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

슬라이딩 윈도우 기반의 스트림 하이 유틸리티 패턴 마이닝 기법 성능분석☆

Performance Analysis of Siding Window based Stream High Utility Pattern Mining Methods

양 흥 모[1] 윤 은 일[1*]
Heungmo Ryang Unil Yun

요 약

최근 무선 센서 네트워크, 사물 인터넷, 소셜 네트워크 서비스와 같은 다양한 응용 분야에서 대용량 스트림 데이터가 실시간으로 생성되고 있으며, 효율적인 기법을 통해 처리 및 분석하여 유용한 정보를 찾아내고, 이를 의사 결정을 위해 사용할 수 있도록 하는 것은 중요한 이슈 중에 하나이다. 스트림 데이터는 끊임없이 빠른 속도로 생성되므로 최소한의 접근을 통해 처리해야 하며, 신속한 저전력 처리를 필요로 하는 자원이 제한된 환경에서 분석될 수 있도록 적합한 기법이 요구된다. 이러한 문제를 해결하기 위해, 슬라이딩 윈도우 개념이 제안되어 연구되고 있다. 한편, 대용량 데이터로부터 의미 있는 정보를 찾아내기 위한 데이터 마이닝 기법 중에 하나인 패턴 마이닝은 중요 정보를 패턴 형태로 추출한다. 전통적인 빈발 패턴 마이닝은 이진 데이터베이스를 대상으로 하고 모든 아이템을 동일한 중요도로 고려함으로써 데이터 마이닝 분야에서 중요한 역할을 수행해 왔지만, 실제 데이터 특성을 반영하지 못하는 단점을 지닌다. 하이 유틸리티 패턴 마이닝은 비 이진 데이터베이스로부터 상대적인 아이템 중요도를 반영하여 더욱 의미 있는 정보를 찾아내기 위해 제안되었다. 정적 데이터를 대상으로 하는 하이 유틸리티 패턴 마이닝 기법은 그러나 스트림 데이터 처리에 적합하지 못하다. 제한된 환경에서 스트림 데이터의 특성을 반영하고 효율적으로 처리하여 중요한 정보를 찾아내기 위해 슬라이딩 윈도우 기반의 접근법이 제안되었다. 본 논문은 슬라이딩 윈도우 기반 하이 유틸리티 패턴 마이닝 기법들의 성능을 평가하고 분석하여 해당 기법들의 특성 및 발전 방향을 고찰한다.

☞ 주제어 : 패턴 마이닝, 하이 유틸리티 패턴 마이닝, 슬라이딩 윈도우 모델, 제한된 스트림 환경

ABSTRACT

Recently, huge stream data have been generated in real time from various applications such as wireless sensor networks, Internet of Things services, and social network services. For this reason, to develop an efficient method have become one of significant issues in order to discover useful information from such data by processing and analyzing them and employing the information for better decision making. Since stream data are generated continuously and rapidly, there is a need to deal with them through the minimum access. In addition, an appropriate method is required to analyze stream data in resource limited environments where fast processing with low power consumption is necessary. To address this issue, the sliding window model has been proposed and researched. Meanwhile, one of data mining techniques for finding meaningful information from huge data, pattern mining extracts such information in pattern forms. Frequency-based traditional pattern mining can process only binary databases and treats items in the databases with the same importance. As a result, frequent pattern mining has a disadvantage that cannot reflect characteristics of real databases although it has played an essential role in the data mining field. From this aspect, high utility pattern mining has suggested for discovering more meaningful information from non-binary databases with the consideration of the characteristics and relative importance of items. General high utility pattern mining methods for static databases, however, are not suitable for handling stream data. To address this issue, sliding window based high utility pattern mining has been proposed for finding significant information from stream data in resource limited environments by considering their characteristics and processing them efficiently. In this paper, we conduct various experiments with datasets for performance evaluation of sliding window based high utility pattern mining algorithms and analyze experimental results, through which we study their characteristics and direction of improvement.

