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

트랜잭션 가중치 기반의 빈발 아이템셋 마이닝 기법의 성능분석☆

Performance analysis of Frequent Itemset Mining Technique based on Transaction Weight Constraints

윤 은 일1* 편 광 범1
Unil Yun Gwangbum Pyun

요 약

최근, 아이템들의 가치를 고려한 빈발 아이템셋 마이닝 방법은 데이터 마이닝 분야에서 가장 중요한 이슈 중 하나로 활발히 연구되어왔다. 아이템들의 가치를 고려한 마이닝 기법들은 적용 방법에 따라 크게 가중화 빈발 아이템셋 마이닝, 트랜잭션 가중치 기반의 빈발 아이템셋 마이닝, 유틸리티 아이템셋 마이닝으로 구분된다. 본 논문에서는 트랜잭션 가중치 기반의 빈발 아이템셋 마이닝들에 대해 실증적인 분석을 수행한다. 일반적으로 트랜잭션 가중치 기반의 빈발 아이템셋 마이닝 기법들은 데이터베이스 내 아이템들의 가치를 고려함으로써 트랜잭션 가중치를 계산한다. 또한, 그 기법들은 계산된 각 트랜잭션의 가중치를 바탕으로 가중화 빈발 아이템셋들을 마이닝 한다. 트랜잭션 가중치는 트랜잭션 내에 높은 가치의 아이템이 많이 포함 될수록 높은 값으로 나타나기 때문에 우리는 각 트랜잭션의 가중치의 분석을 통해 그 가치를 파악할 수 있다. 우리는 트랜잭션 가중치 기반의 빈발 아이템셋 마이닝 기법 중에서 가장 유명한 알고리즘인 WIS와 WIT-FWIs, WIT-FWIs-MODIFY, WIT-FWIs-DIFF의 장·단점을 분석하고 각각의 성능을 비교한다. WIS는 트랜잭션 가중치 기반의 빈발 아이템셋 마이닝의 개념과 그 기법이 처음 제안된 알고리즘이며, 전통적인 빈발 아이템셋 마이닝 기법인 Apriori를 기반으로 하고 있다. 또 다른 트랜잭션 가중치 기반의 빈발 아이템셋 마이닝 방법인 WIT-FWIs와 WIT-FWIs-MODIFY, WIT-FWIs-DIFF는 가중화된 빈발 아이템셋 마이닝을 더 효율적으로 수행하기 위해 격자구조(Lattice) 형태의 특별한 저장구조인 WIT-tree를 이용한다. WIT-tree의 각 노드에는 아이템셋 정보와 아이템셋이 포함된 트랜잭션의 ID들이 저장되며, 이 구조를 사용함으로써 아이템셋 마이닝 과정에서 발생되는 다수의 데이터베이스 스캔 과정이 감소된다. 특히, 전통적인 알고리즘들이 수많은 데이터베이스 스캔을 수행하는 반면에, 이 알고리즘들은 WIT-tree를 이용해 데이터베이스를 오직 한번만 읽음으로써 마이닝과정에서 발생 가능한 오버헤드 문제를 해결한다. 또한, 공통적으로 길이 N의 두 아이템셋을 이용해 길이 N+1의 새로운 아이템셋을 생성한다. 먼저, WIT-FWIs는 각 아이템셋이 동시에 발생되는 트랜잭션들의 정보를 활용하는 것이 특징이다. WIT-FWIs-MODIFY는 조합되는 아이템셋의 정보를 이용해 빈도수 계산에 필요한 연산을 줄인 알고리즘이다. WIT-FWIs-DIFF는 두 아이템셋 중 하나만 발생한 트랜잭션의 정보를 이용한다. 우리는 다양한 실험환경에서 각 알고리즘의 성능을 비교분석하기 위해 각 트랜잭션의 형태가 유사한 dense 데이터와 각 트랜잭션의 구성이 서로 다른 sparse 데이터를 이용해 마이닝 시간과 최대 메모리 사용량을 평가한다. 또한, 각 알고리즘의 안정성을 평가하기 위한 확장성 테스트를 수행한다. 결과적으로, dense 데이터에서는 WIT-FWIs와 WIT-FWIs-MODIFY가 다른 알고리즘들보다 좋은 성능을 보이고 sparse 데이터에서는 WIT-FWI-DIFF가 가장 좋은 효율성을 갖는다. WIS는 더 많은 연산을 수행하는 알고리즘을 기반으로 했기 때문에 평균적으로 가장 낮은 성능을 보인다.

