J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr

점진적 가중화 맥시멀 대표 패턴 마이닝의 최신 기법 분석, 유아들의 물품 패턴 분석 시나리오 및 성능 분석☆

Recent Technique Analysis, Infant Commodity Pattern Analysis Scenario and Performance Analysis of Incremental Weighted Maximal Representative Pattern Mining

윤 은 일[1] 윤 은 미[2*]
Unil Yun Eunmi Yun

요 약

데이터마이닝 기법들은 의미 있고 유용한 정보를 효율적으로 찾기 위해서 제안되어 왔다. 특별히, 빅 데이터 환경에서 데이터가여러 응용들에서 축적되어짐에 따라, 관련된 패턴 마이닝 방법들이 제안되고 있다. 최근에는 파일이나 데이터베이스에 이미 저장되어 있는 정적 데이터를 분석하는 대신에 점진적으로 생성되는 동적 데이터를 마이닝 하는 것이 더 흥미 있는 연구영역으로 고려되고 있는데 동적데이터는 단지 한번만 스캔하여 읽을 수 있기 때문이다. 이와 같은 이유로, 어떻게 동적 데이터를 효율적으로 마이닝 하는지에 대한 연구들이 진행되고 있다. 더불어서, 마이닝 결과로 거대한 수의 패턴들이 생성되기 때문에, 맥시멀 패턴 마이닝과 같은 대표 패턴들을 마이닝하는 접근방법들도 제안되고 있다. 또 다른 이슈로, 실세계에서 더 의미있는 패턴들을 발견하기 위해, 가중화 패턴 마이닝에서 아이템들의 가중치가 사용되고 있다. 실제 상황에서 아이템의 이익이나 가격 등이 가중치로 사용 될 수 있다. 본 논문에서는 점진적으로 생성되는 데이터에 대한 가중화 맥시멀 패턴 마이닝, 맥시멀 대표 패턴 마이닝 그리고 점진적 패턴 마이닝 기법들에 대해 분석한다. 그리고 가중화 대표 패턴 마이닝을 적용하여서 유아들에게서 필요로 하는 물품 패턴들을 분석하기 위한 응용 시나리오를 제시한다. 추가로, 분석한 마이닝 알고리즘들에 대한 성능 평가를 수행한다. 결과적으로, 점진적 가중화 맥시멀 패턴 마이닝 기법이 점진적 가중화 패턴 마이닝과 가중화 패턴 마이닝 기법보다 좋은 성능을 가짐을 보인다.

☞ 주제어 : 가중화 맥시멀 패턴 마이닝, 점진적 마이닝, 대표 패턴, 응용 시나리오, 성능 평가

ABSTRACT

Data mining techniques have been suggested to find efficiently meaningful and useful information. Especially, in the big data environments, as data becomes accumulated in several applications, related pattern mining methods have been proposed. Recently, instead of analyzing not only static data stored already in files or databases, mining dynamic data incrementally generated in a real time is considered as more interesting research areas because these dynamic data can be only one time read. With this reason, researches of how these dynamic data are mined efficiently have been studied. Moreover, approaches of mining representative patterns such as maximal pattern mining have been proposed since a huge number of result patterns as mining results are generated. As another issue, to discover more meaningful patterns in real world, weights of items in weighted pattern mining have been used, In real situation, profits, costs, and so on of items can be utilized as weights. In this paper, we analyzed weighted maximal pattern mining approaches for data generated incrementally. Maximal representative pattern mining techniques, and incremental pattern mining methods. And then, the application scenarios for analyzing the required commodity patterns in infants are presented by applying weighting representative pattern mining. Furthermore, the performance of state-of-the-art algorithms have been evaluated. As a result, we show that incremental weighted maximal pattern mining technique has better performance than incremental weighted pattern mining and weighted maximal pattern mining.

☞ keyword : Weighted maximal pattern mining, Incremental mining, Representative pattern, Application scenario, Performance evaluation,

1 Dept. of Computer Engineering, Sejong University, Seoul, 143-747, Korea
2 Dept. of Infantile Education, Baekseok Art University, Seoul, Korea
* Corresponding author (lovemind114@bau.ac.kr)
[Received 29 October 2019, Reviewed 11 November 2019(R2 30 December 2019), Accepted 22 January 2020]
☆ 본 연구는 2018년도 정부 교육과학기술부의 재원으로 한국 연구재단(NRF)의 지원을 받아 수행된 연구 사업 (NRF No. 2018R1D1A1A09083109)의 연구수행으로 인한 결과물임을 밝힙니다.
☆ 본 논문은 2019년도 한국인터넷정보학회 춘계학술발표대회 우수 논문 추천에 따라 확장 및 수정된 논문임.

1. 서 론

데이터 마이닝에서 가장 활발한 연구가 진행되는 패턴 마이닝[1]은 마켓 데이터, 메디컬 데이터, 트래픽 데이터, 웹 클릭 데이터 및 네트워크 데이터 분석과 같은 다양한 분야들에 활용되어 왔다[1, 5, 9]. 다양한 연구영역으로써 패턴 마이닝에서 가중화 패턴 추출을 위한 방법[2, 3, 4, 10, 12, 13], 전체 결과 패턴 중 대표 패턴의 마이닝 방법 [1, 5, 6, 8, 11, 14], 그리고 이미 저장되어 있는 데이터 뿐만 아니라 실시간으로 생성되는 데이터들에 대한 스트림 마이닝 방법 [6, 15, 16, 17]등 다양한 연구가 진행되고 있다. 본 논문에서는 점진적으로 가중화나 맥시멀 대표 패턴을 마이닝 하기 위해 필요한 기법들[3, 7, 13, 16]을 살펴보고 이 방법을 통합하여 점진적으로 가중화 맥시멀 대표 패턴을 마이닝 하는 기법 및 이에 대한 알고리즘을 분석하고, 가중화 대표패턴 마이닝을 적용하여 유아들의 필요 물품 패턴을 분석하기 위한 응용 시나리오에 대해 제시하며 데이터 크기가 늘어나는 환경에서 성능 평가를 진행한다.