☞ keyword : Pattern mining, high utility pattern mining, sliding window model, resource-limited environments

[1] Dept. of Computer Engineering, Sejong University, Seoul, 05006, Korea
* Corresponding author (yunei@sejong.ac.kr)
[Received 10 August 2016, Reviewed 6 September 2016, Accepted 8 November 2016]
☆ 본 연구는 2015년도 정부 교육과학기술부의 재원으로 한국 연구재단(NRF)의 지원을 받아 수행된 연구 사업이며(NRF No. 20152062051, NRF No. 20155054624), 중소기업청에서 지원하는 2015년도 산학연협력 기업부설연구소 지원사업(No. C0261068)의 연구수행으로 인한 결과물임을 밝힙니다.
☆ 본 논문은 2016년도 인터넷정보학회 춘계학술발표대회 최우수논문 추천에 따라 확장 및 수정된 논문임.

1. 서 론

최근 무선 센서 네트워크, 사물 인터넷, 소셜 네트워크 서비스와 같은 다양한 응용 분야에서 대용량 스트림 데이터가 실시간으로 생성됨에 따라 이를 효율적인 기법을 통해 처리 분석하여 유용한 정보를 찾아내기 위한 데이터 마이닝 기술의 중요성이 더욱 높아지고 있다. 스트림 데이터는 끊임없이 빠른 속도로 생성되어 전송되므로 최소한의 접근이 요구된다. 특히, 무선 센서 네트워크나 사물 인터넷 네크워크 같은 환경에서는 신속한 저전력 처리를 필요로 하며, 따라서 이러한 자원이 제한된 환경에서 스트림 데이터 분석을 효율적으로 수행할 수 있도록 하는 적합한 기법이 요구된다. 거대한 데이터베이스 내 지식 발견의 분석 단계인 데이터 마이닝 분야에서, 슬라이딩 윈도우 모델 [5]은 이러한 문제를 해결하기 위해 제안되었으며, 해당 모델은 정해진 크기의 최신 데이터 정보를 유지 및 갱신함으로써 한정된 메모리 자원만을 필요로 한다.

한편, 데이터 마이닝 기법 중에 하나인 패턴 마이닝 [10, 11]은 거대한 데이터베이스로부터 패턴 형태의 의미 있는 정보를 찾아낸다. 이때, 패턴은 아이템 집합으로서 아이템은 마켓 데이터베이스의 상품이나 의료 데이터베이스의 질병 등이 될 수 있다. 전통적인 빈발 패턴 마이닝 [1, 4]은 아이템 발생이 0 또는 1로 표현된 이진 데이터베이스로부터 모든 아이템이 동일한 중요도를 가진다는 가정 하에 빈도수 정보를 기반으로 패턴 마이닝을 수행한다. 즉, 사용자 정의 최소 빈도수 이상으로 빈발하게 발생하는 패턴 정보가 마이닝 된다. 예를 들어, 마켓 데이터베이스에서 맥주와 과자가 동시에 많이 판매되면, 두 아이템의 조합인 {맥주, 과자}라는 패턴이 빈발 패턴으로서 추출된다.

비록 빈발 패턴 마이닝이 데이터 마이닝 분야에서 중요한 역할을 수행해 왔지만, 실제 데이터베이스는 판매 데이터의 수량과 같은 비 이진 정보를 포함할 뿐만 아니라 각 아이템이 판매 이윤과 같은 상대적인 중요도를 가지므로 실제 데이터 특성을 반영하기 어렵다는 단점을 지닌다. 데이터에 대한 다각적 분석이 필요해짐에 따라 이러한 요구를 충족시키기 위한 다양한 패턴 마이닝 접근법들이 개발되었으며, 이러한 접근법들 중에 하나인 하이 유틸리티 패턴 마이닝 [3, 8, 9]은 아이템 수량 정보가 포함된 비 이진 데이터베이스를 대상으로 상대적 아이템 중요도를 반영하여 중요 패턴들을 마이닝 한다. 예를 들어, 마켓 데이터베이스에서 껌이 많이 팔리면 빈발 패턴 마이닝에서는 {껌} 패턴을 추출하지만, 실제 이윤은 많이 남지 않을 수 있으며, 따라서 판매 이윤을 최대화하려는 목적에서는 의미 없는 패턴일 수 있다. 반대로 보석류는 많이 팔리지 않아 빈발 패턴으로 추출되지 않더라도, 판매 이윤이 높아 중요한 패턴으로서 추출될 수 있다. 이를 위해, 하이 유틸리티 패턴 마이닝에서는 빈도수가 아닌 유틸리티라는 개념을 사용하며, 수량과 중요도 곱으로 정의된다.