☞ 주제어 : 트랜잭션 가중치, 데이터 마이닝, 빈발 아이템셋 마이닝, 성능평가, 확장성

ABSTRACT

In recent years, frequent itemset mining for considering the importance of each item has been intensively studied as one of important issues in the data mining field. According to strategies utilizing the item importance, itemset mining approaches for discovering itemsets based on the item importance are classified as follows: weighted frequent itemset mining, frequent itemset mining using transactional weights, and utility itemset mining. In this paper, we perform empirical analysis with respect to frequent itemset mining algorithms based on transactional weights. The mining algorithms compute transactional weights by utilizing the weight for each item in large databases. In addition, these algorithms discover weighted frequent itemsets on the basis of the item frequency and weight of each transaction. Consequently, we can see the importance of a certain transaction through the database analysis because the

[1] Dept. of Computer Engineering, Sejong University, Seoul, 143-747, Korea
[*] Corresponding author (yunei@sejong.ac.kr)
[Received 28 August 2014, Reviewed 17 September 2014, Accepted 21 January 2014]
☆ 본 연구는 미래창조과학부 및 정보통신산업진흥원의 ICT/SW 창의연구과정의 연구결과로 수행되었으며(NIPA-2014-H0502-14-3008) 또한, 2014년도 정부 교육과학기술부의 재원으로 한국 연구재단(NRF)의 지원을 받아 수행된 연구사업이며(NRF No.2013-005682), 중소기업청에서 지원하는 2015년도 산학연협력 기술개발사업(No. C0232102)의 연구수행으로 인한 결과물임을 밝힙니다.
☆ 본 논문은 2014년도 인터넷정보학회 춘계학술발표대회 우수 논문 추천에 따라 확장 및 수정된 논문임

weight for the transaction has higher value if it contains many items with high values. We not only analyze the advantages and disadvantages but also compare the performance of the most famous algorithms in the frequent itemset mining field based on the transactional weights. As a representative of the frequent itemset mining using transactional weights, WIS introduces the concept and strategies of transactional weights. In addition, there are various other state-of-the-art algorithms, WIT-FWIs, WIT-FWIs-MODIFY, and WIT-FWIs-DIFF, for extracting itemsets with the weight information. To efficiently conduct processes for mining weighted frequent itemsets, three algorithms use the special Lattice-like data structure, called WIT-tree. The algorithms do not need to an additional database scanning operation after the construction of WIT-tree is finished since each node of WIT-tree has item information such as item and transaction IDs. In particular, the traditional algorithms conduct a number of database scanning operations to mine weighted itemsets, whereas the algorithms based on WIT-tree solve the overhead problem that can occur in the mining processes by reading databases only one time. Additionally, the algorithms use the technique for generating each new itemset of length N+1 on the basis of two different itemsets of length N. To discover new weighted itemsets, WIT-FWIs performs the itemset combination processes by using the information of transactions that contain all the itemsets. WIT-FWIs-MODIFY has a unique feature decreasing operations for calculating the frequency of the new itemset. WIT-FWIs-DIFF utilizes a technique using the difference of two itemsets. To compare and analyze the performance of the algorithms in various environments, we use real datasets of two types (i.e., dense and sparse) in terms of the runtime and maximum memory usage. Moreover, a scalability test is conducted to evaluate the stability for each algorithm when the size of a database is changed. As a result, WIT-FWIs and WIT-FWIs-MODIFY show the best performance in the dense dataset, and in sparse dataset, WIT-FWI-DIFF has mining efficiency better than the other algorithms. Compared to the algorithms using WIT-tree, WIS based on the Apriori technique has the worst efficiency because it requires a large number of computations more than the others on average.

☞ keyword : Transaction weight, Data Mining, Frequent Itemset mining, Performance evaluation, Scalability