2. 점진적으로 맥시멀 대표 패턴을 추출하기 위해 필요한 기법 분석

2.1. 가중화 빈발 패턴 마이닝 (Weighted frequent pattern mining)

전통적인 빈발 패턴 마이닝[1, 5, 9]에서 하나의 패턴, P = {i_1, i_2, i_3, …, ir}는 하나 또는 그 이상의 유일한 아이템들로 구성한다. 또한, 어느 한 패턴, P와 사용자 특정 최소 지지도 임계값(Minimum support threshold), min_sup가 주어지면, P는 Sup(P) ≥ min_sup이면, 빈발 패턴[1, 5]이라 정의한다. 전통적인 패턴 마이닝과 다르게 실세계에서 아이템들의 실제적인 가중치를 반영하기 위해 가중화 빈발 패턴 마이닝 접근방법들[2, 4, 10, 12]이 제안되었다. 하나의 가중화 패턴은 이것의 가중화 지지도(Weighted support)를 가지며, 이 값은 이것의 서포트(support)와 평균 가중치(weight)의 곱으로 계산된다. 평균 가중치는 이 패턴을 구성하는 모든 아이템의 평균 값이다. 이 가중화 지지도가 주어진 최소 임계값보다 크거나 같다면 우리는 이 패턴을 가중화 빈발 패턴으로 정의[6]한다. 대부분의 가중화 빈발 패턴 마이닝[2, 3, 4, 10, 12, 13]에서, 아이템들의 특성에 따라 각기 다른 가중치 범위를 가지고 있기 때문에 아이템들의 실제 가중치 대신 정규화 된 값을 사용한다. 가중화 빈발 패턴 마이닝의 대표적인 알고리즘으로, WFIM[12]는 원래 데이터베이스를 트리 자료구조에 두 번만 스캔해서 마이닝하는 가중화 빈발 패턴 마이닝 기법이며 프루닝을 효과적으로 하고 엔티-모노톰 법칙(Anti-monotone property)을 사용[1, 5]하기 위해 트리의 정렬을 가중치 오름차순으로 정렬하여 성능을 높였다.

2.2. 맥시멀 대표 패턴 마이닝 (Maximal frequent itemset mining)

마이닝 해야 할 데이터가 커짐에 따라 여기서 생성되는 결과 패턴의 수도 많아지게 되어 전통적인 빈발 패턴 마이닝 알고리즘들처럼 전체 결과 패턴을 추출하는 대신 맥시멀 패턴 마이닝[1]과 같이, 대표 패턴을 마이닝하는 기법[1, 5, 6, 8, 7, 11, 14]이 연구 되고 있다. 어느 한 빈발 패턴이 주어졌을 때, 이것의 어떠한 상위 집합도 빈발하지 않는다면, 이것은 맥시멀 빈발 패턴[1]으로 정의된다. 모든 빈발 패턴들을 마이닝하는 것과 비교해, 맥시멀 빈발 패턴들만을 마이닝 하는 것은 더 적은 수의 패턴 생성을 이끌어 낸다. 뿐만 아니라 이들은 어떤 패턴이 빈발한지 아닌지를 판단하는 경계값으로 사용될 수도 있다. 그 예로, MAFIA[3]는 맥시멀 패턴 마이닝 알고리즘이고 MWFIM[13]은 가중화 맥시멀 패턴 마이닝 알고리즘이다. 이와 같이 가중화 패턴들을 대표하는 패턴들만을 추출함으로써, 더욱 적은 수의 유용한 마이닝 결과를 제공한다.

2.3. 점진적 패턴 마이닝 (Incremental pattern mining)

실시간 패턴 마이닝[6, 15, 16]은 스트림 패턴 마이닝이라고도 하는데, 데이터가 계속적으로 추가되는 동적인 환경에서 마이닝하는 방법[3, 16]을 말한다. 기존의 전통적인 방식들[1, 5]은 오로지 정적인 데이터베이스들에 초점을 맞추기 때문에 마이닝을 진행하기 위해 데이터베이스를 여러번 스캔을 해야 해서 기존의 정적 데이터를 마이닝하는 알고리즘들은 실시간 마이닝에 사용 될 수 없다. 또 다른 이슈는 실시간 마이닝 환경에서는 현재 빈발하지 않은 아이템들이 새로운 트랜잭션 데이터가 입력됨에 따라 빈발한 것으로 변할 수 있다. 하지만 기존의 전통적인 Apriori 알고리즘 [1]이나 트리기반의 FP-growth [5] 알고리즘은 처음 스캔해서 찾은 길이가 1인 빈발하지 않은 아이템들을 제거하여 스트림 마이닝의 특성을 반영할 수

없다. 실시간 마이닝 기법에서는 데이터베이스를 한번만 스캔하는 싱글 스캔만으로 마이닝이 진행되며 정적 데이터를 자료 구조에 모두 저장하고 이후에 재구축이라는 과정을 거쳐 효율적으로 마이닝하기 위한 압축된 정렬된 구조를 생성한다.

