ISSN 1598-0170 (Print)
ISSN 2287-1136 (Online)
http://www.jksii.or.kr

랜드마크 윈도우 기반의 빈발 패턴 마이닝 기법의 분석 및 성능평가☆

Analysis and Evaluation of Frequent Pattern Mining Technique based on Landmark Window

편 광 범[1] 윤 은 일[1*]
Gwangbum Pyun Unil Yun

요 약

본 논문에서는 랜드마크 윈도우 기반의 빈발 패턴 마이닝 기법을 분석하고 성능을 평가한다. 본 논문에서는 Lossy counting 알고리즘과 hMiner 알고리즘에 대한 분석을 진행한다. 최신의 랜드마크 알고리즘인 hMiner는 트랜잭션이 발생할 때 마다 빈발 패턴을 마이닝 하는 방법이다. 그래서 hMiner와 같은 랜드마크 기반의 빈발 패턴 마이닝을 온라인 마이닝이라고 한다. 본 논문에서는 랜드마크 윈도우 마이닝의 초기 알고리즘인 Lossy counting와 최신 알고리즘인 hMiner의 성능을 평가하고 분석한다. 우리는 성능평가의 척도로 마이닝 시간과 트랜잭션 당 평균 처리 시간을 평가한다. 그리고 우리는 저장 구조의 효율성을 평가하기 위하여 최대 메모리 사용량을 평가한다. 마지막으로 우리는 알고리즘이 안정적으로 마이닝이 가능한지 평가하기 위해 데이터베이스의 아이템 수를 변화시키면서 평가하는 확장성 평가를 수행한다. 두 알고리즘의 평가 결과로, 랜드마크 윈도우 기반의 빈발 패턴 마이닝은 실시간 시스템에 적합한 마이닝 방식을 가지고 있지만 메모리를 많이 사용했다.

☞ 주제어 : 랜드마크 윈도우, 빈발 패턴 마이닝, 온라인 마이닝, 성능평가, 확장성

ABSTRACT

With the development of online service, recent forms of databases have been changed from static database structures to dynamic stream database structures. Previous data mining techniques have been used as tools of decision making such as establishment of marketing strategies and DNA analyses. However, the capability to analyze real-time data more quickly is necessary in the recent interesting areas such as sensor network, robotics, and artificial intelligence. Landmark window-based frequent pattern mining, one of the stream mining approaches, performs mining operations with respect to parts of databases or each transaction of them, instead of all the data. In this paper, we analyze and evaluate the techniques of the well-known landmark window-based frequent pattern mining algorithms, called Lossy counting and hMiner. When Lossy counting mines frequent patterns from a set of new transactions, it performs union operations between the previous and current mining results. hMiner, which is a state-of-the-art algorithm based on the landmark window model, conducts mining operations whenever a new transaction occurs. Since hMiner extracts frequent patterns as soon as a new transaction is entered, we can obtain the latest mining results reflecting real-time information. For this reason, such algorithms are also called online mining approaches. We evaluate and compare the performance of the primitive algorithm, Lossy counting and the latest one, hMiner. As the criteria of our performance analysis, we first consider algorithms' total runtime and average processing time per transaction. In addition, to compare the efficiency of storage structures between them, their maximum memory usage is also evaluated. Lastly, we show how stably the two algorithms conduct their mining works with respect to the databases that feature gradually increasing items. With respect to the evaluation results of mining time and transaction processing, hMiner has higher speed than that of Lossy counting. Since hMiner stores candidate frequent patterns in a hash method, it can directly access candidate frequent patterns. Meanwhile, Lossy counting stores them in a lattice manner; thus, it has to search for multiple nodes in order to access the candidate frequent patterns. On the other hand, hMiner shows worse performance than that of Lossy counting in terms of maximum memory usage. hMiner should have all of the information for candidate frequent patterns to store them to hash's buckets, while Lossy counting stores them, reducing their information by using the lattice method. Since the storage of Lossy counting can share items concurrently included in multiple patterns, its memory usage is more efficient than that of hMiner. However, hMiner presents better efficiency than that of Lossy counting with respect to scalability evaluation due to the following reasons. If the number of items is

