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

확률 기법에 기반한 근접 빈발 패턴 마이닝 기법의 성능평가☆

Performance evaluation of approximate frequent pattern mining based on probabilistic technique

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

요 약

근접 빈발 패턴 마이닝은 향상된 효율성을 위해 정확한 패턴보다 허용되는 범위 안에서 근접 빈발 패턴을 마이닝한다. 데이터베이스의 크기가 증대함에 따라 거대한 데이터베이스를 처리하기 위해서 더 빠른 마이닝 기법이 필요하게 되고 있다. 또한, 노이지나 데이터의 다양성 때문에 패턴을 마이닝 하는 것에 대한 정확한 결과를 찾기가 더 어렵다. 이러한 경우들에 대해, 근접 빈발 패턴 마이닝을 함으로 실행시간, 메모리 사용량, 그리고 확장성의 관점에서 더 효율적인 마이닝을 수행할 수 있다. 이 논문에서는 확률 기법에 근간한 근접 패턴 마이닝 알고리즘에 대한 특성을 살펴보고 척도가 되는 확률 기법에 기반한 근접 패턴 마이닝 알고리즘에 대해 성능 평가를 한다. 최종적으로 성능의 향상을 위해 테스트 결과를 분석한다.

☞ 주제어 : 근접 빈발패턴마이닝, 체르노프 기법, 확률기법, 성능평가, 확장성

ABSTRACT

Approximate Frequent pattern mining is to find approximate patterns, not exact frequent patterns with tolerable variations for more efficiency. As the size of database increases, much faster mining techniques are needed to deal with huge databases. Moreover, it is more difficult to discover exact results of mining patterns due to inherent noise or data diversity. In these cases, by mining approximate frequent patterns, more efficient mining can be performed in terms of runtime, memory usage and scalability. In this paper, we study the characteristics of an approximate mining algorithm based on probabilistic technique and run performance evaluation of the efficient approximate frequent pattern mining algorithm. Finally, we analyze the test results for more improvement.

☞ keyword : approximate frequent pattern mining, Chernoff technique, probabilistic technique, Performance evaluation, scalability

1. 서 론

데이터마이닝 기술 중 하나인 근접 빈발 패턴 마이닝 [11], [13]은 데이터베이스에 노이지가 있거나 빠르게 결과를 얻기 위해 실제 빈발 패턴에 근접한 패턴을 마이닝 하는 알고리즘이다. 최근 서비스해야 정보가 기하급수적으로 증가하면서 정보 제공을 위한 데이터베이스의 크기도 빠르게 증가하고 있는 추세이다 [5]. 근접 빈발 패턴마이닝은 네트워크 분야에서 활용되고 있으며 [9] 또한 바이오 분석 쪽에도 사용되고 있다 [8]. 이러한 다양한 분야에 사용하기 위해선 실시간으로 데이터를 분석하고 할 수 있어야 한다. 기존의 유명한 빈발 패턴 마이닝 알고리즘으로 Aprioiri [1]와 FP-growth [4] 이 있다. 두 알고리즘은 자주 발생하는 패턴을 효율적으로 마이닝 하는 알고리즘이지만 매우 큰 데이터베이스나 매우 복잡한 패턴을 가지는 데이터에 대하여 빈발 패턴을 마이닝 할 때 시간이 오래 걸리며 메모리 사용량이 많아지게 된다. 이로 인해 많은 컴퓨터 자원이 필요하게 되고 빠르게 분석 결과를 서비스하기 어려운 상황에 도달하게 된다. 실제로 위 두 가지 알고리즘은 정확한 결과를 도출하기 위해 데이터베이스의 모든 데이터를 사용하기 때문에 이러한 문제가 발생하게 된다. 데이터에 노이지가 있는 경우 마이닝 기술을 통해 빈발 패턴을 마이닝 한다 해도 그 결과가 노이지에 의해 정확하지 않은 결과를 도출하게 되고 보통 이런 경우에는 정확하지 않더라도 실제 빈발 패턴에 근

1 Dept. of Computer Science and Research Institute for Computer and Information Communication, Chungbuk National University, Chungcheongbuk-do Cheongju-si, 361-763, Korea
* Corresponding author (yunei@chungbuk.ac.kr)
[Received 31 October 2012, Reviewed 18 September 2012(R2 26 December), Accepted 14 January 2013]
☆ 이 논문은 2012년도 정부 교육과학기술부의 재원으로 한국연구재단의 지원을 받아 수행된 연구사업임(No. 2012-0003740 and 2012-0000478)
☆ 본 논문은 2012년도 한국인터넷정보학회 하계학술발표대회 우수논문의 확장버전임.