1. 서 론

최신의 데이터마이닝의 이슈는 다양하거나 특수한 어플리케이션에 사용되는 데이터베이스로부터 중요한 정보를 마이닝 하는 방법에 대한 것이다. 반면에, 전통적인 빈발 아이템셋 마이닝은 다양한 정보를 고려하지 않고 단순하게 데이터의 아이템셋(또는 패턴)만을 이용해 빈발한 패턴을 마이닝 하는 Apriori [1] 및 FP-growth [3]와 같은 방법들이 주류였다. 또한, 전통적인 패턴 마이닝을 응용함으로써 다양한 환경에서의 패턴 마이닝 방법들 [7, 9, 10]이 제안되어왔다. 그러나 비즈니스 환경에서 상품의 판매 데이터를 분석하는 경우 물건의 매매가격 및 수익 등을 고려해야 하기 때문에 일반적인 빈발 패턴 마이닝 방법으로는 정확한 패턴들을 마이닝 할 수 없는 문제가 발생한다. 그래서 상품에 해당하는 아이템과 상품의 수익에 해당하는 가중치를 동시에 고려하면서 빈발 패턴을 마이닝 할 수 있는 다양한 기법들 [2, 5]이 제안되었다. 아이템의 가중치와 아이템들의 빈도수를 동시에 고려하면서 마이닝하는 방법들은 WFIM(Weighted Frequent Itemset Mining) [12]과 FWI(Frequent Weighted Itemset) 기반의 마이닝 알고리즘 [8] 그리고 HUFIM(High Utility Itemset Mining) [6]으로 나눌 수 있다. WFIM에서 패턴의 가중화 빈도수를 계산하는 방법은 다음과 같다. WFIM은 패턴 성장 방법[3]에 의해 패턴의 빈도수를 계산한다. 그리고 WFIM은 계산된 패턴의 빈도수에 패턴의 평균 가중치를 곱한다. 패턴의 평균 가중치는 패턴에 포함된 아이템들에 대한 가중치들의 평균값이다. 즉, WFIM은 독립적으로 패턴 자신이 가진 아이템들의 중요도만을 고려하여 가중화 빈도수를 계산한다. HUFIM은 트랜잭션 내 아이템가치와 수량을 동시에 고려한다. 수량을 고려하기 때문에 추가적으로 연산이 필요하며 더 많은 정보를 포함한 결과를 얻을 수 있지만 WFIM과 동일하게 아이템의 가치를 독립적으로 고려한다.

FWI기반의 마이닝 알고리즘의 가중화 빈도수 계산 방법은 WFIM 및 HUFIM과 다르다. FWI 기반의 마이닝 알고리즘은 트랜잭션들의 각 가중치를 계산하고 트랜잭션 가중치를 기반으로 패턴의 가중화 빈도수를 계산한다. 두 알고리즘은 이러한 차이점을 가지고 있으며 본 논문에서는 FWI 기반의 마이닝 알고리즘에 대하여 분석하고 성능을 평가한다.

본 논문은 2장에서 가중치 기반 빈발 패턴 알고리즘들을 소개하고, 3장에서 트랜잭션 가중치 기반의 마이닝 알고리즘들에 대해 분석한다. 각 알고리즘들의 분석된 내용을 바탕으로 다양한 데이터셋들을 이용해 4장에서 마이닝 시간과 최대 사용 메모리, 확장성에 대한 성능평가를 진행한다. 끝으로, 성능분석 결과를 기반으로 5장에서 결론과 함께 향후연구방향을 제시한다.

2. 관련연구

가중치 기반 패턴 마이닝 방법은 전통적인 빈발 패턴 마이닝의 한 변형으로 현실세계의 특수한 제약사항들을

고려하기 위해 제안되었다. 일반적으로, 각 아이템에 해당하는 가중치들을 기반으로 아이템들의 집합인 아이템셋에 대한 가중치를 이용해 전통적인 빈발 패턴 마이닝에서 사용하는 빈도수 보다 복합적으로 아이템셋에 대한 추가적인 정보를 반영하는 특징을 갖는다. WFIM [12]는 다음의 수식을 이용해 패턴 P에 포함된 각 아이템, i, 의 가중치 평균을 구한다.

$$Average\,weight(P) = \sum Weight(i)\,/length\,of\,P$$

계산된 평균값은 P의 가중치가 되며 이를 P의 빈도수에 곱해 마이닝 과정에서 사용한다. 또 다른 알고리즘으로, HUFIM [6]은 각각의 아이템에 대한 공통적인 가중치와 함께 트랜잭션 내부에 개별적인 추가 가중치를 고려하는 알고리즘이다. 기존의 가중치 기반 패턴 마이닝 알고리즘들이 하나의 가중치만 고려했다면, HUFIM은 두 가지의 가중치를 통해 좀 더 복합적으로 현실세계를 반영할 수 있다. 특히, 이 방법은 비즈니스 환경에서 상품의 판매 데이터를 분석할 때 유용하게 사용된다. 최근에는 IWI-Miner [2]와 같이 빈발하지 않은 패턴들을 추출할 때에도 가중치 조건을 고려함으로써 더 효율적으로 결과 패턴의 품질을 높이는 방법들이 제안되고 있다.

