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

대표 패턴 마이닝에 활용되는 패턴 압축 기법들에 대한 분석 및 성능 평가☆

Analysis and Performance Evaluation of Pattern Condensing Techniques used in Representative Pattern Mining

이 강 인[1] 윤 은 일[1*]
Gang-in Lee Un-il Yun

요 약

데이터 마이닝에서 활발히 연구되고 있는 주요 분야들 가운데 하나인 빈발 패턴 마이닝은 대규모의 데이터 집합 또는 데이터베이스로부터 숨겨진 유용한 패턴 정보를 추출하기 위한 방법이다. 또한 이 기법으로 얻을 수 있는 결과물을 통해 데이터베이스내의 다양하고 중요한 특징들을 더욱 손쉽게 자동적으로 분석할 수 있기 때문에 많은 응용영역에도 활발히 적용되고 있다. 하지만 이러한 데이터베이스로부터 단순히 사용자에 의해 설정된 최소 지지도 임계값만을 가지고 이를 만족하는 모든 패턴들을 추출하는 기존의 전통적인 빈발 패턴 마이닝 방식은 데이터베이스의 특성과 임계값 설정의 정도에 따라 극도로 많은 수의 결과 패턴을 생성하는 문제를 가지며, 이에 따른 시간 및 공간 자원의 낭비를 초래한다. 또한 과도하게 생성된 패턴에 대한 분석의 어려움 역시 심각한 문제가 된다. 기존의 빈발 패턴 마이닝 접근방법들이 직면한 이러한 문제를 해결하고자, 데이터베이스로부터 가능한 모든 빈발 패턴들을 마이닝하는 것이 아닌, 이들에 대한 대표 패턴들만은 선별적으로 추출할 수 있도록 하는 대표 패턴 마이닝의 개념과 다양한 관련 기법들이 제안되었다. 본 논문에서는 생성되는 각 패턴의 최대성 또는 폐쇄성을 고려하는 패턴 압축 기법들에 대한 특성들을 기술하고, 이에 대한 비교 및 분석을 진행한다. 최대 빈발 패턴 혹은 닫힌 빈발 패턴들을 마이닝함으로써, 효과적인 패턴 압축이 가능하며, 더 적은 시공간 자원으로 마이닝 작업을 수행할 수 있다. 또한 압축된 패턴들은 필요시 다시 원래의 패턴 형태로 복구가 가능한 특징이 있으며, 특히 닫힌 패턴 접근 방법을 이용하면 패턴을 압축하고 다시 해제하는 과정에서 어떠한 정보의 손실도 일어나지 않는다. 본 논문에서는 같은 플랫폼 상에서 동일한 구현 수준의 알고리즘에 대해 실세계로부터 추적된 실 데이터셋들을 가지고 상기 기법들에 대한 성능평가를 진행함으로써, 각 기법이 패턴 생성, 수행 시간, 메모리 사용량과 같은 실제적인 마이닝 성능에 대해 어떠한 영향을 미치는지에 대한 심층적 분석결과를 보인다.

☞ 주제어 : 닫힌 패턴, 데이터 마이닝, 빈발 패턴 마이닝, 최대 패턴, 성능 평가, 대표 패턴 마이닝

ABSTRACT

Frequent pattern mining, which is one of the major areas actively studied in data mining, is a method for extracting useful pattern information hidden from large data sets or databases. Moreover, frequent pattern mining approaches have been actively employed in a variety of application fields because the results obtained from them can allow us to analyze various, important characteristics within databases more easily and automatically. However, traditional frequent pattern mining methods, which simply extract all of the possible frequent patterns such that each of their support values is not smaller than a user-given minimum support threshold, have the following problems. First, traditional approaches have to generate a numerous number of patterns according to the features of a given database and the degree of threshold settings, and the number can also increase in geometrical progression. In addition, such works also cause waste of runtime and memory resources. Furthermore, the pattern results excessively generated from the methods also lead to troubles of pattern analysis for the mining results. In order to solve such issues of previous traditional frequent pattern mining approaches, the concept of representative pattern mining and its various related works have been proposed. In contrast to the traditional ones that find all the possible frequent patterns from databases, representative pattern mining approaches selectively extract a smaller number of patterns that represent general frequent patterns. In this paper, we describe details and characteristics of pattern condensing techniques that consider the maximality or closure property of generated frequent patterns, and conduct comparison and analysis for the techniques. Given a frequent pattern, satisfying the maximality for the pattern signifies that all of the possible super sets of the pattern must have smaller support values than a user-specific minimum support threshold; meanwhile, satisfying the closure property for

[1] Dept. of Computer Engineering, Sejong University, Seoul, 143-747, Korea
* Corresponding author (yunei@sejong.ac.kr)
[Received 26 January 2015, Reviewed 9 February 2015, Accepted 18 March 2015]
☆ 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 연구 사업이며(NRF No.2013-005682), 중소기업청에서 지원하는 2015년도 산학연협력 기술개발사업(No. C0232102)의 연구수행으로 인한 결과물임을 밝힙니다.
☆ 본 논문은 2014년도 인터넷정보학회 추계학술발표대회 우수논문 추천에 따라 확장 및 수정된 논문임.

the pattern means that there is no superset of which the support is equal to that of the pattern with respect to all the possible super sets. By mining maximal frequent patterns or closed frequent ones, we can achieve effective pattern compression and also perform mining operations with much smaller time and space resources. In addition, compressed patterns can be converted into the original frequent pattern forms again if necessary; especially, the closed frequent pattern notation has the ability to convert representative patterns into the original ones again without any information loss. That is, we can obtain a complete set of original frequent patterns from closed frequent ones. Although the maximal frequent pattern notation does not guarantee a complete recovery rate in the process of pattern conversion, it has an advantage that can extract a smaller number of representative patterns more quickly compared to the closed frequent pattern notation. In this paper, we show the performance results and characteristics of the aforementioned techniques in terms of pattern generation, runtime, and memory usage by conducting performance evaluation with respect to various real data sets collected from the real world. For more exact comparison, we also employ the algorithms implementing these techniques on the same platform and Implementation level.