접한 패턴만으로도 의사결정을 하거나 분석 서비스를 할 수 도 있다. 데이터의 크기가 매우 큰 경우에도 빠르게 분석을 해야 하는 상태서는 대략적인 마이닝 결과라도 빠른 분석을 위해 근접 빈발 패턴이 필요 할 수 있다. 근접 빈발 패턴 마이닝 [9] 은 완벽하게 정확하지 않더라도 빈발 패턴을 빠르게 분석해야하는 상황에서 필요한 기술이다. 근접 빈발 패턴 마이닝은 이러한 필요성 때문에 다양한 기법들이 연구되었다. 최근에 연구된 근접 빈발 패턴 마이닝은 빠른 마이닝과 적은 메모리 사용량을 보이면서도 정확도가 실제 빈발 패턴에 매우 근접한 결과를 내고 있다. 근접 빈발 패턴 마이닝의 최신 동향으로 SWCA [6]는 최신 트랜잭션을 유지하면서 마이닝 하는 슬라이딩 윈도우기반의 근접 빈발 패턴 마이닝이다. 스트림 환경에서는 어느 순간이라도 마이닝이 가능해야 하며, 노이지가 발생할 가능성이 높다. 그래서 근접 빈발 패턴마이닝은 스트림 형태의 데이터베이스에 효율성을 보인다. PFIM [2]은 아이템이 불확실한 상태에서 근접 빈발 패턴 마이닝 하는 방법이며 이 방법은 근접한 결과를 도출하는 근접 빈발 패턴 마이닝과 높은 연관성을 가진다. D-FIMA [7]는 근접 빈발 패턴 마이닝을 응용한 방법으로 무선 센서 네트워크에서 수집된 데이터를 스트림 형태로 근접 빈발 패턴 마이닝 한다. 최신 근접 빈발 패턴 마이닝은 스트림 형태의 데이터나 다양한 마이닝 기법에 적용되고 있으며 활발한 연구가 진행되고 있다. 최근의 근접 패턴 마이닝 기법들은 통계적 또는 수학적 기법을 통해 적합한 분야에 적용한다. 본 논문은 기존의 Apriori 알고리즘을 기반으로 하는 Chernoff 알고리즘인 FDPM [10] 과 FDPM에 트리구조의 마이닝인 FP-growth를 결합한 FDPM+를 성능 분석을 통해 근접 빈발 패턴 마이닝 알고리즘에 대한 장단점 및 효율성에 대하여 평가한다. 본 논문은 2장에서 Chernoff bound알고리즘을 사용한 근접 빈발 패턴 마이닝인 FDPM에 대하여 논하고 FP-growth를 결합한 FDPM+에 대하여 이야기 한다. 3장에서는 FDPM과 FDPM+의 성능평가에 대하여, 4장에서는 근접 빈발 패턴 마이닝에 대한 결론을 맺는다.

2. 확률 기법에 기반한 근접 빈발 패턴마이닝

2.1 Chernoff bound 기반의 빈발 패턴 마이닝

확률 기법에 기반한 근접 빈발 패턴 마이닝 알고리즘으로 FDPM [10] 가 제안되었다. FDPM은 Chernoff bound [10] 를 사용한 근접 빈발 패턴 마이닝이다. Chernoff bound는 Bernoulli trial에 의하여 발생하는 현상을 기반으로 한다. Bernoulli trial은 현상 X에 대하여 발생할 수 있는 시도 $o_1,o_2,o_3,\cdots,o_n$에 대하여 임의의 k번째 시도일 때, o_k가 X가 되는 현상일 때 o_k=1이라고 정의하고, X가 발생하지 않았을 때, o_k=0이라고 정의한다. 이때, 총n번의 시도에서 X가 발생한 현상의 합을 $\hat{X}$로 정의하면, n번의 시도에서 실제로 X 현상이 발생한 확률(비율) x는 $\hat{X}$/n 으로 정의 할 수 있다. 이때 Chernoff bound 는 다음 수식과 같이 정의된다.