FWI 기반의 마이닝 알고리즘에서 트랜잭션 가중치는 트랜잭션에 포함된 아이템들의 각 가중치에 대한 평균값이다. 높은 가중치를 가진 아이템들이 많이 포함되어 있는 트랜잭션은 높은 트랜잭션 가중치를 가지고 낮은 가중치를 가진 아이템들이 많이 포함된 트랜잭션은 낮은 트랜잭션 가중치를 가진다. 그러므로 트랜잭션 가중치는 트랜잭션에 포함된 아이템들의 각 가중치에 종속적이다. FWI 기반의 마이닝 알고리즘에서 패턴의 가중화 빈도수는 패턴이 포함된 트랜잭션들에 대한 가중치들의 합이다. 그러므로 높은 가중화 빈도수를 가지는 패턴들은 중요한 패턴들로 볼 수 있으며 FWI 기반의 마이닝 알고리즘은 이러한 중요 패턴들을 마이닝 한다. FWI는 다양한 곳에 유용하게 응용될 수 있다. 예를 들면, 소매점의 물건 판매 데이터는 아이템 가중치가 되는 상품 수익과 손님이 소매점을 한번 방문했을 때의 상품 구매 목록이다. 마케팅 관리자는 소매점의 이윤과 관련된 정보를 얻기 위해 물건 판매 데이터베이스를 분석한다. 마케팅 관리자가 원하는 구매 패턴을 분석하기 위한 방법으로 FWI 기반의 마이닝 알고리즘은 유용하게 사용될 수 있다. FWI 기반의 마이닝 알고리즘의 트랜잭션 가중치는 손님이 매장에 한번 방문했을 때 구매한 상품들에 대한 이윤이다. 만약 특정 구매 패턴이 높은 이윤을 가지는 트랜잭션에 다수 포함되어 있으면 이 구매 패턴의 가중화 빈도수는 높고 FWI가 된다. 그래서 FWI를 마이닝 하는 알고리즘을 통해 물건 판매 데이터로부터 마케팅에 필요한 정보를 얻을 수 있다.

3. 트랜잭션 가중치 기반의 마이닝 알고리즘 분석

3장에서는 FWI 기반의 실제 알고리즘에 대하여 분석한다. 분석하는 알고리즘으로 Apriori 알고리즘을 기반으로 하여 FWI를 마이닝 하는 알고리즘인 WIS [8]를 분석하고 이를 격자구조(Lattice) [13] 기반의 마이닝 방법으로 발전시킨 WIT-FWIs [11] 방법에 대하여 분석한다.

3.1 WIS 알고리즘

WIS는 FWI가 처음으로 제안된 논문이다. FWI는 데이터베이스를 스캔하여 트랜잭션 가중치를 계산하고 이를 기반으로 Apriori 알고리즘에 따라 마이닝을 수행한다. 데이터베이스에 포함된 아이템에 대한 가중화 빈도수를 계산한다. 이중 최소 빈도수보다 작은 아이템을 제외하고 나머지 아이템들이 길이가 1인 빈발 패턴이 된다. 앞서 발견된 길이가 L인 빈발 패턴을 이용해 길이가 L+1인 후보 패턴을 생성하고 데이터베이스를 스캔해 후보 패턴이 포함된 트랜잭션을 찾는다. 그리고 트랜잭션 가중치를 이용해 패턴의 가중화 빈도수를 계산한다. 그리고 가중화 빈도수가 최소 빈도수보다 같거나 높은 패턴들이 빈발 패턴이 되고 더 이상 빈발 패턴이 발생되지 않을 때까지 패턴을 확장한다. WIS는 조합된 후보 패턴들의 빈도수를 계산하기 위해 데이터베이스를 스캔하고 확인해야 한다. 그래서 대용량 데이터베이스를 분석하기 어렵다. 그래서 후보 패턴이 조합될 때마다 데이터베이스를 스캔하지 않도록 하기 위해 특별한 자료구조와 마이닝 방법이 제안되었다.

3.2 WIT-FWIs 알고리즘

WIT-FWIs는 격자구조 방식으로 패턴을 확장하고 FWI를 마이닝 하는 알고리즘이다. 격자구조의 노드에는 패턴의 정보와 빈도수 그리고 패턴이 포함된 트랜잭션의 ID들의 집합이 저장되어 있다. 패턴을 확장하기 위해서