[1] Dept. of Computer Engineering, Sejong University, Seoul, 143-747, Korea
* Corresponding author (yunei@sejong.ac.kr)
[Received 05 February 2014, Reviewed 18 February 2014, Accepted 14 April 2014]

☆ 본 논문은 2013년도 정부 교육과학기술부의 재원으로 한국 연구재단(NRF)의 지원을 받아 수행된 연구사업임(No.2013005682 and 20080062611).
☆ 본 논문은 2014년도 인터넷정보학회 춘계학술발표대회 우수논문 추천에 따라 확장 및 수정된 논문임.

increased, shared items are decreased in contrast; thereby, Lossy counting's memory efficiency is weakened. Furthermore, if the number of transactions becomes higher, its pruning effect becomes worse. From the experimental results, we can determine that the landmark window-based frequent pattern mining algorithms are suitable for real-time systems although they require a significant amount of memory. Hence, we need to improve their data structures more efficiently in order to utilize them additionally in resource-constrained environments such as WSN(Wireless sensor network).

☞ keyword : Landmark Window, Frequent pattern mining, Online mining, Performance evaluation, Scalability

1. 서 론

최근 데이터베이스는 온라인 서비스 기술이 증가하면서 스트림을 지원하는 형태로 바뀌어 가고 있다. 데이터베이스에서 유용한 정보 또는 지식을 얻는 기술인 데이터마이닝도 스트림 연구가 활발하게 진행되고 있다. 스트림 기반의 데이터베이스를 데이터 스트림이라고 하며 데이터 스트림 기반의 데이터마이닝 기술은 계속 변동되는 데이터베이스에 대한 마이닝 기술이다. 데이터 스트림 기반의 빈발 패턴 마이닝의 특징은 계속 트랜잭션이 발생하는 데이터베이스를 마이닝 하기 위하여 데이터베이스를 한 번만 스캔하여 빈발 패턴을 마이닝 하는 것이다. 초기의 스트림기반의 빈발패턴 마이닝은 데이터베이스 전체를 한번 스캔하여 트리를 구성하고 마이닝의 효율성을 위해 정렬된 트리로 재구축 하는 방식이었고 비효율적 이다. 그래서 빠르게 마이닝을 해야 하는 스트림 환경에 적합한 빈발 패턴 마이닝을 위해 윈도우 기술이 개발되었다. 첫 번째 윈도우 기술은 슬라이딩 윈도우 기술로 일정 수의 최신 트랜잭션을 유지하면서 유지하고 있는 트랜잭션을 이용하여 마이닝 하는 기술이다. 두 번째 윈도우 기술은 템프드 윈도우 기술로 시간이 지날수록 트랜잭션의 가치가 감소하는 형태로 마이닝 하는 기술이다. 마지막으로 세 번째 기술은 본 논문에서 분석하고 평가하는 랜드마크 윈도우 기술이다. 본 논문에서는 랜드마크 윈도우의 초기 기술인 Lossy counting[2]기술과 최신 랜드마크 윈도우 기술인 hMiner[3]의 성능을 비교하고 랜드마크 윈도우 기술에 대하여 분석한다.