$$\Pr\{|\hat{x}-x|\geq xr\}\leq 2e^{\frac{-nxr^2}{2}} \quad (1)$$

여기서 xr을 실제 X가 발생한 횟수와 X가 발생할 확률에 대한 오차라고 정의하고 λ는 신뢰율 이라고 하며 $\lambda=2e^{\frac{-nxr^2}{2}}$ 로 정의하고 xr에 대하여 정리하면 다음과 같은 수식이 만들어진다.

$$xr=\sqrt{\frac{2xln\left(\frac{\lambda}{2}\right)}{n}} \quad (2)$$

여기서 오차 xr은 수식 (1)에 의하여 항상 λ 보다 작거나 같은 특성을 가진다. 그래서 xr을 이용한 수치는 항상 λ 보다 작은 오차를 가지게 된다. FDPM은 오차 xr을 이용하여 근접 빈발 패턴 마이닝을 한다. 빈발 패턴 마이닝을 하기 위해서는 데이터베이스의 정보를 메모리에 로드해야 하며 데이터베이스가 매우 큰 경우 많은 메모리를 사용하게 된다. 그래서 FDPM은 데이터베스 전체를 한 번에 마이닝 하는 것이 아닌 데이터베이스를 구간으로 나누어 빈발 패턴을 마이닝을 하고 구간별로 마이닝 된 빈발 패턴을 이용하여 데이터베이스 전체의 빈발 패턴을 추정하는 방법으로 마이닝 한다. FDPM은 Apriori 알고리즘을 이용하여 구간별로 마이닝 된 결과를 pool 에 저장하고 구간별로 마이닝 된 결과를 업데이트 한다. 그러나 pool에 저장된 빈발 패턴 후보들을 제한 없이 저장하면 많은 메모리를 사용하게 된다. FDPM은 메모리 사용량의 효율성을 위해 pool의 용량 한계를 설정한다. 그 용량 한계는 T_l 의 패턴 수를 가진다. T_l은 다음과 같은 수식에 의하여 계산된다.

$$T_l=\frac{2+2\ln\left(\frac{\lambda}{2}\right)}{P_k} \quad (3)$$

P_k는 데이터베이스 전체의 트랜잭션 수, λ 는 신뢰율이다. 이 수식을 이용하여 pool에 저장된 패턴의 수를 관리한다. 만약 pool에 저장된 패턴의 수가 T_l 보다 크면 (Minimum support – xr)를 기준으로 pool 내부의 패턴을 프루닝 한다. 이 과정이 끝나면 다른 batch를 접근하여 같은 작업을 수행한 뒤 모든 batch가 마이닝 되면 pool 에 남은 패턴중 Minimum support보다 큰 빈도수를 가지는 패턴이 근접 빈발 패턴이 된다.