WIT-FWIs는 합성하기위해 선택한 두 패턴을 이용해 확장된 패턴을 만들고 모두 포함된 트랜잭션 ID에 해당하는 트랜잭션 가중치의 합을 확장된 패턴의 가중화 빈도수로 한다. 만약 가중화 빈도수가 최소 빈도수보다 작다면 이 패턴은 프루닝 되고 이 패턴을 이용해 더 이상 추가적인 패턴을 생성하지 않는다. WIT-FWIs-MODIFY [11] 는 WIT-FWIs에서 두 패턴의 트랜잭션 ID들을 이용해 가중화 빈도수를 계산하는 연산을 줄인 알고리즘이다. 이 방법은 확장된 패턴이 포함된 트랜잭션 ID의 수와 확장하기 위해 선택했던 두 패턴의 트랜잭션 ID 수를 비교하여 두 패턴 중 같은 수의 트랜잭션 ID를 가지는 패턴의 가중화 빈도수가 확장된 패턴의 가중화 빈도수가 되는 방식이다. WIT-FWIs-DIFF [11] 는 확장된 패턴에 해당하는 노드의 트랜잭션 ID를 저장하는 구조에 확장하기 위해 선택된 두 패턴에 대해 배타적 논리합연산 된 트랜잭션 ID들을 저장한다. 그래서 WIT-FWIs-DIFF는 각 노드에 대해 트랜잭션 ID 저장을 줄일 수 있다. 그러나 WIT-FWIs-DIFF는 확장을 위해 선택된 두 패턴이 대부분 다른 트랜잭션 ID를 가지고 있다면 노드에 저장되는 트랜잭션 ID의 수가 매우 많아 질 수 있다. 그래서 세 알고리즘을 비교하면 WIT-FWIs와 WIT-FWIs-MODIFY는 트랜잭션들이 서로 비슷하지 않은 데이터베이스를 마이닝하는 경우 효율적이며 WIT-FWIs-DIFF는 트랜잭션의 아이템들이 서로 비슷한 데이터베이스를 마이닝 할 경우 효율적이다.

4. 트랜잭션 가중치 기반의 마이닝 알고리즘의 성능평가

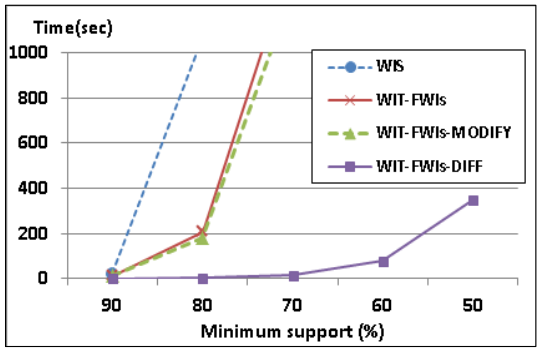
본 절에서는 3장에서 소개한 알고리즘의 성능을 평가하고 각 알고리즘의 성능에 대하여 평가한다. 성능평가에 사용된 데이터는 FIMI(http://fimi.ua.ac.be/)에서 제공하는 데이터 셋인 Connect와 Retail을 이용한다. Connect데이터는 네트워크 온라인 로그 데이터이고 Retail 데이터는 벨기에의 익명의 소매점에서 판매한 상품 판매 정보 데이터이다. Connect 와 Retail 데이터는 표 1과 같이 서로 다른 각각의 특징을 가진다. 표 1에서 Connect 데이터는 아이템의 수가 적고 평균 트랜잭션의 길이가 길다. 그래서 각 트랜잭션들의 형태가 거의 유사한 특징을 갖는다. Retail 데이터는 아이템의 수가 많고 평균 트랜잭션 길이가 비교적 작으며, 각 트랜잭션들의 길이나 아이템의 구성이 대부분 다르다. T10I4D1000K부터 T10I4D5000K까지 5개 데이터는 확장성 평가를 위한 가상 데이터이다. 5개의 데이터는 동일한 아이템과 트랜잭션 평균길이를 가지지만 각기 다른 트랜잭션의 수를 가지며 그 수가 점점 증가하는 모습을 보인다.

(표 1) 성능평가를 위한 데이터 정보
(Table 1) Data information for performance evaluation

Dataset	# of Transactions	# of Items	Avg. Trans. size	Data Type
Connect	65536	128	43	Dense
Retail	88162	21387	10.3	Sparse
T10I4D1000K	1000000	1000	10.1	Sparse
T10I4D2000K	2000000	1000	10.1	Sparse
T10I4D3000K	3000000	1000	10.1	Sparse
T10I4D4000K	4000000	1000	10.1	Sparse
T10I4D5000K	5000000	1000	10.1	Sparse

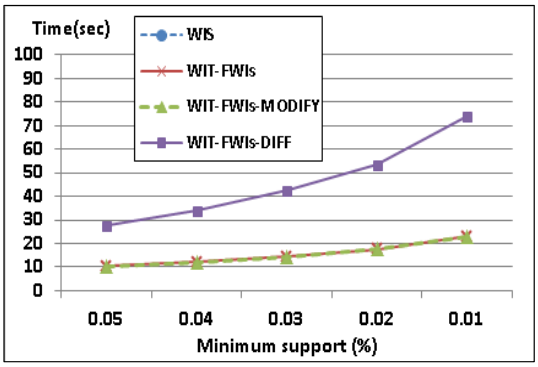
4.1 마이닝 시간 평가

첫 번째 성능평가는 마이닝 시간에 대한 평가이다. 마이닝 시간에 대한 실험 방법은 다음과 같다. WIS, WIT-FWIs, WIT-FWIs-MODIFY, WIT-FWIs-DIFF의 각각 알고리즘에 대해 마이닝을 수행한다. 그리고 Connect 와 Retail 데이터를 이용해 테스트하였으며 아이템의 가중치는 1부터 10까지의 실수 값을 지정하였다. 우리는 Connect 데이터에 대하여 최소 빈도수(Minimum support)를 90%에서 50%까지 변화시키며 테스트하였다.