2. 랜드마크 윈도우 기반의 빈발 패턴 마이닝 분석

2.1 관련 연구

스트림 데이터베이스를 위한 빈발 패턴 마이닝은 데이터베이스를 한 번만 스캔해야하는 특수성이 있다. 빈발 패턴마이닝의 초기 알고리즘인 apriori[1] 알고리즘은 마이닝으로 얻고자 하는 패턴의 최대 길이가 N 일 때, 데이터베이스를 N 번 스캔해야한다. 그래서 스트림 환경의 빈발 패턴 마이닝을 위해 데이터베이스를 한 번만 스캔하고 마이닝을 수행하는 P-tree[5] 가 제안되었다. P-tree는 단순하게 데이터베이스를 한 번만 스캔하는 방법만 제안했기 때문에 다양한 스트림 환경에 적합하지 않았다. 윈도우 기술은 랜드마크 윈도우[1], 슬라이딩 윈도우 [4], 템프드 윈도우 [6]가 제안되었다. 슬라이딩 윈도우는 데이터베이스에서 최신 구간을 지정하고 최신 구간을 지속적으로 유지하는 방법이다. 템프드 윈도우는 트랜잭션에 가중치를 부여하여 최신 트랜잭션에 많은 가중치를 주고 오래된 트랜잭션에 작은 가중치를 준다. 마지막으로 랜드마크 윈도우는 스트림 데이터에서 일부 구간 또는 트랜잭션 단위로 마이닝을 수행하고 최종적으로 전체적인 마이닝 결과를 얻을 수 있는 방법이다. 랜드마크 윈도우는 시간이 지남에 따라 추가된 구간만을 마이닝 하기 때문에 스트림환경에 적합한 방식이다. 랜드마크 윈도우의 마이닝 기술들은 구간 또는 실시간으로 마이닝을 지속적으로 수행하기 때문에 언제든지 필요할 때 마이닝 결과를 바로 얻을 수 있다. 그래서 랜드마크 윈도우 기술이 발전함에 따라 최근에는 온라인 마이닝이라고 한다. 랜드마크 윈도우 방식의 마이닝은 Closed 패턴 마이닝 기술과 접목한 FP-CDS [7] INSTANT+ [9]는 Maximal 패턴 마이닝 기술과 접목한 것이다. 또한 Top-k 마이닝 기술과도 접목한 방법 [10]도 있다. 랜드마크 윈도우 기반의 빈발 패턴 마이닝 기술을 응용한 방법으로 랜드마크 윈도우 최신 기술인 hMiner를 분산 시스템에 적용한 알고리즘 [13]이 제안되었다. 그리고 Molecular Dynamic Simulation 의 온라인 분석 [8] 과 같이 의학 데이터에도 접목할 수 있으며 데이터 요약 [14] 에도 응용될 수 있다.

2.1 Lossy counting 알고리즘

랜드마크 윈도우 기술은 Lossy counting 알고리즘부터 시작되었다. Lossy counting 알고리즘은 에러 파라미터에

따라 일정한 크기의 batch를 정의하고 batch 크기만큼의 트랜잭션이 발생하면 batch내의 트랜잭션에서 발생된 패턴을 검사하고 이전 batch의 빈발 패턴들과 합 연산한다. batch의 크기 w 는 ϵ를 에러 파라미터라고 정의하였을 때 $w=\frac{1}{\epsilon}$ 로 정의된다. 예를 들어 ϵ가 0.001이면 batch의 크기는 1000이다. Lossy counting는 batch의 트랜잭션을 읽어 발생할 수 있는 모든 패턴을 만들고 빈도수를 계산한다. 이때, 패턴을 빈발 패턴 저장장소에 저장한다. 패턴이 e이고 빈도수가 f, bcurrent가 Δ일 때 (e,f,Δ)형태로 저장장소에 저장한다. 그리고 bcu1rrent 값보다 큰 빈도수를 가진 패턴만 빈발 패턴 저장 장소에 저장한다. bcurrent는 트랜잭션 전체의 수가 N일 때, $b_{current}=\lceil\frac{N}{w}\rceil$ 로 계산된다. batch의 패턴을 저장장소에 저장하는 방법은 batch 안의 트랜잭션으로부터 모든 패턴을 조합한다. 조합된 패턴을 하나씩 꺼내어 저장장소의 빈발 패턴과 비교한다. 저장장소는 트리 형태의 구조를 가지고 있으며 각 노드에는 아이템 이름을 저장한다. 예를 들어 {a, b, c}의 패턴을 저장구조에 저장하기 위해선 {a, b, c}를 아이템의 이름 순서로 정렬한다. 그 다음 트리의 루트노드에 접근하여 자식노드 중에서 패턴의 첫 아이템이 있는지 확인한다. {a, b, c}의 첫 아이템은 'a' 아이템 이므로 루트노드의 자식노드 중에서 'a' 아이템을 가진 노드를 찾는다. 그리고 해당 노드로 이동한다. 현재 방문한 노드의 자식노드 중에서 다음 아이템인 'b' 아이템을 찾고 이동한다. 마지막으로 현재 노드에서 'c' 아이템을 가진 자식노드로 이동하면 현재 노드에서 루트까지 경로의 아이템 집합은 {a, b, c}가 된다. Lossy counting는 이러한 방식으로 빈발 패턴을 저장한다. batch의 모든 트랜잭션 중 빈발 패턴을 빈발 패턴 저장장소에 저장하면 빈발 패턴 저장 장소에서 f +Δ < bcurrent 인 패턴을 제거한다. 그러면 빈발 패턴 저장 장소에는 빈발한 패턴만 남는다.