하이 유틸리티 패턴 마이닝을 통해 실제 데이터의 특성을 반영할 수 있지만, 정적 데이터를 대상으로 하는 기법들 [6, 8, 9]은 데이터를 배치 방식으로 한 번에 처리하므로 스트림 데이터 처리에 적합하지 않다. 이러한 문제를 해결하기 위해, 슬라이딩 윈도우 기반의 하이 유틸리티 패턴 마이닝 [2]이 제안되어 연구되고 있다. 본 논문은 슬라이딩 윈도우 기반 하이 유틸리티 패턴 마이닝 기법들의 성능을 실제 데이터 집합들을 사용한 다양한 실험을 수행하여 성능을 평가 분석하고, 이를 통해 해당 기법들의 특성 및 발전 방향을 고찰한다.

본 논문은 다음과 같이 구성된다. 2장에서 슬라이딩 윈도우 기반 하이 유틸리티 패턴 마이닝과 연관된 영향력 있는 관련 연구들을 살펴본다. 3장에서 슬라이딩 윈도우 기반 하이 유틸리티 패턴 마이닝에 대한 특성을 분석하며, 4장에서 다양한 실험을 통해 해당 기법들의 성능을 평가 분석한다. 마지막으로 5장에서 결론을 내린다.

2. 관련 연구

2.1 빈발 패턴 마이닝

Apriori [1] 그리고 FP-Growth [4]는 빈발 패턴 마이닝을 위한 가장 잘 알려진 기법들로서, 전자는 너비 우선 탐색 (Breath First Search; BFS) 방식을 사용하며, 후자는 깊이 우선 탐색 방식 (Depth First Search; DFS)을 적용한다. Apriori 기반의 접근법은 후보 생성 그리고 검증 방식으로 패턴 마이닝을 수행함으로써, 그 과정에서 수많은 후보 패턴들을 생성하고, 여러 번의 데이터베이스 스캔을 필요로 한다는 단점을 지닌다. FP-Growth 기반의 접근법은 이러한 문제를 극복하기 위해 분할 정복 방식을 적용함으로써, 오직 두 번의 데이터베이스 스캔을 통해 후보 생성 없이 빈발 패턴들을 마이닝 한다.

패턴 마이닝 분야에서 효율적인 마이닝을 위한 근본적인 요소인 안티 모노톤 속성 [1]이 사용되며, 이는 유효하지 않은 패턴으로부터는 이를 포함하는 어떠한 유효 슈

퍼 패턴 (Super pattern)도 생성되지 않음을 의미한다. 즉, 해당 속성을 적용하면, 마이닝 과정에서 주어진 임계치를 만족하지 못하는 후보 패턴이 발견될 때마다 관련 검색 공간을 제거함으로써 효율적으로 유효 패턴들을 찾아낼 수 있다.

2.2 하이 유틸리티 패턴 마이닝

하이 유틸리티 패턴 마이닝 [2, 3, 6]에서는 아이템 수량 및 중요도를 반영함으로써 유효하지 않은 패턴의 슈퍼 패턴도 유효해질 수 있으며, 이로 인해 안티 모노톤 속성을 유지하여 효율적으로 마이닝 과정을 수행하는 데 어려움이 따른다. Two-Phase [6]는 해당 속성을 만족시킨 Apriori 기반의 첫 번째 기법으로서, 주어진 패턴의 슈퍼 패턴들이 가질 수 있는 최대 유틸리티를 의미하는 과추정 개념을 개발 적용하였다. 과추정 개념이 적용된 하이 유틸리티 패턴 마이닝은 먼저 첫 번째 단계에서 주어진 최소 임계치를 만족하는 과추정 유틸리티를 지닌 후보 패턴들을 생성하고, 마지막 단계에서 후보 패턴들의 유틸리티를 계산함으로써 실제 하이 유틸리티 패턴들을 식별한다.