(그림 1) 마이닝 시간 평가(Connect)
(Figure 1) Mining time test(Connect)

그림 1은 Connect 데이터에서 대한 평가 결과를 보여준다. WIS는 가장 느린 마이닝 결과를 보여주었다. 최소 빈도수가 80%일 때 1074초로 다른 알고리즘에 비하여 많은 마이닝 시간을 필요로 하였다. WIT-FWIs-DIFF는 가장 좋은 성능을 보였다. WIT-FWIs-DIFF는 조합되는 패턴의 서로 다른 트랜잭션 ID를 저장하기 때문에 Dense 특성의 데이터에서 좋은 성능을 보인다.

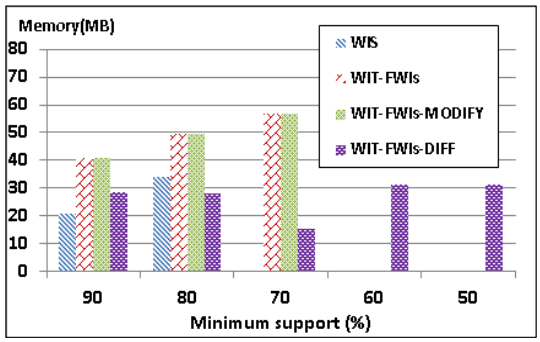


(그림 2) 마이닝 시간 평가(Retail)
(Figure 2) Mining time test(Retail)

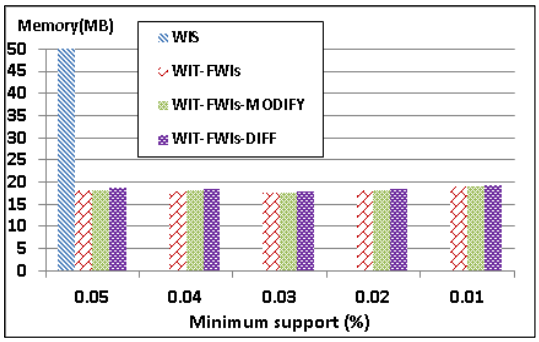
다음은 Retail 데이터를 이용해 마이닝 시간을 평가하였다. 최소 빈도수는 0.05%에서 0.01%까지 변화했고, 마이닝 시간은 알고리즘 수행을 시작하고 종료할 때까지의 시간을 측정하였다. 그림1은 마이닝 시간을 평가한 결과를 보여준다. 평가 결과 WIS의 마이닝 시간은 최소 빈도수가 0.05%일 때 2478초 가장 많은 시간을 필요로 하였다. 그리고 WIT-FWIs-DIFF가 WIS 다음으로 많은 마이닝 시간이 필요하였고 나머지 두 알고리즘이 가장 빠른 마이닝 시간을 보여주었다. 마이닝의 대상 데이터베이스가 커질수록, 임계값이 낮아질수록 필요로 하는 수행시간이 기하급수적으로 증가할 수 있기 때문에, 이러한 환경에서 해당 알고리즘이 얼마나 빠른 속도로 마이닝 과정을 수행할 수 있는지가 중요한 성능 척도로써 고려된다.

4.2 최대 메모리 사용량 평가

다음은 메모리 사용량을 평가하였다. 평가 수행방법은 마이닝 시간 평가와 동일하고 마이닝 시간동안 사용된 메모리 중 최대 사용된 메모리의 양을 측정하였다. 최대 메모리 사용량은 알고리즘이 얼마나 적은 양의 메모리 자원을 가지고 동일한 마이닝 과정을 수행할 수 있는가를 평가하는 척도로써 그 의미가 있다. 그림 3은 Connect 데이터를 이용한 마이닝 과정 중 사용한 최대 메모리 사용량을 보여준다. 평가 결과 WIT-FWIs와 WIT-FWIs-MODIFY는 다른 알고리즘에 비해 많은 메모리 사용량을 보였다. 두 알고리즘은 각 패턴마다 포함된 트랜잭션 ID를 가지고 있기 때문에 dense 특성의 데이터에서는 패턴마다 대량의 트랜잭션 ID를 저장하게 되고 이것은 메모리 사용량이 증가하는 원인이 된다. 그림 4는 메모리 사용량 평가에 대한 결과이다. WIS는 최소 빈도수가 0.05%일 때 529MB의 메모리를 사용하였다. 나머지 세 알고리즘은 15~20MB 내외의 메모리를 서로 비슷하게 사용하였다.