2.2 다양한 랜드마크 윈도우 기반 알고리즘

Lossy counting 알고리즘 이후 다양한 알고리즘들이 제안되었다. in-core [11]는 Apriori를 기반으로 제안되었다. in-core는 패턴에 포함된 아이템의 수가 증가할수록 마이닝 시간이 크게 증가하는 특성을 가진다. 이후 EStream [12]이 제안되었다. EStream은 batch 단위 대신 트랜잭션단위로 마이닝 결과를 도출하는 온라인 마이닝이 가능한 첫 알고리즘이다. EStream은 첫 온라인 마이닝이지만 단순한 빈발 패턴 저장장소와 약한 프루닝 기술로 인해 마이닝 속도와 메모리 사용량의 효율성을 보장하지 못한다.

2.3 hMiner 알고리즘

다음 알고리즘은 hMiner 알고리즘이다. hMiner는 최신의 랜드마크 알고리즘으로 하나의 트랜잭션이 발생할 때마다 빈발 패턴 저장장소에 저장한다. hMiner의 빈발 패턴 저장 장소는 hSynopsis라고 한다. hSynopsis는 해시 테이블과 frequent node로 이루어져 있다. 해시 테이블은 빈발패턴을 바로 접근하기 위한 해시 값을 저장하고 있으며 해시의 크기 H는 신뢰도 p, 에러 파라미터 , 데이터베이스의 아이템 수 M, 평균 트랜잭션 길이 L이 있을 때 다음과 같은 수식에 의해 결정된다.

$$H=e/\epsilon^2 M\times(e^L-1)\times ln((1-2^M)/ln\ p)$$

해시 테이블에 연결된 frequent node는 빈발 패턴을 저장하고 있다. 트랜잭션이 발생하면 트랜잭션에서 패턴을 조합하고 해당하는 패턴이 저장된 hSynopsis를 갱신한다. hSynopsis를 갱신하는 방법은 먼저 조합된 패턴의 해시 값을 계산한다. 해시 값에 따라 hSynopsis의 해시 버킷에 접근한다. 해시 버킷에는 특정 해시 값을 가진 빈발 패턴들이 저장되어 있다. hMiner는 버킷에 저장된 패턴들과 비교하여 조합된 패턴과 같은 정보를 담은 frequent node의 정보를 갱신한다. 그리고 hSynopsis에 저장된 빈발 패턴 중에 Minimum support보다 작은 패턴들은 프루닝 된다. 그러면 hSynopsis에는 빈발 패턴만 남는다.

3. 성능평가

성능평가는 랜드마크 윈도우의 두 알고리즘인 Lossy counting 와 hMiner를 비교하고 평가하였다. 평가 척도는 마이닝 시간, 메모리 사용량, 확장성 평가이다. 성능평가에 사용된 데이터는 IBM에서 제공하는 가상 데이터[15]를 이용하였다.

표 1 은 성능평가에 사용된 데이터 셋의 정보를 보여준다. T4I20D500K와 T5I20D50K는 마이닝 시간과 메모리 사용량 평가에 사용되고 T4I10D500K, T4I20D500K, T4I30D500K, T4I40D500K는 확장성 평가에 사용되었다. Lossy counting의 오류 파라미터는 Minimum support의 1% 이다. hMiner의 신뢰도 p는 0.9 이며 오류 파라메터는 0.0009이다.