Two-Phase의 과추정 개념이 제안된 이후로 FP-Growth를 기반으로 하는 IHUP [3]를 포함한 정적 데이터베이스를 대상으로 하는 다양한 하이 유틸리티 패턴 마이닝 기법들이 제안되었다.

2.3 스트림 패턴 마이닝

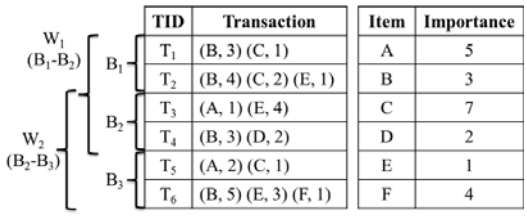
스트림 데이터는 끊임없이 빠르게 지속적으로 생성되어 길이에 제한이 없다는 특성을 지니므로 기존 데이터를 참조하여 처리하는 데는 어려움이 따른다. 특히, 스트림 데이터의 지속적인 빠른 생성 및 제한 없는 길이로 인해 최소한의 접근만으로 처리해야 할 뿐만 아니라 전체 데이터를 메모리에 모두 적재하여 다루는 데는 한계가 있다. 이러한 문제점을 다루기 위해, 다양한 스트림 패턴 마이닝 접근법들이 제안되었으며, 크게 랜드 마크 윈도우 모델, 템프드 윈도우 모델, 슬라이딩 윈도우 모델로 나뉜다. 랜드 마크 윈도우 모델은 특정 지점부터 현재까지 생성된 데이터를 처리하며, 템프드 윈도우 모델은 시간이 지남에 따라 오래된 데이터의 중요도를 감소시켜 스트림 데이터를 처리한다. 슬라이딩 윈도우 모델은 고정된 크기의 윈도우를 이용하여 최신 데이터만을 처리함으로써 자원이 제한된 환경에서도 효율적으로 스트림 데이터를 다룰 수 있도록 한다. 해당 모델에서, 윈도우는 정해진 수의 배치들로 구성되고, 각 배치는 또한 고정된 수의 트랜잭션들을 포함한다. 즉, 윈도우는 동일한 수의 트랜잭션들을 포함하는 배치들의 집합이며, 크기는 하나의 배치가 포함하는 트랜잭션 개수에 윈도우 내 배치 수를 곱하여 계산할 수 있다. 또한, 슬라이딩 윈도우는 최신 데이터가 추가됨에 따라 가장 오래된 배치 데이터를 제거하며, 따라서 고정된 수의 최신 데이터가 저장되어 유지된다.

3. 슬라이딩 윈도우 기반 하이 유틸리티 패턴 마이닝

스트림 패턴 마이닝에서, $I = \{i_1, i_2, ..., i_m\}$ 그리고 $T = \{i_1, i_2, ..., i_l\}$를 서로 구별되는 m개의 아이템 집합 및 I의 부분 집합인 트랜잭션 ($T \subseteq I \wedge 1 \leq l \leq m$)이라고 하면, 데이터 스트림 $DS = [T_1, T_2, ..., T_n]$은 연속적으로 생성된 트랜잭션 집합으로 정의된다. 슬라이딩 윈도우 모델 [5]에서, 윈도우는 동일한 고정된 수만큼의 트랜잭션들 포함하는 겹치지 않는 배치들로 구성된다. 스트림 데이터가 전송되어 저장되는 과정에서 윈도우가 가득 찬 후 새로운 데이터가 도착하면, 가장 오래된 배치 정보를 윈도우로부터 제거하고, 최신 배치를 추가하여 새로운 데이터 정보를 저장한다.