2.2 트리 기반의 알고리즘을 이용한 근접 빈발 패턴 마이닝

FDPM은 Apriori 기반으로 마이닝을 진행한다. 최신동향의 빈발 패턴 마이닝 알고리즘은 대부분 FP-growth와 같이 트리 기반의 마이닝 알고리즘이다[6]. 그리고 Apriori 기반보다 FP-growth 기반으로 FDPM을 구성하는 것이 유리하다 [4]. FDPM+는 트리 형태의 구조를 가지는 FP-growth 기반의 마이닝 알고리즘을 통해 근접 빈발 패턴 마이닝을 하는 알고리즘이다. 즉, FDPM+는 FDPM의 Apriori 대신 FP-growth를 기반으로 빠른 마이닝 속도를 낼 수 있도록 한다. (그림 1)은 FDPM+의 알고리즘을 보여준다. 마이닝 전에 batch의 크기를 지정한다(line 2). k는 batch 의 크기를 보정하는 상수로 프로그램이 시작하기 전에 사용자로부터 입력 받는다. FDPM+는 batch 내부에서 빈발 가능성이 있는 패턴을 구하기 위해 현재 batch 내부의 트랜잭션을 FP-growth로 마이닝 한다(line 5). batch 의 트랜잭션을 이용하여 FP-tree를 생성한다. 그 다음 growth 방법을 이용하여 빈발 패턴을 구하는데 여기에서 사용하는 Minimum support는 데이터베이스 전체의 Minimum support가 s일 때, (s – xr)을 사용한다. 이렇게 되면 [1, s-xr] 구간의 빈도수를 가지는 패턴을 구할 수 있다. 구해진 패턴은 P에 저장하는데 길이 순서에 따라 따로 저장한다. 즉, 길이가 1인 패턴부터 시작해서 길이가 n인 패턴까지 모두 저장한다. 마이닝 결과 후 pool 역할을 하는 F에 P의 패턴을 업데이트 하는데 패턴 길이가 짧은 것부터 순서대로 업데이트 한다(line 6). 그 뒤, F에 저장된 패턴이 $c_u \times T_l$ 보다 크면 FDPM 알고리즘에 따라 빈발하지 않은 패턴 후보를 프루닝 한다(line 7). c_u는 Minimum support가 주어졌을 때 마이닝 과정 중 F의 크기가 가장 클 때의 T_l 의 배율을 뜻한다. 즉 F의 크기가 가장 커졌을 때, T_l를 기준으로 몇 개의 T_l가 있어야 모든 후보 패턴을 저장할 수 있는지에 대한 비율이다. 이 값은 마이닝 수행 전에 사용자가 지정한다. 메모리 공간이 필요한 모든 batch가 마이닝 되면 pool에 저장된 후보 빈발 패턴 중에서 Minimum support 보다 같거나 큰 패턴이 빈발 패턴이 된다(line 11).

Algorithm FDPM+

1. let T_l be the required number of observations
2. batch size ← $k \times T_l$
3. $n \leftarrow 0$, $F \leftarrow 0$, $P \leftarrow 0$
4. **For** every batch size transactions do
5. Mining potential frequent patterns using FP-growth
6. $F \leftarrow P \cup F$
7. prune potential infreqeunt pattern from F further if $|F| > cu \times T_l$
8. $P \leftarrow 0$
9. $n \leftarrow n +$ batch size
10. **End For**
11. output the patterns in F whose count ≥ Minimum support on demand

(그림 1) FDPM+ 알고리즘
(Figure 1) FDPM+ algorithm

3. 근접 패턴 마이닝의 성능 분석

본 논문은 통계적인 방법을 이용하여 빈발한 패턴을 마이닝 하는 근접 패턴 마이닝의 성능을 비교한다. 비교 대상은 Chernoff bound를 이용한 FDPM과 FDPM에 트리 형태 마이닝 알고리즘인 FP-growth를 접목한 FDPM+이다. 두 알고리즘은 C++로 구현되었으며 FDPM에 사용되는 Apriori는 http://fuzzy.cs.uni-magdeburg.de/~borgelt/software.html#accoc에서 다운로드하였다. 그리고 FDPM+에 사용하는 FP-growth는 http://adrem.ua.ac.be/~goethals/software/에서 다운로드하였다. 성능평가에 사용된 컴퓨터의 환경은 Intel Core i5 3.3GHz, 8Gbyte RAM이다. 운영체제는 Windows 7을 이용하였다. 각 마이닝 기법의 마이닝 수행 시간과 메모리 사용량, 확장성[10]을 평가하고 [10]의 성능평가에 준하여 진행한다. (표 1)은 마이닝 시간과 메모리 사용량을 평가하기 위한 실제 데이터 셋이다. Connect 데이터는 온라인 네트워크의 접속 정보를 나타내고, Retail은 Belgian retail store의 물품 판매 데이터이다. 두 데이터 셋은 http://fimi.cs.helsinki.fi/data/ [14] 에서 다운로드 할 수 있다. (표 2)는 알고리즘의 확장성 평가를 위해 준비한 데이터 셋이다. T10I4D1000K ~ T10I4D5000K 데이터 셋은 트랜잭션이 증가하는 데이터 셋이다. 5개의 데이터 셋은 서로 비슷한 정보를 가지고 있으며 트랜잭션의 크기가 일정하게 증가하므로 트랜잭션의 증가에 따른 확장성 평가를 할 수 있다.