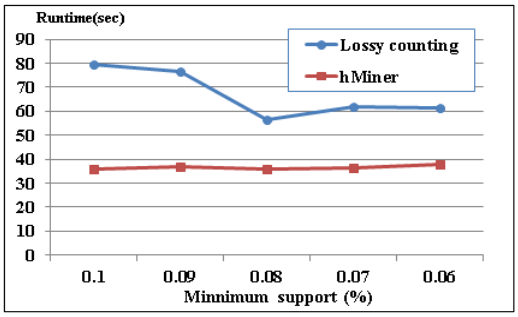
(표 1) 성능평가 데이터셋
(Table 1) data set of Performance evaluation

데이터 셋	평균 트랜잭션	아이템 수	트랜잭션 수
T4I20D500K	4	20,000	500,000
T5I20D50K	5	20,000	500,000
T4I10D500K	4	10000	500,000
T4I30D500K	4	30000	500,000
T4I40D500K	4	40000	500,000

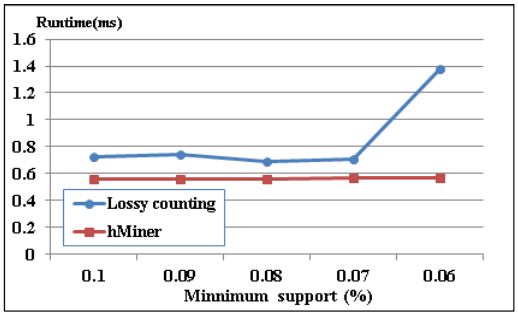
3.1 마이닝 시간

마이닝 시간에 대한 성능평가는 2가지에 대하여 평가한다. 첫 번째 평가는 성능평가 방법은 50만개의 트랜잭션을 처리하는 동안 소요된 마이닝 시간을 측정하였다. 두 번째 평가는 50만개의 트랜잭션을 처리하는 동안 트랜잭션 평균 처리 시간을 측정하였다. 두 평가 모두 Minimum support는 0.1%에서 0.06%까지 평가하였다.

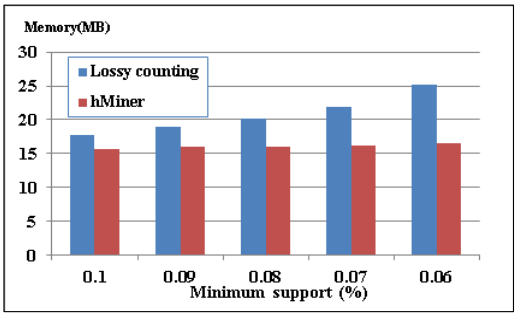
그림 1은 두 랜드마크 알고리즘의 마이닝 수행시간을 평가한 그래프이다. Lossy 알고리즘은 에러 파라미터에 따라 batch의 크기가 결정되므로 Minimum support가 낮을수록 batch의 크기가 커진다. 그러므로 Minimum support가 0.08%가 될 때 까지 처리해야할 batch의 수가 감소하여 마이닝 시간이 감소되는 추세이다. 그러나 batch의 크기가 커지기 때문에 트랜잭션이 batch 만큼 채워지지 않으면 최신 트랜잭션이 반영된 마이닝 결과를 얻을 수 없다. Minimum support가 0.07%부터 0.06% 구간은 처리해야할 패턴의 수가 증가하여 마이닝 시간이 0.08% 일 때 보다 증가하였다. hMiner는 hMiner는 트랜잭션 하나가 발생될 때마다 처리하기 때문에 Lossy counting보다 온라인 처리에 유리한 장점이 있다. 그림 2는 평균 트랜잭션의 길이를 5로 증가시키고 트랜잭션 처리시간의 평균을 평가한 것이다. 평가 결과 hMiner가 Lossy counting 보다 더 빠른 트랜잭션 처리 시간을 가진다. 그러므로 hMiner는 메모리의 제약이 없다면 Lossy counting 보다 효율적으로 마이닝이 가능하다.