하이 유틸리티 패턴 마이닝 [2, 3, 6]에서, 임의의 트랜잭션 T_d ($1 \leq d \leq n$) 내 각 아이템 i_p ($1 \leq p \leq m$)은 내부 유틸리티라 하는 비 이진 값을 가지며, $iu(i_p, T_d)$로 표현된다. 또한, 각 아이템 i_p이 트랜잭션들에서 개별적인 내부 유틸리티를 가지는 것과 동시에 전체 데이터베이스에 대해 외부 유틸리티라 하는 고정된 중요도 값을 가지며, $eu(i_p)$로 표현된다. 빈발 패턴 마이닝 [1, 4]에서 빈도수를 기준으로 패턴의 유효성을 검증하는 것과는 달리, 하이 유틸리티 패턴 마이닝에서는 내부 그리고 외부 유틸리티 곱으로 표현되는 유틸리티를 기준으로 한다. 임의의 트랜잭션 T_d에 포함된 각 아이템 i_p는 T_d 내에서 $iu(i_p, T_d) \times eu(i_p)$로 정의되는 유틸리티 $u(i_p, T_d)$를 가지고, T_d의 유틸리티는 내부 아이템들의 유틸리티 합 $\sum_{p=1}^{l} u(i_p, T_d)$으로 정의되며, 전체 데이터베이스에 대한 i_p의 유틸리티 $u(i_p)$는 해당 아이템을 포함하는 모든 트랜잭션 내 i_p의 유틸리티 합이다. 따라서 각 패턴 $P = \{i_1, i_2, ..., i_k\}$ ($1 \leq k \leq m$)은 해당 패턴을 포함하는 모든 트랜잭션 내 P의 유틸리티 합을 의미하고, $u(P)$로 표현된다.

그림 1은 데이터 스트림 예제로서 T_1부터 T_6까지 차례대로 트랜잭션들이 생성되어 도착한 결과이며, 슬라이딩 윈도우는 2개의 트랜잭션들로 구성된 2개의 배치 집합이다. 즉, T_1부터 T_4까지 트랜잭션들이 도착함에 따라 첫 번째 윈도우 W_1가 가득 차며, 새로운 배치 정보인 T_5부터 T_6가 도착하면, 가장 오래된 배치 B_1이 제거되고, B_3가 추가되어, 두 번째 윈도우 W_2로 갱신된다. 첫 번째 트랜잭션 T_1은 B 그리고 C로 구성되고, 해당 아이템들은 T_1에서 $iu(B, T_1) = 3$ 그리고 $iu(C, T_1) = 1$의 내부 유틸리티를 가진다. 또한, B와 C가 전체 데이터베이스에 대해 $eu(B) = 3$ 그리고 $eu(C) = 7$의 외부 유틸리티를 가지므로 해당 아이템들은 T_1에서 유틸리티가 $u(B, T_1) = iu(B, T_1) \times eu(B) = 9$ 그리고 $u(C, T_1) = iu(C, T_1) \times eu(C) = 7$이다. 따라서 T_1의 유틸리티는 $u(B, T_1) + u(C, T_1) = 16$이다. 마지막으로 패턴 {B, C}는 T_1을 포함하여 T_2에도 포함돼 있으므로 {B, C}의 유틸리티는 $u(BC) = u(BC, T_1) + u(BC, T_2) = 16 + 26 = 42$이다.



TID	Transaction
T_1	(B, 3) (C, 1)
T_2	(B, 4) (C, 2) (E, 1)
T_3	(A, 1) (E, 4)
T_4	(B, 3) (D, 2)
T_5	(A, 2) (C, 1)
T_6	(B, 5) (E, 3) (F, 1)

Item	Importance
A	5
B	3
C	7
D	2
E	1
F	4

(그림 1) 데이터 스트림 및 아이템 중요도 예제
(Figure 1) Example of Stream Data and Item Importance

슬라이딩 윈도우 기반 하이 유틸리티 패턴 마이닝 [2]은 사용자 요청이 들어오면, 현재 윈도우에 저장된 최신 데이터로부터 최소 유틸리티 임계치를 만족하는 모든 하이 유틸리티 패턴들을 마이닝 한다. HUPMS [2]는 과추정 모델 [6]이 적용된 슬라이딩 윈도우 기반 하이 유틸리티 패턴 마이닝 기법으로서 트리 자료구조를 이용하여 마이닝 과정을 수행한다.

4. 슬라이딩 윈도우 기반 하이 유틸리티 패턴 마이닝 기법의 성능분석

4.1 실험 환경

본 논문은 과추정 모델 [6]이 적용된 슬라이딩 윈도우 기반 하이 유틸리티 패턴 마이닝 기법인 HUPMS [2]의 특성을 분석하기 위해, 동일한 과추정 모델을 적용한 일반 하이 유틸리티 패턴 마이닝 기법인 IHUP [3]와 성능 비교 분석을 수행한다. 모든 기법들은 C/C++ 언어로 구현되었으며, 4.0GHz 인텔 프로세서 그리고 32GB 메모리 환경의 Windows 7 운영체제에서 실행되었다.