(표 1) 실제 데이터셋
(Table 1) Real datasets

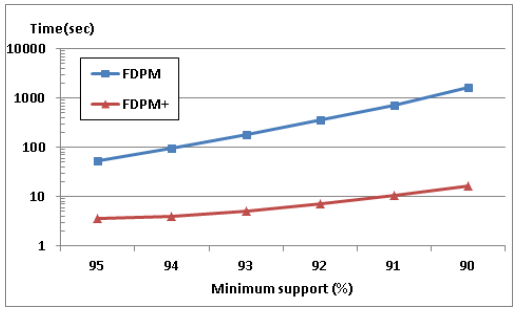
Dataset	#Items	avg.Length	#Transactions
Connect	129	43	67557
Retail	16469	10.3	88162

(표 2) 확장성 평가용 데이터셋
(Table 2) Datasets for scalability test

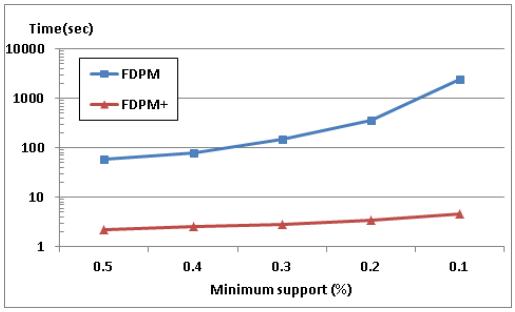
Dataset	#Items	avg.Length	#Transactions
T10I4D1000K	10000	10	1000000
T10I4D2000K	10000	10	2000000
T10I4D3000K	10000	10	3000000
T10I4D4000K	10000	10	4000000
T10I4D5000K	10000	10	5000000

3.1 마이닝 시간 평가

마이닝 시간에 대한 평가는 표 1의 실제 데이터 셋을 이용하였으며 시간 측정 방법은 알고리즘을 시작한 시각부터 알고리즘이 결과를 내고 종료되는 시각까지의 시간을 측정하였다 [10]. (그림 2)는 Connect 데이터에서 minimum support를 95%~90%로 감소시키고 batch의 크기를 10000으로 하며, λ값을 0.1로 하였을 때 마이닝 시간을 보여준다. Connect 데이터는 대부분 비슷한 정보를 가지는 트랜잭션으로 이루어져 있다. 평가 결과 FDPM+는 FDPM에 비하여 빠른 마이닝 속도를 보여주었다. FDPM은 많은 후보 패턴 생성과 데이터베이스 스캔으로 인해 마이닝 시간이 오래 걸리지만 FDPM+ 는 Apriori보다 효율적인 FP-growth를 기반으로 하기 때문에 2번의 데이터베이스 스캔과 후보패턴을 생성하지 않는다. 그래서 FDPM+는 적은 마이닝 시간으로도 근접 빈발 패턴 마이닝을 할 수 있다.