(표 1) 데이터베이스 예제와 아이템의 가중치의 예
(Table 1) An example of transaction database and weights of items (제한점, s: 0.8)

TID	Transaction	Item	Weight
1	A B D	A	0.95
2	A B C D E F G	B	0.55
3	C D E G H	C	0.85
4	B C F G H	D	0.6
		E	0.7
		F	1.0
		G	0.5
		H	0.75

3. 실시간 가중화 맥시멀 패턴 마이닝 (Incremental Weighted Maximal Pattern Mining)

3.1. 실시간 가중화 맥시멀 패턴 마이닝 기법 및 알고리즘

정의1(평균 가중치 값): 패턴, $P = \{i_1, i_2, i_3, \cdots, i_r\}$가 주어지면, 가중화 빈발 패턴 마이닝에서, 패턴, P의 각 아이템은 각자의 고유 정규화 가중치 값을 갖는다. 우리는 P에 대한 가중치 정보를 $W_P = \{w_1, w_2, w_3, \cdots, w_r\}$로 표시한다. 패턴 P에 대한 가중치는 이것을 구성하는 모든 아이템들의 평균 가중치 값으로 계산된다.

정의2 (가중화 빈발 패턴): P의 가중화 서포트(Weighted Support), WSup(P)은 패턴 P의 빈도수Support(P)에 Weight(P)를 곱한 값으로 계산된다. 패턴 P의 가중화빈도수, WSup(P)와 제한점 (Threshold)를 비교해서 제한점보다 크거나 같으면 패턴 P를 가중화 빈발 패턴 (Weighted Frequent Pattern)이라 한다.

표 1은 트랜잭션 데이터베이스의 예를 보이고 있다 트랜잭션 데이터베이스는 트랜잭션들로 구성되며 각 트랜잭션은 함께 발생한 아이템들로 이루어지고 있다. 예를 들면, 표 1에서 패턴 {A, B, D}는 트랜잭션 1번과 2번에서 발생되므로 빈도수, Support({A, B, D})는 2가 된다. 또한, 패턴 {A, B, D}의 가중치, Weight({A, B, D})는 (0.95 + 0.55 + 0.6) / 3 = 0.7 이고 그 패턴의 가중화 빈도수, WSup({A, B, D})는 패턴 P의 평균 가중치, 0.7와 패턴 P의 빈도수, 2를 곱해서 1.4가 된다. 그러므로 이 패턴 {A, B, D}은 제한점(threshold), s가 0.8이면 가중화 빈도수(1.4)가 제한점(0.8) 이상이므로 패턴 {A, B, D}는 가중화 빈발 패턴이 된다.

정의 3 (맥시멀 빈발 패턴): 빈발 패턴 P가 주어지면, 이것의 어떠한 상위 패턴도 s보다 높거나 같은 지지도를 같지 않을 때 이 패턴은 맥시멀 빈발 패턴으로 정의된다.

예를 들면, 테이블에서 제한점, s가 0.8 일때, 패턴 {A, B, D}는 빈발 패턴이지만맥시멀 대표 패턴은 아니다. 왜냐하면 그 패턴보다 수퍼패턴인 빈발 패턴(예를들면, 패턴 {A, B, C, D, E, F, G}이 있기 때문이다. 다른 예로, 패턴 {A, B, C, D, E, F, G}은 맥시멀 빈발 패턴이며 이 패턴을 포함하는 어떤 빈발한 슈퍼 패턴이 없기 때문이다.

정의 4(가중화 맥시멀 빈발 패턴): 패턴 P가 있을 때, 이것의 어떠한 상위 패턴의 가중화 지지도도 s보다 크거나 같지 않다면, 우리는 이것을 가중화 맥시멀 빈발 패턴(WMFI)으로 정의된다.

예를 들면, 패턴 {A, B, D}는 가중화 빈발 패턴이면서 가중화 맥시멀 빈발 패턴이다. 왜냐하면 이 패턴의 가중화 빈도수가 제한점(0.8)보다 크며, 이 패턴의 어떤 슈퍼 패턴도 가중화 빈발 패턴이 이니기 때문이다. 위 테이블을 근간으로 패턴 {A, B, D}을 포함하는 슈퍼 패턴은 {A, B, C, D}, {A, B, D, E}, {A, B, D, F}, {A, B, D, G}, {A, B, C, D, E}, {A, B, C, D, F}, {A, B, C, D, G}, {A, B, D, E, F}, {A, B, D, E, G}, {A, B, D, F, G}, {A, B, C, D, E, F}, {A, B, C, D, E, G}, {A, B, C, D, F, G}, {A, B, D, E, F, G}, {A, B, C, D, E, F, G} 이며 이들 패턴들의 가중화 빈도수 값들은 모두 제한점보다 작다.

위의 정의에 근간해서 점진적으로 늘어나는 데이터베이스에서 실시간으로 증대되는 데이터베이스를 한번만 스캔해서 점진적으로 가중화 맥시멀 대표 패턴을 마이닝 [9]은 다음의 순서로 마이닝 과정이 수행된다.

① 주어진 데이터베이스를 점진적으로 읽어가며 트랜잭션들을 사전순서에 따라 삽입하고 이를 삽입후에는, 현재 전역 트리의 현재 아이템 서포트 감소 순서로 정렬하여 갱신하여 변경된 정렬 순서에 따라 전역트리가 재구축된다.

② 유효한 가중화 맥시멀 대표 패턴을 찾기 위한 패턴 성장(growth) 과정이 수행된다. 각 트랜잭션은 위의 순서에 따라 정렬되며, 전역트리에 순차적으로 삽입된다. 트리의 재구축 과정은 동적으로 데이터베이스가 추가로 생성되는 각 패스에 대해 수행하며 노드 링크들이 재정렬된 트리에 적합하도록 업데

이트 된다.

③ 트리의 재구축 과정 후에는, 재귀적으로 마이닝이 진행되며. 트리의 테이블의 각 아이템에 대해 성장 과정으로 마이닝 효율성을 높이기 위해 분할 및 정복 (divide and conquer) 방식으로 bottom-up 방식으로 진행된다.