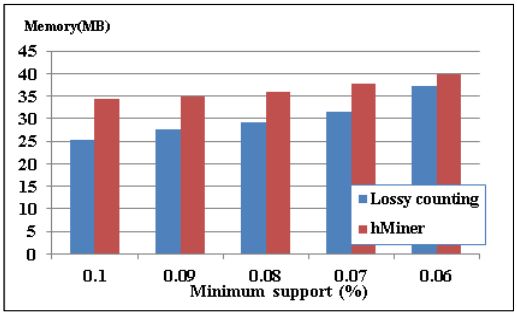
(그림 1) 마이닝 수행시간 평가
(Figure 1) Runtime test



(그림 2) 평균 트랜잭션 처리 시간 평가
(Figure 2) Average Processing time of transaction test



(그림 3) 최대 메모리 사용량 평가(T4I20D500K)
(Figure 3) Memory test(T4I20D500K)



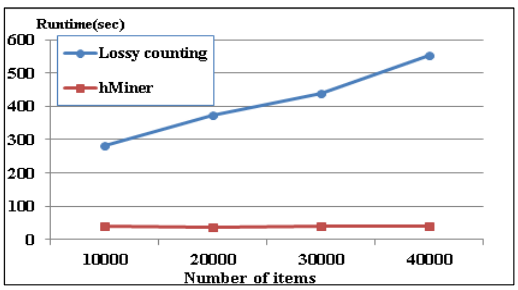
(그림 4) 최대 메모리 사용량 평가(T5I20D500K)
(Figure 4) Memory test(T5I20D500K)

3.2 최대 메모리 사용량

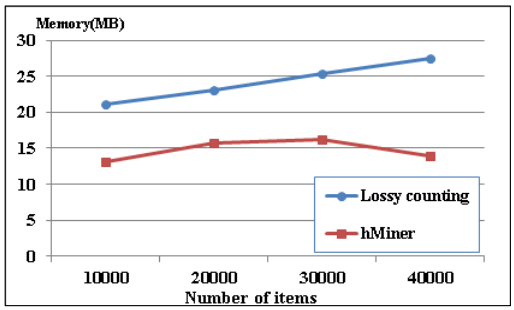
다음 평가는 두 알고리즘에 대한 메모리 사용량을 평가하였다. 최대 메모리 사용량은 마이닝 과정 중 가장 많이 사용한 메모리 수치이다. 그림 3은 T4I20D500K 데이터를 이용해 최대 메모리 사용량을 평가하였다. 그림 4는 T5I20D500K를 이용해 최대 메모리 사용량을 평가하였다. 그림 4는 두 랜드마크 알고리즘에 대한 최대 메모리 사용량을 평가한 그래프이다. 평가 결과 Minimum support가 감소할수록 처리해야할 패턴이 많아져 두 알고리즘 모두 메모리 사용량이 증가하였다. 그림 5는 트랜잭션의 평균 길이를 5로 증가시키고 마이닝 했을 때 최대 메모리 사용량을 보여준다. 그림 5에 의하면 Lossy counting 알고리즘이 hMiner 알고리즘보다 좋은 메모리 효율성을 보여준다.

3.3 확장성 평가

다음은 확장성 평가이다. 확장성 평가는 아이템의 수가 증가해도 마이닝 성능이 안정적으로 유지되는지 평가한다. 평가에 사용되는 데이터는 T4I10D500K, T4I20D500K, T4I30D500K, T4I40D500K 데이터로 아이템의 수가 10000개에서 40000개까지 증가한다. 첫 번째 평가는 마이닝 시간 평가로 평가 방법은 각 데이터를 모두 처리할 때까지의 마이닝 시간을 측정한다. 그림 5는 아이템의 수를 10000개에서 40000개까지 증가시켰을 때 50만개의 트랜잭션을 처리한 시간을 측정한 결과다. 평가 결과 아이템의 수가 증가할수록 hMiner가 효율적인 마이닝 시간을 보였다. 그림 6은 아이템의 수를 10000개에서 40000개까지 증가시켰을 때 최대 메모리 사용량을 보여준다. 평가 결과 아이템의 수가 증가할수록 처리해야 할 패턴의 수가 증가하고 메모리 사용량이 증가하게 된다. 메모리 사용량에 대해서도 hMiner가 Lossy counting보다 효율적인 결과를 보여주었다.