(그림 2) 마이닝 시간 평가(Connect)
(Figure 2) Runtime test(Connect)



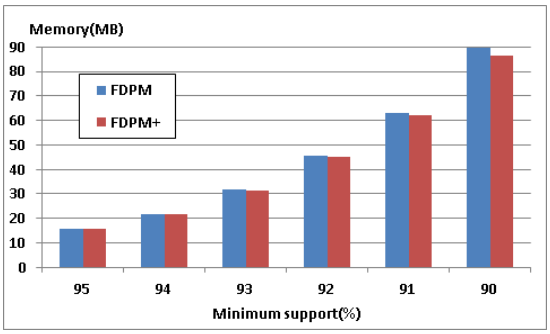
(그림 3) 마이닝 시간 평가(Retail)
(Figure 3) Runtime test(Retail)

(그림 3)은 Retail 데이터 셋에서 Minimum support를 0.5~0.1%로 변화 시키면서 batch의 크기를 20000으로 하고, λ값을 0.1로 하였을 때 마이닝 시간을 평가 하였다. Retail 데이터 셋은 트랜잭션의 수가 많으며 특히 아이템의 수가 많은 특징을 가지고 있다. 평가 결과 FDPM은 많은 아이템 수로 인하여 후보 패턴을 많이 만들게 되어서 Minimum support가 증가할수록 마이닝 시간이 급격하게 증가한다. 반대로 FDPM은 후보 패턴을 만들지 않는 FP-growth를 기반으로 하기 때문에 FDPM보다 작은 마이닝 시간이 소요되었다.

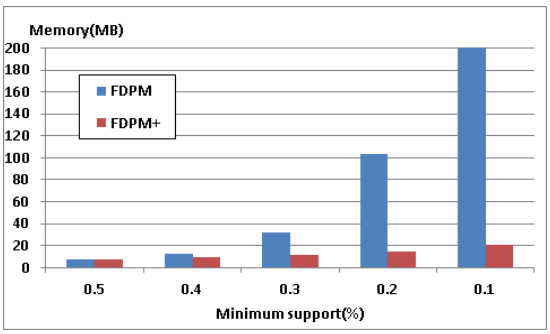
3.2 메모리 사용량 평가

다음 평가는 마이닝 수행 시 메모리 사용량에 대한 것이다. 평가에 사용되는 데이터 셋은 표 1의 실제 데이터 셋을 이용하였다. 메모리 사용량에 대한 평가는 알고리즘이 시작되는 시각부터 종료되는 시각까지 사용한 메모리 중에서 메모리를 최대로 사용한 수치를 측정하였다[4]. 즉 알고리즘을 수행하면서 가장 많은 메모리를 사용한 수치를 측정한 것이다. (그림 4)는 Connect 데이터에서 batch의 크기를 10000, λ를 0.1로 고정한 상태에서 Minimum support를 95~90%까지 감소시키면서 최대 메모리 사용량을 평가하였다. Connect 데이터는 트랜잭션이 대부분 비슷하므로 Minimum support가 감소하더라도 메모리에 로드되는 데이터의 수가 비슷한 특징을 가지게 된다. 평가 결과 FDPM과 FDPM+이 서로 비슷한 메모리 사용량을 보여주었다. 그러나 93%부터 FDPM+가 더 적은 메모리를 사용하기 시작하고 Minimum support가 낮아질수록 FDPM+가 더 적은 메모리를 사용한다. 그 이유는 FDPM이 후보 패턴을 만드는 양이 Minimum support가 감소하면서 증가하기 때문이다. (그림 5)는 Retail 데이터에서

batch의 크기를 20000으로 하고, λ를 0.1로 고정한 채 Minimum support를 0.5~0.1% 변경하면서 최대 메모리 사용량을 평가한 결과를 보여준다. Retail 데이터는 아이템의 수가 많다. 그래서 마이닝 시 후보 패턴을 생성해야하는 FDPM은 Minimum support가 감소하면서 메모리에 로드되는 아이템의 수가 많아져 후보패턴이 기하급수적으로 생성되고 메모리 사용량이 급격하게 상승되는 모습을 볼 수 있다. FDPM+는 트리에 아이템과 패턴을 저장하기 때문에 FDPM에 비해 메모리 사용량이 작다.