④ 알고리즘은 트리의 가장 단말 로드의 한 아이템을 프리픽스(prefix)를 선택하고, 과추정 가중화 서포트 (Overestimate weighted support) 값을 구하기 위해 나올 수 있는 최대 가중화값 (Maximum Weight)을 계산한다. 만약 이 값이 주어진 최소 지지도 임계값, min_sup보다 작다면, 해당하는 성장 과정은 수행되지 않는다.

⑤ 현재 트리, T가 싱글패스를 가지고 있다면, 알고리즘은 프리픽스와 T의 아이템들을 합친 후보 패턴을 찾고 이 후보패턴을 맥시멀 특성을 만족하는지 확인하기 위해 지금껏 찾은 가중화 맥시멀 패턴들과의 서브셋 여부를 확인하고 서브셋이 아니라면, 이것의 실제 가중화 서포트(Weighted support)가 임계값보다 크거나 같은 가중화 맥시멀 대표패턴을 찾고 이 패턴을 지금까지 발견한 가중화 맥시멀 대표패턴들의 트리 저장소에 저장한다.

⑥ 트리 T가 멀티플 패스들을 가지고 있다면, 현재 프리픽스에 대한 조건부 트리를 생성하고 마이닝이 재귀적으로 진행된다. 이 반복적 과정을 통해, 가중화 패턴 후보가 찾아지면 서브셋 체킹을 하여 맥시멀 조건이 만족하는지를 확인하여 가중화 대표 맥시멀 패턴을 찾아낸다. 효율적인 마이닝 과정을 위해, 가장 긴 길이의 가중화 맥시멀 대표패턴을 먼저 찾아내어 서브셋 모듈의 효율성을 높인다.

⑦ 전역 트리의 헤더테이블의 모든 아이템들에 대해 이러한 연산들을 수행한 결과로 가중화 맥시멀 대표 패턴을 찾으며 위의 과정은 DB에서 점진적으로 들어오는 트랜잭션들을 가진 추가 데이터베이스 DB+에 대해서 반복적으로 수행한다. 이 과정을 근간으로 그림 1에서는 데이터베이스에서 인크리멘탈 가중화 맥시멀 패턴 마이닝 알고리즘 [16]을 보이고 있다.

```
Input: an incremental database, DB;
       an incremental product database, DB+;
       item weight information, W;
       a given minimum support threshold, s.
Output: a set of WMFIs, R.

Main_procedure
01. for each item, i in T.table //bottom-up order
02.   add i to pref;
03.     if (T is the global tree)
04.       calculate MaxW for i;
05.     if (Owsup(pref) < s) //pre-pruning
06.       delete pref.last;
07.       go to line 01 and continue;
08.     else
09.       if (T has a single path)
10.         candidate WMFI, CWMFI ß pref È T;
11.         if (check_subset(CWMFI, SC) is true);
12.           delete pref.last;
13.           go to line 01 and continue;
14.         else
15.           if (Wsup(CWMFI) ³ s)
16.             insert CWMFI into R;
17.             update SC with CWMFI;
18.           else
19.             call extract_sub_WMFIs(CWMFI, SC);
20.             delete pref.last;
21.             go to line 01 and continue;
22.           else //T has multiple paths
23.           construct conditional IM_WMFI-tree for
                  pref, T';
24.             call IM_WMFI_growth(T', SC, pref);
                  //recursive call
25.             delete pref.last;
```

(그림 1) IM_WMFI 알고리즘
(Figure 1) IM_WMFI algorithm

3.2. 유아들의 필요 물품 패턴을 분석하기 위한 가중화 대표패턴 적용 시나리오

본 절에서는 가중화 대표 패턴 마이닝을 유아들에게 필요 물품을 구매하는 트랜잭션 데이터베이스에서 의미있는 패턴들을 추출하는 적용 시나리오에 대해 제시한다. 본 논문에서 성능평가를 진행한 실제(real) 데이터 셋의 예도 물품(retail) 데이터로써 구매 데이터셋에 대해서 마이닝 기법을 실행을 통해서 의미 있는 패턴을 추출 할 수 있다. 본 응용 시나리오는 유아들이 필요로 하는 물품들 중에서 중요 물품들을 정의하고 이에 대한 가중치를 적용하고 이에 대해 리테일 (물품) 데이터베이스에서 의미 있는 대표 패턴을 찾는 시나리오를 생각해 볼 수 있다. 유아들이 사용하며 필요로 하는 물품에는 유아교육용품, 유아교육 책, 유아 의류, 유아 문구, 유아용 전자기기, 유아 지능 향상을 위한 장난감, 놀이매트, 기저기, 젖병, 유아컵, 물티슈, 어린이 침구, 유모차, 수유용품등 매우 다양한 많은 아이템들이 있고 이들 용품들은 ,인터넷상에서 다양한 업체에서 판매하고 있다. 먼저 인터넷에서 판매하는 이들 물품들은 가격(price)과 이득(profit)이 있고

(표 2) 데이터 셋의 특성
(Table 2) Characteristics of datasets

Dataset	Size	# of Trans.	# of Items	Avg. Trans. Size
Chess	0.326MB	3,196	75	37
Retail	3.97MB	88,162	16,470	10.306
T10I4D100-1000K	3.83-38.3MB	100k-1,000k	1,000	10

이는 1000원 정도부터 비싸게는 100만원이 넘는 유아용 컴퓨터등의 물품들이 있다. 이들 가격을 근간으로 물품들의 가중치의 근간인 가중치를 선정하고 이를 정규화를 통해 일정 범위의 가중치 범위(weight range)로 설정 할 수 있다. 이후, 일정 제한점을 근간으로 가중치와 판매 갯수의 곱으로 가중화 빈도수(weighed support)를 계산할 수 있고 물품들의 이 가중화 빈도수 값들을 이용하여 가중화 빈발 패턴(weighted frequent pattern) 과 이들 중 이 패턴들을 대표 할 수 있는 가중화 대표 패턴을 추출할 수 있으며 이 패턴들은 전체유아 물품 데이터베이스에서 가격을 고려하는 중요 물품들 중 대표하는 패턴을 마이닝하여 유아들을 위해 가격이 고려된 대표 구매 패턴들의 추이를 분석 하는데 사용될수 있다. 또한, 추출된 대표 패턴에서 같은 패턴 내에 포함되어 있는 아이템들은 그 아이템들 간에 연관도가 있는 것도 확인 할 수 있다.