(표 1) 데이터 집합들의 특성
(Table 1) Characteristics of Datasets

데이터 집합	트랜잭션 수	아이템 수	평균 길이
Chain-store	1,112,949	468	7.2
Accidents	340,183	468	33.8
T10I4D100K	100,000	1,000	10

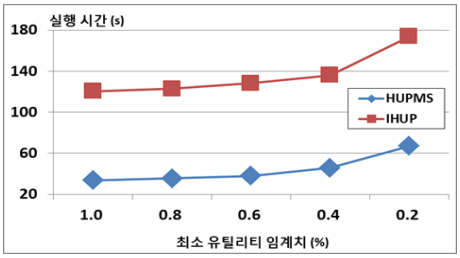
비교 기법들에 대한 성능평가 실험에는 표 1의 실제 그리고 가상 데이터 집합들이 사용되었다. 실제 데이터 집합인 Accidents 그리고 가상 데이터 집합인 T10I4D100K는 FIMI Repository (http://fimi.ua.ac.be/data)에서 획득하였고, 실제 유틸리티 정보가 포함돼 있는 나머지 실제 데이터 집합인 Chain-store는 MineBench 2.0 [7]에서 획득하였다. 한편, Accidents는 교통 사고 정보 그리고 Chain-store는 리테일 판매 데이터가 포함돼 있으며, T10I4D100K는 IBM 가상 데이터 생성기 [1]를 통해 생성되었다. 특히, Chain-store를 제외한 나머지 데이터 집합들은 유틸리티 정보를 포함하고 있지 않으므로 1부터 10까지의 정수 내부 유틸리티 그리고 0.01부터 10.00까지의 실수 외부 유틸리티를 임의로 생성하여 적용하였다.

성능평가 실험은 2개의 배치로 구성된 2개의 윈도우로 나누어 수행하며, HUPMS는 슬라이딩 윈도우 모델을 적용하고, IHUP는 마이닝 과정을 2번 나눠 실행한다. 성능분석은 변화하는 최소 유틸리티 임계치에 대한 비교 기법들의 전체 실행 시간 그리고 최대 메모리 사용량 관점에서 수행된다.

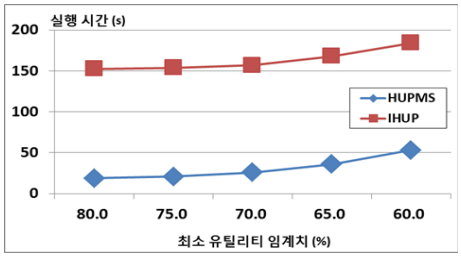
4.2 실행시간 성능분석

그림 2, 3, 4는 변화하는 최소 유틸리티 임계치에 따른 두 비교 알고리즘들, HUPMS 그리고 IHUP, 의 세 데이터 집합들, Chain-store, Accidents, T10I4D100K, 을 사용한 실행시간 측면에 대한 실험 결과들이다. 성능평가 결과 그림들에서, 비교 알고리즘들은 모두 최소 유틸리티 임계치가 감소함에 따라 성능 또한 악화되었으며, 그 이유는 동일하게 적용된 과추정 모델에 의해서 생성되는 후보 패턴 수가 증가하여 더 많은 처리 시간을 필요로 하였기 때

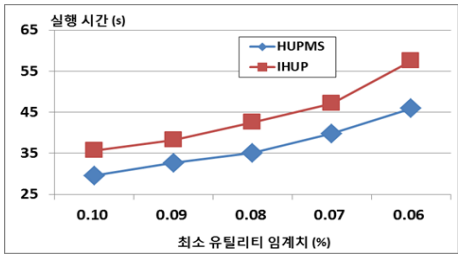
문이다. 그럼에도, HUPMS가 스트림 데이터로부터 하이 유틸리티 패턴 마이닝을 수행하는 데 더 효율적인 것을 실행시간 차이로부터 알 수 있다.



