. Mehta, and S. Sriviasan, "Robust Periodicity Detection Algorithms", *CIKM '06 Proceedings of the 15th ACM International Conference on Information and knowledge Management*, 2006, 874-875.

Tang, L., B. Cui, H. Li, G. Miao, D. Yang, and X. Zhou, "Effective Variation Management for Pseudo Periodical Streams", *Sigmod '07 Proceedings of the 2007 ACM Sigmod International Conference on Management of Data*, 2007, 257-268.

Tao, Y. and M.T. Ozsu, "Mining Data Streams with Periodically Changing Distributions", *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, 2009, 887-896.

About the Authors

No Image

Hak-joo Lee (oasisys@nate.com)

Hak-joo Lee received the M.S. degree in the Department of Computer science at Yonsei Univ., S. Korea in 2012. His Research interests are Big data, Data mining, Data stream processing

Jae-wan Kim (kukuri98@gmail.com)

Jae-wan Kim received the B.S. degree in the Department of Computer science at Yonsei Univ., S. Korea in 2005. His Research interests are Big data, Data mining on Hadoop-echosystem

Won-Suk Lee (leewo@database.yonsei.ac.kr)

Won-Suk Lee received the B.S. degree in Computer Engineering from Boston University, Boston and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from Purdue University, West Lafayette, IN. He is currently a professor of Department of Computer science at Yonsei Univ., Korea. Also, he works as a representative in Software&Contents Laboratory.