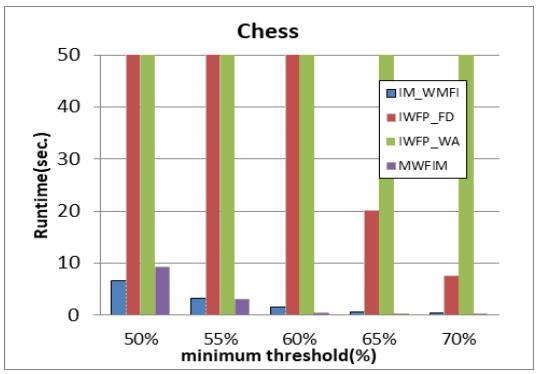
4. 성능평가 및 분석

4.1. 테스트 환경

점진적, 가중화 그리고 맥시멀 대표패턴 기법을 적용한 마이닝 알고리즘의 성능 분석을 위해 IM_WMFI 알고리즘 [16], $IWFP_{FD}$ [3], $IWFP_{WA}$ [3], 그리고 MWFIM 알고리즘 [13]에 대한 성능 분석[16]한다. 테스트 환경은 Intel Core, 3.33GHz, 3.00GB RAM, Windows 7 OS이며, 모든 알고리즘은 C++언어로 구현되었다. 실험을 위해 패턴 마이닝 분야에서 널리 쓰이는 FIMI(Frequent Itemset MIning) 레파지토리 (http://fimi.ua.ac.be/data/)에서 Chess, Retail 데이터셋을 이용한다. 또한 점점 증가하는 확장성에 대한 성능 테스트를 위해 IBM 데이터셋 생성기를 통해 (http://www.almaden.ibm.com/software/projects/hdb/resources.shtml) 점점 트랜잭션 수가 증가하는 가상 데이터셋 그룹과 아이템들이 변화하는 가상 데이터셋들을 생성하였으며 가중화를 위한 가중치는 0.5-1.0 사이 값으로 가중치 범위를 설정하여 성능분석에 이용해 알고리즘들의 성능평가를 진행하였다.

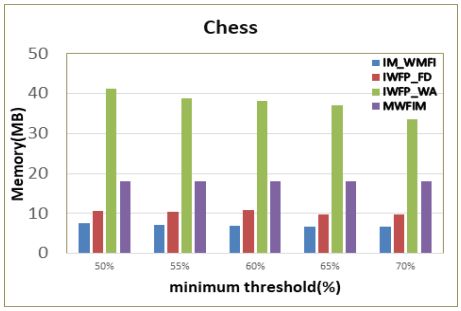
4.2. 제한점(threshold) 변화에 따른 실행시간과 메모리 성능 평가

본 절에서는 실 데이터 Chess 와 Retail 데이터셋에 대해 실행시간과 메모리 사용량을 테스트하고 그 결과를 분석한다. 그림 2, 3, 4, 5는 두 데이터셋의 결과를 보이고 있으며 x축에서 제한 수(minimum threshold)를 늘림에 따라 테스트한 알고리즘들 모두 실행시간이 줄어들어 있으며 특히 IM_WMFI 알고리즘이 실행시간을 적게 쓰고 있음을 알 수 있으며 이는 효율적인 알고리즘 작동과 대상이 되지 않는 패턴들을 효율적으로 푸루닝(pruning)하기 때문이다. 메모리 테스트에서 알고리즘이 수행되는 동안 최대 메모리 사용량을 보이고 있으며 이 결과도 대상이 되지 않는 패턴들을 효율적으로 제거하면서 알고리즘이 불필요한 데이터를 제거하기 메모리를 적게 사용한 결과를 볼 수 있다.

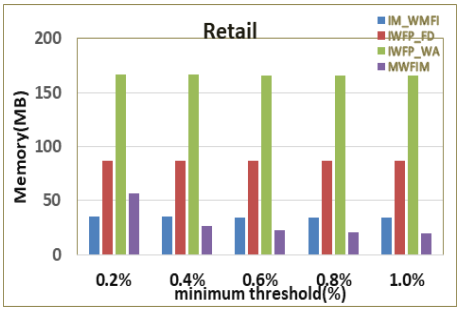


(그림 2) 체스 데이터셋에서 제한점에 따른 수행시간
(Figure 2) Runtime according to thresholds in Chess dataset

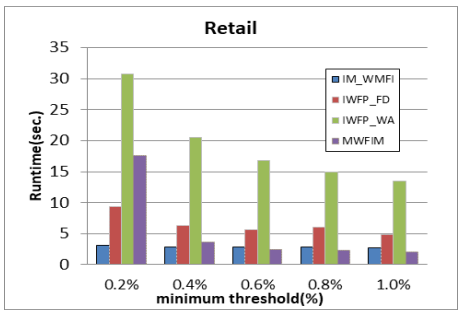
전반적으로 IM_WMFI 과 MWFIM 알고리즘이 성능이 좋으며 이는 두 알고리즘은 전체 패턴들을 추출하는 것이 아니라 대표패턴만을 마이닝하기에 더 효율성이 커지