(그림 3) 메모리 사용량 평가(Connect)
(Figure 3) Memory usage test(Connect)

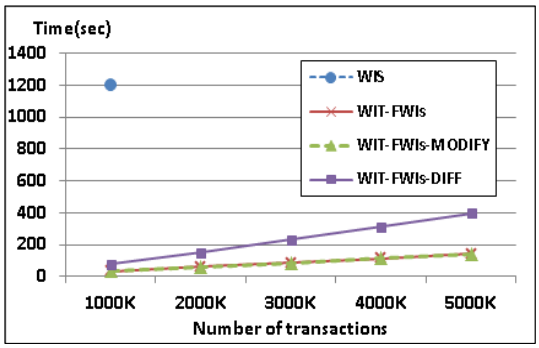


(그림 4) 메모리 사용량 평가(Connect)
(Figure 4) Memory usage test(Connect)

4.3 확장성 평가

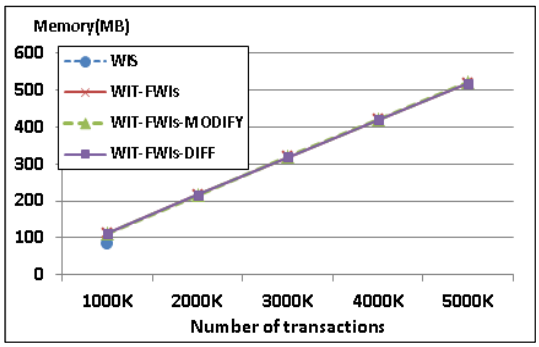
우리는 알고리즘의 확장성을 평가하기 위하여 표1의 데이터 중에서 T10I4D1000K부터 T10I4D5000K까지 5개

의 데이터를 이용해 마이닝 시간과 최대 메모리 사용량을 평가하였다. 확장성 평가는 점점 증가하는 크기의 데이터베이스에 대해 해당 알고리즘이 마이닝 성능을 얼마나 효율적으로 유지할 수 있는가를 평가하는 척도이다. 그림 5는 마이닝 시간에 대한 평가 결과를 보여준다. WIS를 제외한 모든 알고리즘은 트랜잭션이 증가해도 안정적으로 마이닝이 가능하였다. 하지만 WIS는 가장 많은 마이닝 시간을 필요로 하였으며 100만개의 트랜잭션보다 더 많은 트랜잭션을 가진 데이터 셋에서 메모리 사용량 초과로 인해 더 이상 측정 할 수 없었다. 그리고 WIT-FWIs-DIFF는 Sparse한 데이터의 특징에 따라 다른 두 알고리즘에 비해 많은 마이닝 시간을 필요로 하였다.



(그림 5) 확장성 평가(Runtime)
(Figure 5) Scalability test(Connect)

그림 6은 트랜잭션이 증가하면서 메모리 사용량을 측정한 결과를 보여준다. 평가 결과 WIS를 제외한 나머지 알고리즘은 거의 비슷한 메모리 사용량을 보여주었으며 그림 4의 메모리 사용량과 같은 결과를 보여주었다.



(그림 6) 확장성 평가(Memory)
(Figure 6) Scalability test(Memory)

5. 결 론

본 논문에서는 FWI의 정의와 활용 방법 그리고 FWI 기반의 알고리즘에 대하여 분석하고 성능을 평가했다. FWI는 수익과 같은 추가적인 정보를 반영하여 마이닝 하는 방법으로 트랜잭션에 가중치를 계산하고 이를 기반으로 빈발 패턴을 찾는다. 그래서 다른 방식의 마이닝 알고리즘에 비해 트랜잭션의 구조에 영향을 받기 때문에 트렌젝션의 가중치 특성을 고려한 패턴들을 추출할 수 있다. 성능평가 결과로 WIS는 Apriori 기반의 알고리즘으로 후보 패턴 생성 과정을 포함하고 데이터베이스를 다수 스캔하는 단점을 가지고 있기 때문에 매우 많은 마이닝 시간과 메모리를 필요로 했다. WIT-FWIs는 Lattice를 기반으로 후보 패턴을 생성하지 않고 트랜잭션과 아이템셋의 성질을 이용해 마이닝 하므로 WIS에 비해 효율적인 속도와 메모리 사용량을 보였다. 패턴 마이닝 분야에서는 주어진 데이터베이스의 크기가 커질수록, 임계값의 수준이 낮아질수록 기하급수적인 수행시간과 메모리 소모량을 필요로 할 수 있기 때문에, 상황에 따라 마이닝 작업 자체가 실패하는 경우가 존재하므로, 알고리즘의 수행시간과 메모리 효율성은 이러한 점에서 큰 의미를 갖는다. 현재 빈발 패턴 마이닝의 중요한 이슈 중 하나는 빅 데이터를 효과적으로 처리 할 수 있는 알고리즘을 연구 및 개발하는 것이다. 우리는 향후 연구로서, 빅 데이터를 처리하기 위해 WIT-FWIs의 방식보다 더 효율적인 자료구조와 마이닝 방법을 개선함과 동시에, 스트림 처리 [4]가 가능한 알고리즘에 대한 연구를 진행할 것이다.