(그림 2) 실행시간 결과 (Chain-store)
(Figure 2) Runtime Result (Chain-store)



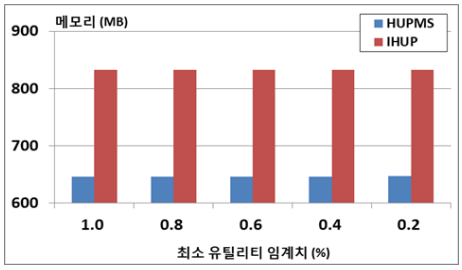
(그림 3) 실행시간 결과 (Accidents)
(Figure 3) Runtime Result (Accidents)



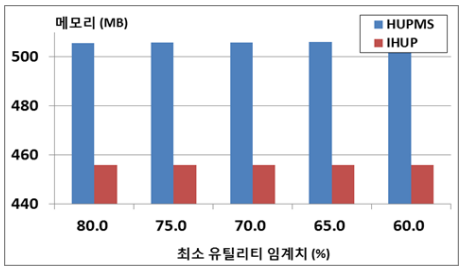
(그림 4) 실행시간 결과 (T10I4D100K)
(Figure 4) Runtime Result (T10I4D100K)

4.3 메모리 사용량 성능분석

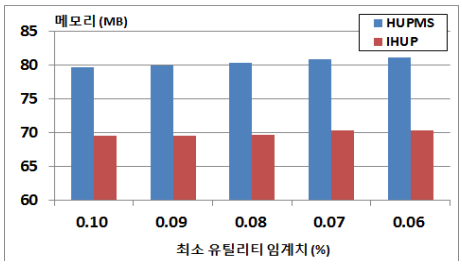
그림 5, 6, 7은 4.2의 실행시간 측면의 성능평가를 위한 실험과 동일한 조건에서 수행된 비교 알고리즘들에 대한 메모리 사용량 실험 결과들이다. 최소 유틸리티 임계치가 감소함에 따라 성능이 악화되었던 이전 실험과는 달리, 메모리 사용량에 대한 이번 실험에서는 최소 유틸리티 임계치의 변화에 상관 없이 비교 알고리즘들이 거의 일정한 메모리를 사용했음을 알 수 있다. 그 이유는 데이터 집합에 대한 최소한의 접근을 통해 하이 유틸리티 패턴 마이닝 과정을 수행하는 HUPMS 그리고 IHUP는 한 번의 스캔을 통해 데이터 정보를 자료구조에 모두 삽입하므로 전체 메모리 사용량에 큰 차이가 없다. 즉, 전역 자료구조를 구축하는 데 필요한 메모리가 그로부터 재귀적으로 마이닝 과정을 수행하는 데 필요한 메모리보다 월등히 많기 때문이다. 또한, 데이터 집합에 따라서 메모리 사용량은 HUPMS가 더 적을 수도 또는 그 반대일 수도 있음을 알 수 있었다.



(그림 5) 메모리 사용량 결과 (Chain-store)
(Figure 5) Memory Usage Result (Chain-store)



(그림 6) 메모리 사용량 결과 (Accidents)
(Figure 6) Memory Usage Result (Accidents)



(그림 7) 메모리 사용량 결과 (T10I4D100K)
(Figure 7) Memory Usage Result (T10I4D100K)

5. 결 론

본 논문은 하이 유틸리티 패턴 마이닝에서 안티 모노톤 속성을 만족시키기 위한 동일한 과추정 모델을 사용하고, 주어진 데이터에 대한 한 번의 스캔을 통해 전역 자료구조를 구축하여 그로부터 마이닝 과정을 재귀적으로 수행하는 두 알고리즘들, HUPMS 그리고 IHUP, 대한 성능평가 분석을 수행하였다. 실제 그리고 가상 데이터 집합들을 사용한 실행 시간 그리고 최대 메모리 사용량 측면에 대한 성능평가 실험 결과, 슬라이딩 윈도우 기반의 기법인 HUPMS가 일반 배치 방식의 기법인 IHUP보다 더 빠른 속도로 패턴 마이닝 과정을 수행함을 알 수 있었다. 특히, 두 기법들은 최소 유틸리티 임계치에 상관없이 고정된 크기의 메모리 자원을 사용하였다.