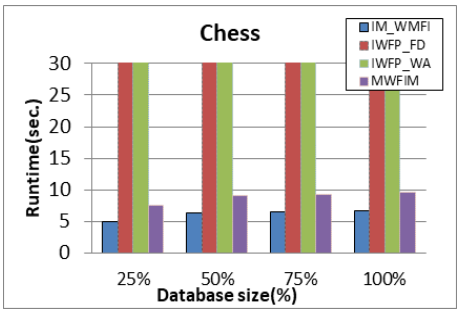
(그림 3) 체스 데이터셋에서 제한점에 따른 메모리 사용량
(Figure 3) Memory usage according to thresholds in Chess dataset



(그림 5) 리테일 데이터셋에서 제한점에 따른 메모리 사용량
(Figure 5) Memory usage according to thresholds in Retail dataset



(그림 4) 리테일 데이터셋에서 제한점에 따른 수행시간
(Figure 4) Runtime according to thresholds in Retail dataset

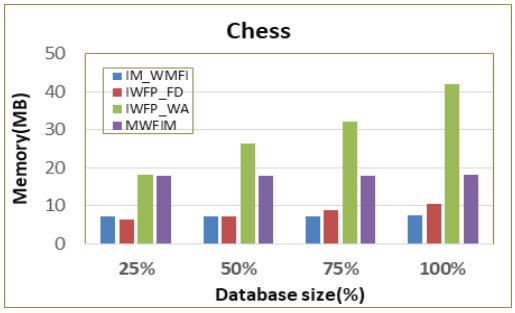


(그림 6) 체스 데이터셋에서 데이터베이스 크기에 따른 수행시간
(Figure 6) Runtime according to database size in Chess dataset

며 이 두 알고리즘 중에서도 IM_WMFI 알고리즘이 성능이 더 좋은 것을 볼 수 있는데 이는 IM_WMFI 알고리즘이 MWFIM 알고리즘보다 점진적처리를 위해 효율적인 진행을 하기 때문이다. MWFIM 알고리즘은 정적인 기법이기 때문에 새로운 데이터가 점진적으로 늘어날 때마다 마이닝 작업을 처음부터 다시 시작 해야만 하므로 IM_WMFI 알고리즘 보다 성능이 떨어지게 된다.

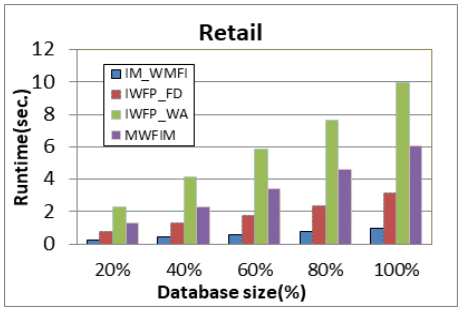
4.3. 데이터베이스 증가에 따른 실행시간과 메모리 성능평가

4.3 절에서는 데이터베이스 크기가 점진적으로 커짐에 따라 4개의 알고리즘들의 실행시간과 메모리 사용량을 테스트한 결과를 보이고 있다. 그림 6, 7, 8, 9는 4개 알고리즘들을 Chess와 Retail 데이터셋을 적용하여 테스트하였다.



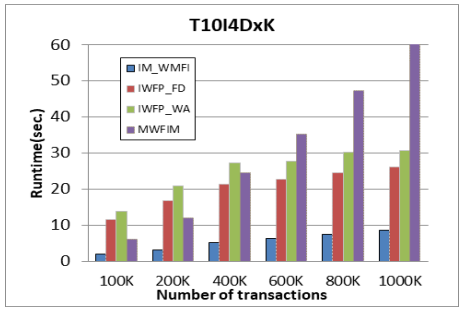
(그림 7) 체스 데이터셋에서 데이터베이스 크기에 따른 메모리 사용량
(Figure 7) Memory usage according to database size in Chess dataset

이 테스트에서 두 데이터셋에 적용하는 제한 수(minimum threshold)는 50%와 0.2%로 세팅하였으며 x축



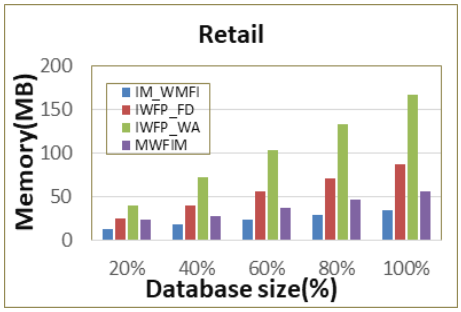
(그림 8) 리테일 데이터셋에서 데이터베이스 크기에 따른 수행시간

(Figure 8) Runtime according to database size in Retail dataset



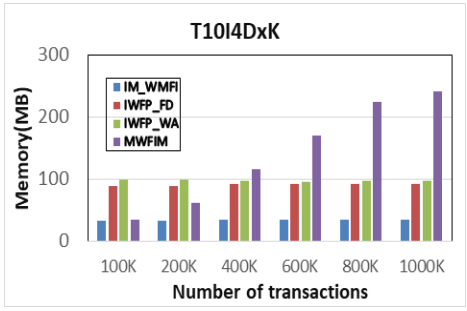
(그림 10) 트랜잭션 수에 대한 실행시간 확장성

(Figure 10) Runtime Scalability of number of transactions



(그림 9) 리테일 데이터셋에서 데이터베이스 크기에 따른 메모리 사용량

(Figure 9) Memory usage according to database size in Retail dataset



(그림 11) 트랜잭션 수에 대한 메모리 확장성

(Figure 11) Memory Scalability of number of transactions

은 데이터베이스가 점진적으로 25%씩 증가하는 것으로 하여 테스트하고 이에 대한 결과를 보이고 있다. 데이터베이스 크기가 증대됨에 따라 IWFPFD, IWFPWA 알고리즘은 IM_WMFI 과 MWFIM 알고리즘보다 실행시간을 많이 걸리고 메모리도 많은 량을 사용한다.