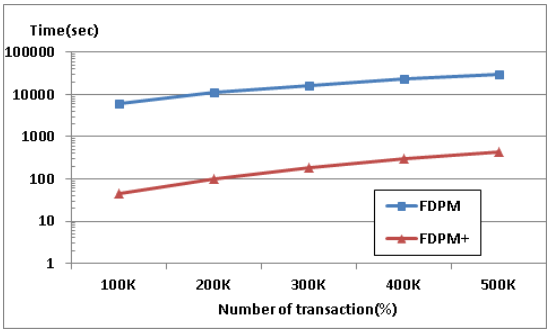
(그림 4) 메모리 사용량 평가(Connect)
(Figure 4) Memory test(Connect)



(그림 5) 메모리 사용량 평가(Retail)
(Figure 5) Memory test(Retail)

3.3 확장성 평가

마지막 평가는 확장성에 대한 평가이다. 확장성평가는 서로 비슷한 정보를 가지는 데이터를 이용하여 트랜잭션의 크기를 증가시키면서 마이닝 시간과 메모리 사용량을 평가하는 방법으로 데이터베이스의 크기가 커지면서 안정적으로 알고리즘이 수행되는지 평가하는 방법이다. (그림 6)은 (표 2)의 데이터에서 트랜잭션이 증가하는 데이터를 이용하여 마이닝 시간을 평가한 결과이다. 평가에 사용한 데이터의 트랜잭션은 100만개에서 500만개로 증가 시켰다. 그리고 Minimum support는 0.03%에 고정하여 평가하였다. 평가 결과 두 알고리즘 모두 비슷한 시간 증가율을 보여주었다. FDPM은 트랜잭션이 증가함에 따라 FDPM과 FDPM+모두 트랜잭션의 크기가 증가된 만큼의 시간 증가율을 보여주었다. 트랜잭션이 100만개에서 FDPM+이 44.777초, FDPM이 6099.93초가 소요되었으며 트랜잭션이 200만개일 때는 FDPM+가 97.955초, FDPM이 12368.54초가 소요 되어 약 2배의 마이닝 시간이 증가하였다. 트랜잭션이 증가함에 따라 선형적으로 마이닝 시간이 증가하므로 안정적으로 트랜잭션을 처리하는 것으로 판단 할 수 있다.



(그림 6) 확장성 평가(트랜잭션 증가:시간)
(Figure 6) Scalability test(Runtime)

4. 결 론

본 논문은 확률 기법에 기반한 근접 패턴 마이닝 기법에 대하여 분석하고 성능평가를 수행하였다. 성능평가에서는 기존의 Apriori가 적용된 FDPM알고리즘과 트리 구조 형태의 FP-growth가 적용된 FDPM+ 알고리즘에 대한 마이닝 시간, 메모리 사용량, 확장성 평가를 수행하였다. 성능평가 결과 FDPM+가 마이닝 수행 시간에 대하여 FDPM과 비교하여 빠른 속도를 보여주었다. 메모리 사용량은 Accidents 데이터를 제외한 나머지 데이터에 대하여 Minimum support가 낮아질수록 효율적인 성능을 보여주었다. 확장성 평가에서는 FDPM+가 FDPM에 비하여 더 빠른 마이닝 속도를 가지면서도 트랜잭션이 증가함에 따라 안정적으로 마이닝 시간이 증가하였다. 성능평가를 전체적으로 분석하면 FDPM+가 FDPM에 비하여 마이닝 속도와 메모리 사용량 그리고 확장성에 대하여 일부를 제

외하고 전반적으로 높은 효율성을 보여주었다. 실시간 시스템은 빠른 마이닝 속도가 필요로 하기 때문에 근접 패턴마이닝이 적합할 것으로 보이며 센서 네트워크의 경우 적은 자원으로 대용량의 데이터를 처리해야 하기 때문에 메모리 제한이 있는 근접 빈발 패턴 마이닝이 필요하다. 본 논문에서 소개하고 구현한 FDPM+는 FDPM에 비하여 매우 빠른 마이닝 속도를 보여주기 때문에 기존의 Chernoff bound 기반의 근접 빈발 패턴 마이닝보다. 빠르게 데이터 분석을 수행할 수 있다. 또한 효율적인 근접 빈발 패턴 마이닝이 연구되고 있는 상태이고 이러한 근접 패턴 마이닝이 다양한 분야에 적용된다면 네트워크나 대용량 처리에 매우 큰 변화를 보여줄 것으로 예상한다.