두 비교 알고리즘들 모두 과추정 모델을 적용함으로써 비록 한 번의 스캔을 통해 전역 자료구조를 구축하고, 그로부터 하이 유틸리티 패턴들을 마이닝 하였지만, 마이닝 과정에서 생성된 후보 패턴들로부터 실제 하이 유틸리티 패턴 정보를 식별하기 위한 추가적인 스캔을 필요로 한다. 이는 최소한의 접근을 통한 처리가 필요한 스트림 환경에서 단점으로 작용할 수 있다. 최근 마이닝 과정에서 후보 패턴 생성 없이 하이 유틸리티 패턴들을 마이닝 하기 위한 기법이 제안되었으며, 따라서 해당 기법을 활용하여 한 번의 스캔으로 자료구조를 구축하면서 후보 패턴을 생성하지 않으면, 추가적인 스캔 없이 스트림 데이터를 더욱 효과적으로 처리할 수 있을 것으로 예상된다.

참 고 문 헌 (Reference)

[1] R. Agrawal, R. Srikant, “Fast algorithms for mining association rules”, in Proc. of the 20th International Conference on Very Large Data Bases, 1994, pp. 487-499.

[2] C.F. Ahmed, S.K. Tanbeer, B.S. Jeong, H.J. Choi, "Interactive mining of high utility patterns over data streams", Expert Systems with Applications, vol. 39, no. 15, 2012, pp. 11979-11991.

[3] C.F. Ahmed, S.K. Tanbeer, B.S. Jeong, Y.K. Lee, "Efficient tree structures for high utility pattern mining in incremental databases", IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 12, 2009, pp. 1708-1721.

[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, No.1, pp.53-87, 2004. http://dx.doi.org/10.1023/B:DAMI.0000005258.31418.83

[5] H.F. Li, S.Y. Lee, "Mining frequent itemsets over data streams using efficient window sliding techniques", Expert Systems with Applications, vol. 36, no. 2, 2009, pp. 1466-1477.

[6] Y. Liu, W.K. Liao, A.N. Choudhary, "A two-phase algorithm for fast discovery of high utility itemsets", in Proc. of the 9th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, 2005, pp. 689-695.

[7] J. Pisharath, Y. Liu, B. Ozisikyilmaz, R. Narayanan, W.K. Liao, A. Choudhary, G. Memik, NU-MineBench version 2.0 dataset and technical report, http://cucis.ece.northwestern.edu/projects/DMS/MineBench.html

[8] V.S. Tseng, B.-E. Shie, C.-W. Wu, and P.S. Yu, "Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases", IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 8, 2013, pp. 1772-1786. http://dx.doi.org/10.1109/TKDE.2012.59

[9] U. Yun, H. Ryang, and K. Ryu, "High utility itemset mining with techniques for reducing overestimated utilities and pruning candidates", Expert Systems with Applications, Vol. 41, No. 8, pp. 3861-3878, 2014. http://dx.doi.org/10.1016/j.eswa.2013.11.038

[10] G. Lee and U. Yun, "Analysis and Performance Evaluation of Pattern Condensing Techniques used in Representative Pattern Mining", Journal of Internet Computing and Services, Vol. 16, No. 2, pp. 77-83, 2015. http://dx.doi.org/10.7472/jksii.2015.16.2.77

[11] G. Pyun and U. Yun, "Performance evaluation of approximate pattern mining based on probabilistic technique", Journal of Internet Computing and Services, Vol. 14, No. 1, pp. 63-69, 2013. http://dx.doi.org/10.7472/jksii.2013.14.63

◐ 저 자 소 개 ◑

양 흥 모 (Heungmo Ryang)
2011년 충북대학교 컴퓨터공학전공 학사. (공학사)
2013년 충북대학교 대학원 컴퓨터공학 석사. (공학석사)
2013년~현재 세종대학교 대학원 컴퓨터공학 박사과정. (공학박사)
관심분야 : 데이터마이닝, 정보검색, 데이터베이스.
E-mail : ryang@sju.ac.kr

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