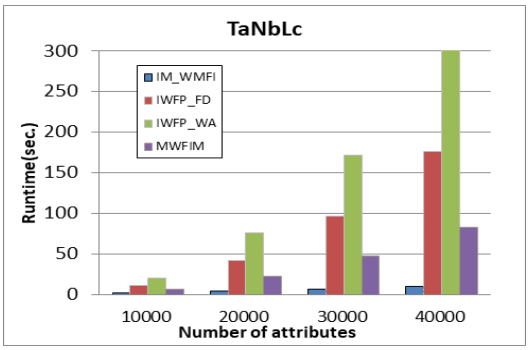
IWFPFD, 와 IWFPWA 알고리즘들은 점진적 기법을 사용하고 있지만 모등 패턴들을 추출하며 MWFIM 알고리즘은 가중화 맥시멀 패턴들을 추출하지만 정적인 방법을 사용하여 성능이 제한적이다. 반면, IM_WMFI 알고리즘은 점진적인 동적인 기법과 대표패턴을 마이닝 하는 방법을 적용하여 성능이 좋으며 특히, 데이터베이스 크기가 커짐에 따라 성능 효율이 더 커짐을 알 수 있다.

4.4. 트랜잭션과 아이템 수에 대한 확장성 평가

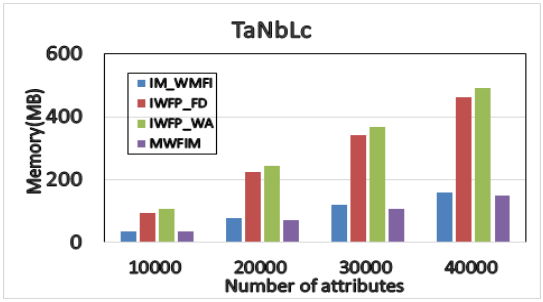
이 절에서는 데이터셋의 크기를 트랜잭션의 수와 트랜잭션 안에 있는 아이템들의 수(에트리뷰트수)를 늘려감에 따라 알고리즘들이 어떻게 성능의 변화를 가져오는지를 확장성 테스트하고 그 결과를 분석하였다. 데이터셋들은 트렌젝션들의 수가 증가하는 것을 특징으로 하는 T10I4D100K - T10I4D1000K 데이터셋(T10I4DxK)과 속성들의 수가 증가하는 것을 특징으로 하는 T10L1000N10000 - T40L4000N40000 (TaLbNc) 데이터셋이며 이들의 최소지지도 임계값과 가중치 범위 세팅은 각각 0.5%와 0.5~0.8로 설정하였다.

그림 10과 11은 데이터셋에서 트랜잭션의 수를 100k에서 1000k까지 늘려가며 트래재션의 수가 많아짐에 따라 알고리즘의 실행시간과 메모리 사용량이 데이터 크기에 비례해서 어느 정도 낮은 기울기의 직선형태(linear)로 늘어나는지 알고리즘의 확장성 정도에 대한 성능평가 결과를 보이고 있다. *IM_WMFI* 알고리즘이 가장 안정적인 성

능을 보이고 있으며 특히, 데이터셋의 크기가 증가함에 따라 실행시간이 가장 적은 시간으로 직선 형태로 실행되며 메모리 사용량도 적게 사용하면서 트랜잭션의 수가 늘어 남에 따라 가장 기울기가 낮은 직선 형태로 늘어남을 알 수 있다. 반면 4.2 절의 제한점(threshold) 변화나 4.3. 데이터베이스 증가에 따라 상대적으로 좋은 성능을 보인 MWFIM 알고리즘은 확장성 테스트 결과가 가장 좋지 않은 결과를 보이고 있다. 그 이유는 MWFIM 알고리즘은 정적 알고리즘으로 성능의 제한을 가지기 때문이다.



(그림 12) 속성 수에 대한 실행시간 확장성
(Figure 12) Runtime Scalability of number of attributes



(그림 13) 속성 수에 대한 실행시간 확장성
(Figure 13) Runtime Scalability of number of attributes

그림 12와 13은 반대로 트랜잭션의 수를 고정시키고 아이템(속성)수를 증가시키는 환경에 대한 실행시간과 메모리 사용량에 대해 비교 알고리즘별로 아이켐수가 커짐에 따라 어느 정도 낮은 기울기의 직선형태(linear)로 늘어나는지 확장성 성능 결과 보이고 있다. IM_WMFI이 실행시간에서는 가장 좋은 성능을 보이고 메모리 사용량에 대해서는 IM_WMFI과 MWFIM 알고리즘이 아이템수의 증가에 따라 안정적인 메모리 확장성을 보이고 있다.

5. 결 론

본 논문에서는 실시간으로 처리되어야 하는 스트림 환경에서 가중화 맥시멀 대표 패턴 마이닝을 하기 위해 필요한 가중화 패턴 마이닝 기법, 맥시멀 대표 패턴 마이닝 기법, 그리고 실시간 패턴 마이닝 기법을 살펴보았고 이 기법들을 결합한 실시간 가중화 맥시멀 대표 패턴 마이닝 기법에 대한 알고리즘의 작동과정을 분석하였고 이에 따른 실시간 가중화 맥시멀 패턴 마이닝 알고리즘을 살펴보았고 이를 근간으로 마이닝 알고리즘들에 대해 제한점(threshold) 변화, 데이터베이스 증가, 트랜잭션수와 아이템 수에 대한 확장성 변화에 대한 실행시간과 메모리 사용량에 대한 성능 평가를 하고 분석하였다.