참 고 문 헌 (Reference)

[1] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules", Proceedings of 20th International Conference on Very Large Data Bases, pp. 487-499, 1994. http://dl.acm.org/citation.cfm?id=672836

[2] L. Cagliero and P. Garza, "Infrequent Weighted Itemset Mining Using Frequent Pattern Growth", IEEE Transactions on Knowledge and Data Engineering, Vol. 26, no. 4, pp. 903-915, 2014. http://dx.doi.org/10.1109/TKDE.2013.69

[3] 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, no. 1, pp. 53-87, 2004. http://dl.acm.org/citation.cfm?id=954525

[4] Y. Kim, W. Kim, and U. Kim, "Mining Frequent Itemsets with Normalized Weight in Continuous Data Streams", The Journal of Information Processing Systems, Vol. 6, no. 1, pp. 79-90, 2010. http://65.54.113.26/Publication/13268251/mining-frequent-itemsets-with-normalized-weight-in-continuous-data-streams

[5] Y. Lee and S. Park, "Optimal Moving Pattern Mining using Frequency of Sequence and Weights", Journal of Korean Society for Internet Information, Vol. 10, no. 5, pp. 79-94, 2009. http://www.koreascience.or.kr/article/ArticleFullRecord.jsp?cn=OTJBCD_2009_v10n5_79

[6] C. Lin, T. Hong, G. Lan, J. Wong, W. Lin, "Incrementally mining high utility patterns based on pre-large concept", Applied Intelligence, Vol. 40, no. 2, pp. 343-357, 2014. http://dl.acm.org/citation.cfm?id=2584602

[7] H. Min, J. Park, D. Lee, and I. Kim, "Outlier Detection Method for Mobile Banking with User Input Pattern and E-finance Transaction Pattern", Journal of Korean Society for Internet Information, Vol. 15, no. 1, 157-170, 2014. http://www.researchgate.net/publication/264171355_Outlier_Detection_Method_for_Mobile_Banking_with_User_Input_Pattern_and_E-finance_Transaction_Pattern

[8] G.D. Ramkumar, S. Ranka, and S. Tsur, "Weighted Association Rules: Model and Algorithm", Proceedings of 4th ACM International Conference on Knowledge Discovery and Data Mining, pp. 661-666, 1998. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.22.9320

[9] H. Ryang and U. Yun, "Performance Analysis of Frequent Pattern Mining with Multiple Minimum Supports", Journal of Korean Society for Internet Information, Vol. 14, no. 6, 1-8, 2013. http://dx.doi.org/10.7472/jksii.2013.14.6.01

[10] M. Shin and W. Paik, "Design and Implementation of Sequential Pattern Miner to Analyze Alert Data Pattern", Journal of Korean Society for Internet Information, Vol. 10, no. 2, pp. 1-13, 2009. http://ocean.kisti.re.kr/IS_mvpopo001P.do?method=multMain&poid=ksii1&free=

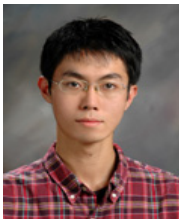
[11] B. Vo, F. Coenen, and B. Le, "A new method for mining Frequent Weighted Itemsets based on WIT-trees", Expert system with applications, Vol. 40, pp. 1256-1264, 2013. http://dl.acm.org/citation.cfm?id=2400944

[12] U. Yun, "On pushing weight constraints deeply into frequent itemset mining", Intelligent Data Analysis, Vol. 13, no. 2, pp. 359-383, 2009. http://iospress.metapress.com/content/b1720248602407ut/

[13] S. Zhang, P. Guo, Jifu Z., X. Wang, and W. Pedrycz, "A completeness analysis of frequent weighted concept lattices and their algebraic properties", Data and Knowledge Engineering, Vols. 81-82, pp. 104-117, 2012. http://www.sciencedirect.com/science/article/pii/S016902 3X12000833

◐ 저 자 소 개 ◑

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



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