(그림 5) 확장성 평가(마이닝 시간)
(Figure 5) Scalability test(Runtime)



(그림 6) 확장성 평가(최대 메모리 사용량)
(Figure 6) Scalability test(Memory usage)

3.4 성능 분석

본 논문에서 성능을 평가한 두 알고리즘에 대한 마이닝 수행시간, 평균 트랜잭션 처리 시간, 최대 메모리 사용량, 확장성 평가 대한 전체 결과를 비교한 내용은 표 2와 같다.

(표 2) 두 알고리즘의 성능비교
(Table 2) Performance comparison of two algorithms

평가 방법	Lossy counting	hMiner
마이닝 수행시간	37.61초	61.2초
트랜잭션 처리시간	1.37밀리초	0.56밀리초
메모리 사용량 (T4I20D500K)	25.1MB	16.46MB
메모리 사용량 (T5I20D500K)	37.21MB	39.74MB
확장성 평가 (마이닝수행시간)	554.26초	40.42초
확장성 평가 (메모리사용량)	27.5MB	13.98MB

hMiner는 Lossy counting과 비교하면 빠른 마이닝 속도와 효율적인 확장성을 보여주지만 데이터의 평균 트랜잭션의 길이가 커질수록 메모리 효율성이 감소하는 특징을 보였다. hMiner는 해시의 버킷에 후보 빈발 패턴을 저장하기 때문에 패턴 정보를 모두 저장한다. 반면에 Lossy counting은 lattice 방식을 사용하여 후보 빈발 패턴을 저장한다. 후보 빈발 패턴을 저장하기 위한 Lossy counting의 저장 구조 는 여러 패턴들에 동시에 포함된

아이템들을 공유하여 저장하기 때문에 메모리 사용량은 Lossy counting이 hMiner보다 효율적이다. 확장성 평가는 hMiner가 Lossy counting보다 높은 효율성을 보여준다. 그 이유는 아이템의 수가 증가하면 공유되는 아이템의 수가 감소되어 Lossy counting의 메모리 효율성이 약해지고, 트랜잭션이 증가하면 Lossy counting의 프루닝 효과가 감소하기 때문이다.

4. 결 론

두 알고리즘을 평가한 결과 랜드마크 윈도우 기반의 빈발 패턴 마이닝 알고리즘은 트랜잭션이 지속적으로 발생하는 스트림 환경에서 효율적으로 마이닝 하는 알고리즘이다. 이 알고리즘들은 일정 수의 트랜잭션 또는 하나의 트랜잭션이 발생할 때마다 처리하기 때문에 중간 중간 마이닝 결과가 필요한 실시간 시스템에 적합하다. 랜드마크 윈도우 기반의 초기 알고리즘인 Lossy counting 알고리즘은 일정 크기의 트랜잭션이 발생 하였을 때 처리하기 때문에 어느 정도 기간 또는 시간적 여유가 있는 시스템에 적용될 수 있으며 hMiner는 트랜잭션마다 처리하기 때문에 실시간 시스템에 적합하다. hMiner 알고리즘은 신뢰도와 에러 파라미터를 통해 메모리 제약을 줄 수 있다. 그러나 WSN(Wireless Sensor Network)와 같은 휴대용 단말기 또는 소형 센서에 적용하기 위해선 작은 메모리를 사용하는 구조가 필ㄹ요하다. hMiner는 트랜잭션 평균 길이가 증가하면 메모리 사용량이 증가하는 단점을 가지고 있다. 그래서 hMiner와 같은 최신 랜드마크 윈도우 기반의 빈발 패턴 마이닝이 제한적인 환경에 적용되기 위해선 더욱 효율적인 구조를 개발해야 하며 앞으로 더 많은 연구가 필요하다.

참 고 문 헌(Reference)

[1] R. Agrawal and R. Srikant, Fast algorithms for Mining Association Rules, in Proc. of the 20th int'l Conf. on Very Large Data Bases(VLDB), pp.487-499, 1994.