성능평가 결과에서 IWFPFD, 와 IWFPWA 알고리즘들은 점진적 기법을 사용하며 마이닝을 위한 트리 저장구조 구축을 위해 빈도수 내림차순 (Frequency Descending) 순서와 가중화 올림차순으로 정렬한다. 하지만, IWFPFD, 와 IWFPWA 알고리즘들은 모든 가중화 빈발 패턴들을 마이닝한다. 반면, MWFIM 알고리즘은 가중화 맥시멀 패턴들을 추출하지만 정적인 방법을 사용하여 실제 점진적으로 늘어나는 환경에서는 성능이 매우 제한적이다. IM_WMFI 알고리즘은 점진적인 기법과 맥시멀 가중화 대표패턴을 마이닝 하는 방법을 적용하여 가장 좋은 성능을 보였다.

참고문헌(Reference)

[1] R. Agrawal, R. Srikant, Fast algorithms for mining association rules, In Proceedings of the 20th International Conference on Very Large Data Bases, pp. 487-499, 1994. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.219.6784

[2] C. F. Ahmed, S.K. Tanbeer, B.S. Jeong, Y.K. Lee, and H.J. Choi, Single-pass incremental and interactive mining for weighted frequent patterns, Expert Systems with Applications, 39(9), pp. 7976-7994, Jan. 2012,. https://dl.acm.org/doi/10.1016/j.eswa.2012.01.117

[3] D. Burdick, M. Calimlim, J. Flannick, J. Gehrke, T. Yiu, MAFIA: A maximal frequent itemset algorithm, IEEE Transactions on Knowledge and Data Engineering, 17(11), pp. 1490-1504, 2005.

https://dl.acm.org/doi/abs/10.1109/TKDE.2005.183

[4] A. Chanda, C. Ahmed, Md. Samiullah, C. Leung, A new framework for mining weighted periodic patterns in time series databases, Expert Systems With Applications, 79, pp. 207-224, 2017. https://dl.acm.org/doi/10.1016/j.eswa.2017.02.028

[5] J. Han, J. Pei, Y. Yin, R. Mao, Mining frequent patterns without candidate generation: A frequent-pattern tree approach, Data Min. Knowl. Discov. 8 (1), pp. 53–87, 2004. https://dl.acm.org/doi/10.1145/335191.335372

[6] G. Lee, U. Yun, K. Ryu, Sliding window based weighted maximal frequent pattern mining over data streams, Expert Systems with Applications, 41(2), pp. 694-708, 2014. https://dl.acm.org/doi/abs/10.1016/j.eswa.2013.07.094

[7] G. Lee, U. Yun, Analysis and performance Evaluation of pattern condensing techniques used in representative pattern mining, Journal of Internet Computing and Services, 16(2), pp. 77-83, 2015. https://doi.org/10.7472/jksii.2015.16.2.77

[8] L. Nguyen, G. Nguyen, B. Le, Fast algorithms for mining maximal erasable patterns, Expert Systems With Applications, 124, pp. 50-66, 2019. https://doi.org/10.1016/j.eswa.2019.01.034

[9] G. Pyun, U. Yun, Performance evaluation of approximate frequent pattern mining based on probabilistic technique, Journal of Internet Computing and Services, 14(1), pp. 63-69, 2013. https://doi.org/10.7472/jksii.2013.14.63

[10] G. Pyun, U. Yun, Performance analysis of frequent itemset mining technique based on transaction weight constraints, Journal of Internet Computing and Services, 16(1), pp. 67-74, 2015. https://doi.org/10.7472/jksii.2015.16.1.67

[11] T. Truong, H. Duong, B. Le, P. Fournier-Viger, FMaxCloHUSM: An efficient algorithm for mining frequent closed and maximal high utility sequences, Engineering Applications of Artificial Intelligence, 85, pp. 1-20, 2019. https://doi.org/10.1016/j.engappai.2019.05.010

[12] U. Yun, On pushing weight constraints deeply into frequent itemset mining, Intelligent Data Analysis 13, pp. 359-383, 2009. https://dl.acm.org/doi/abs/10.5555/1551582.1551591

[13] U. Yun, H. Shin, K. Ryu, and E. Yoon, An efficient mining algorithm for maximal weighted frequent patterns in transactional databases, Knowledge-Based Systems, vol. 33, pp. 53-64, 2012. https://dl.acm.org/doi/10.1016/j.knosys.2012.02.002

[14] U. Yun, K. Ryu, Efficient mining of maximal correlated weight frequent patterns, Intelligent Data Analysis, 17(5), pp. 917-939, 2013. https://dl.acm.org/doi/10.5555/2595588.2595598

[15] U. Yun, G. Lee, K. Ryu, Mining maximal frequent patterns by considering weight conditions over data streams, Knowledge Based Systems, 55, pp. 49-65, 2014. https://doi.org/10.1016/j.knosys.2013.10.011

[16] U. Yun, G. Lee, Incremental mining of weighted maximal frequent itemsets from dynamic databases, Expert Systems With Applications, 54, pp. 304–327, 2016. https://doi.org/10.1016/j.eswa.2016.01.049

[17] U. Yun, H. Nam, G. Lee, E. Yoon, Efficient approach for incremental high utility pattern mining with indexed list structure, Future Generation Computer Systems, 95, pp. 221-239, 2019. https://doi.org/10.1016/j.future.2018.12.029

◐ 저 자 소 개 ◑

윤 은 일(Unil Yun)

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

윤 은 미(EunMi Yun)

2011년~현재 백석예술대학교 유아교육학과 조교수
2009년~2010년 아주대학교 교육대학원 겸임교수
2007년 중앙대학교 문학박사
2004년~2010년 뮤직[오디] 연구원
2000년 중앙대학교 문학석사.
1990년 중앙대학교 문학사.
관심분야: 유아컴퓨터 교육, 부모교육, 유아수・과학교육
Email: lovemind114@bau.ac.kr