참 고 문 헌(Reference)

[1] R. Agrawal and R. Srikant, “Fast Algorithms of Mining Association Rules”, International conference on Very Large Data Bases(VLDB), vol. 20, pp.487-499, 1994.

[2] T. Calders, C. Garboni, B. Goethals, Approximation of Frequentness Probability of Itemsets in Uncertain Data. International Conference on Data Mining (ICDM), pp. 749-754, 2010.

[3] C. Chen, X. Yan, F. Zhu and J. Han, gApprox: Mining Frequent Approximate Patterns from a Massive Network. ICDM, pp.445-450, 2007.

[4] J. Han, J. Pei, Y. Yin and R. Mao, “Mining frequent patterns without candidate generation : a frequent pattern tree approach”, Data Mining and Knowledge Discovery, vol 8, pp.53-87. 2004.

[5] J. Han, H. Cheng, D. Xin and X.Yan, Frequent pattern mining : current status and future directions, Data Mining and Knowledge Discovery(DMKD), vol.15, no.1, pp. 55-86, Aug 2007.

[6] C.W. Li, K.F. Jea, An adaptive approximation method to discover frequent itemsets over sliding-window-based data streams, Expert System with Applications(ESWA) 38(10), pp.13386-13404, 2011.

[7] M. Ren, L. Guo, Mining Recent Approximate Frequent Items in Wireless Sensor Networks, Fuzzy Systems and Knowledge Discovery, pp. 463-467, 2009.

[8] P. Wong, T. Chan, M. H. Wong and K. Leung, Predicting Approximate Protein-DNA Binding Cores Using Association Rule Mining, ICDE pp.965-976, 2012.

[9] R.C. Wong and A.W. Fu, “Mining top-K frequent itemsets from data streams”, Data Mining Knowledge Discovery. Vol.13, pp.193-217, 2006.

[10] J.X. Yu, Z. Chong, H. Lu and A. Zhou, False Positive or False Negative: Mining Frequent Itemsets from High Speed Transactional Data Streams, International conference on Very Large Data Bases(VLDB) vol. 30, pp.204-215, Aug. 2004.

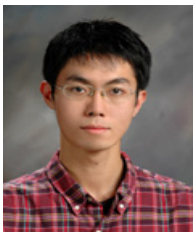
[11] U. Yun and K. Ryu, Approximate Weight frequent pattern mining with/without noisy environments, Knowledge-Based System, vol. 24, no. 1, pp. 73-82, Feb 2011.

[12] Y. Zhao, C. Zhang and S. Zhang, Efficient Frequent Itemsets Mining by Sampling, Advances in Intelligent IT: Active Media Technology, pp.112-117, 2006.

[13] F. Zhu, X. Yan, J. Han and P.S. Yu, Efficient Discovery of frequent Approximate Sequential Patterns, International Conference on DataMining (ICDM), pp.751-756, Dec 2007.

[14] Frequent itemset Mining dataset repository. Availble at (http://fimi.cs.helsinki.fi/data/)

◐ 저 자 소 개 ◑



편 광 범

2010년 충북대학교 컴퓨터공학전공 학사. (공학사)
2012년 충북대학교 대학원 컴퓨터과학 석사. (공학석사)
2012년~현재 충북대학교 대학원 컴퓨터과학 박사과정
관심분야 : 데이터마이닝, 정보검색, 데이터베이스
E-mail: pyungb@chungbuk.ac.kr

윤 은 일

1997년 고려대학교 이학석사. (이학석사)
1997년~2006년 한국통신 멀티미디어연구소 전임/선임연구원.
2005년 Texas A&M Univ. 공학박사 (공학박사)
2005년~2006년 Texas A&M Univ. 포스닥연구원.
2006년~2007년 한국전자통신연구원, 선임연구원.
2007년~현재: 충북대학교 전자정보대학 소프트웨어학과 부교수
관심분야 : 데이터마이닝, 정보검색, 데이터베이스
E-mail: yunei@chungbuk.ac.kr