[2] G. S. Manku, R. Motwani, Approximate Frequency Counts over Data Streams, International conference on Very Large Data Bases, pp. 346-357, 2006.

[3] E. T. Wang, A. P. Chen, A novel hash-based approach for mining frequent itemsets over data streams requiring less memory space, Data Mining and Knowledge Discovery, vol.19, no.1, pp.346-357, 2006.

[4] S.K. Tanbeer, C.F. Ahmed, B.S. Jeong and Y.K. Lee, Sliding window-based frequent pattern mining over data streams, Information sciences, vol.179, no.22, pp.3843-3865, 2009.

[5] H. Huang, X. Wu, R. Relue, Mining Frequent Patterns with the Pattern Tree, New Generation Computing, vol.23, pp.315-337, 2004.

[6] J. H. Chang and W. S. Lee, Finding recently frequent itemsets adaptively over online transactional data strems, Information Systems, vol.31, pp.849-869, 2006.

[7] X. Liu, J. Guan and P. Hu, Mining frequent closed itemsets from a landmark window over online data streams, Computers & Mathematics with Applications, vol.57, no.6, pp.927-936, 2009.

[8] A. Ramanathan, P. K. Agarwal, M. kurnikova and C. J. Langmead, An Online Approach for Mining Collective Behaviors form Molecular Dynamics Simulations, International Conference on Research in Computational Molecular Biology, pp.138-154, 2009.

[9] H. Li, N. Zhaing, Z. Chen, A Simple but Effective Maximal Frequent Itemset Mining Algorithm over Streams", Journal of Software, vol. 7, no. 1, pp. 25-32 Jan. 2012

[10] R.C. Wong, A.W. Fu, "Mining Top-k frequent itemsets from data streams", Data Mining and Knowledge Discovery(DMKD), vol.13, no.2, pp. 193-217, 2006.

[11] R. Jin, G. Agrawal, An Algorithm for In-Core Frequent Itemset Mining on Streaming Data, International Conference on Data Mining(ICDM), pp.210-217, 2005.

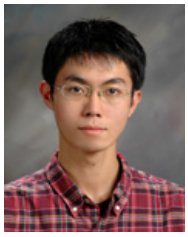
[12] X. H. Dang, W. Ng, K. Ong, Online mining of frequent sets in data streams with error guarantee, Knowledge and Information Systems, vol.16, no.2, pp.245-258, 2008.

[13] E. T. Wang, A. L. Chen, Mining frequent itemsets over distributed data streams by continuously maintaining a global synopsis, Data Mining and Knowledge Discovery, vol.23, no.2, pp.252-299, 2011.

[14] X. Zhu, W. Ding, P. S. Yu, C. Zhang, One-class learning and concept summarization for data streams, Knowledge and Information Systems, vol.28, no.3, pp.523-553, 2011.

[15] Frequent itemset Mining dataset repository. (www.almaden.ibm.com/software/projects/hdb/resources.shtml)

◐ 저 자 소 개 ◑



편 광 범
2010년 충북대학교 컴퓨터공학전공 학사. (공학사)
2012년 충북대학교 대학원 컴퓨터과학 석사. (공학석사)
2012년 ~ 현재 세종대학교 대학원 컴퓨터공학 박사과정
관심분야: 데이터마이닝, 정보검색, 데이터베이스
Email: pyungb@sju.ac.kr

윤 은 일
1997년 고려대학교 이학석사. (이학석사)
1997 ~ 2006 한국통신 멀티미디어연구소 전임/선임연구원.
2005년 Texas A&M Univ. 공학박사 (공학박사)
2005 ~ 2006 Texas A&M Univ. 포스닥연구원.
2006 ~ 2007 한국전자통신연구원, 선임연구원.
2007 ~ 2012 충북대학교 전자정보대학 컴퓨터공학부 조교수
2012 ~ 2013 충북대학교 전자정보대학 소프트웨어학과 부교수
2013 ~ 현재 세종대학교 전자정보공학대학 컴퓨터공학과 부교수
관심분야: 데이터마이닝, 정보검색, 데이터베이스
Email: yunei@sejong.ac.kr