☞ keyword : Closed pattern, Data mining, Frequent pattern mining, Maximal pattern, Performance evaluation, Representative pattern mining

1. 서 론

대규모의 데이터 집합 혹은 데이터베이스로부터 숨겨진 유용한 정보를 찾기 위해 데이터 마이닝의 개념이 제안된 이래로 다양한 접근 방법들이 제안되어 왔다. 이들 가운데 한 분야인 빈발 패턴 마이닝은 유용한 정보를 패턴 형태의 정보로 추출하기 위한 방법을 말하며, 이 방법을 통해 얻을 수 있는 패턴 결과물을 이용해 다양한 데이터베이스의 중요한 특징들을 더욱 손쉽게 자동으로 분석할 수 있기 때문에, 빈발 패턴 마이닝은 또한 많은 응용분야 [2, 3, 10, 11]에서 활발히 연구되고 있고 확률 기반의 근사 마이닝 [8], 다중 최소 지지도를 이용한 마이닝 [9], 그래프 마이닝 [13] 등의 다양한 접근 방법들이 제안되고 있다. 전통적인 빈발 패턴 마이닝의 접근방법들 [1, 5, 7]은 주어진 데이터베이스로부터 사용자 정의 최소 지지도 임계값을 만족하는 모든 가능한 패턴 조합들을 추출하는 것을 목표로 한다. 여기서, 데이터베이스의 크기가 커지고 복잡해질수록, 그리고 주어진 임계값이 낮을수록 생성되는 패턴의 수가 기하급수적으로 커지는 특성을 갖는다. 빈발 패턴 마이닝의 개념이 처음 등장했을 당시와 비교해, 현재의 데이터베이스는 다양한 데이터가 계속적으로 축적됨에 따라 과거에 비해 더욱 복잡해지고 대규모화되는 성향을 보인다. 따라서 이로부터 마이닝되는 패턴의 수 역시 감당할 수 없을 만큼 커지는 경우가 비일비재하며, 이에 따른 연산의 오버헤드와 패턴 분석의 어려움 역시 전통적인 빈발 패턴 마이닝의 주요 한계점으로 지적되고 있다. 이러한 문제를 해결하고자, 모든 빈발 패턴들을 마이닝하는 것이 아닌, 대표 패턴들만을 선별해 추출하기 위한 대표 패턴 마이닝의 개념과 다양한 접근방법 [4, 6, 12]이 제안되어 왔다. 본 논문에서는, 대표 패턴 마이닝의 기법인 패턴의 최대성 혹은 폐쇄성을 고려하는 패턴 압축 기법들에 대한 특성들에 대해 논하고 이들에 대한 객관적인 환경에서의 성능평가를 진행함으로써 각 기법에 대한 심층적인 분석 결과를 제공한다.

2. 관련 연구

빈발 패턴 마이닝의 개념이 등장한 이래로, 다양한 마이닝 방법들이 제안되어 왔다. Apriori [1]는 빈발 패턴을 마이닝하기 위해 고안된 최초의 알고리즘으로써, 너비 우선 탐색 방식 (Breadth-First Search)에 기반을 두고 마이닝 작업을 수행한다. FP-growth [5]은 Apriori 알고리즘의 단점을 해결하고 더욱 효율적으로 마이닝 연산을 수행하기 위해 제안된 알고리즘으로써, FP-tree로 명명되는 특수한 트리구조를 이용해 깊이 우선 탐색 방식 (Depth-First Search)과 분할 정복 (Divide-and-Conquer) 방식을 이용해 빈발 패턴을 추출한다. 빈발 패턴 마이닝의 정의는 다음과 같다. 먼저, 하나의 데이터베이스는 다수의 트랜잭션들로 구성되며, 각 트랜잭션은 다시 다수의 아이템들로 이루어질 수 있다. 즉, 데이터베이스와 트랜잭션은 각각 $DB = \{T_1, T_2, \ldots T_n\}$과 $T_k = \{i_1, i_2, \ldots, i_m\}$ $(1 \leq k \leq n)$로 표현될 수 있다. $I = \{i_1, i_2, \ldots, i_x\}$가 DB를 구성하고 있는 모든 중복되지 않는 아이템의 집합이라고 할 때, 다음과 같은 조건 $m \leq x$와 $\forall T \subseteq I$가 성립한다. 따라서 DB로부터 생성되는 모든 패턴은 I의 부분집합이다. 어느 한 패턴, X의 지지도 (혹은 빈발도), $Support(X)$는 다음과 같이 계산된다.

$$Support(X) = \frac{\sum_{i=1}^{n} f(X, T_i)}{n}, \quad f(X, T_i) = \begin{cases} 1, \text{if } X \subseteq T_i \\ 0, otherwise \end{cases}$$

$Support(X)$가 사용자 정의 최소 지지도 임계값, δ보다 작지 않을 때, X는 빈발 패턴으로 정의된다. 여기서, 안티

모노톤 속성 (*Anti-monotone property*) [1]에 의해, X의 모든 하위 집합들 역시 빈발 패턴이 된다. 그러므로 X를 구성하는 아이템의 수를 k라고 하면, 2^k만큼의 패턴을 마이닝해야 한다. k값이 비교적 작은 경우에는 상대적으로 적은 패턴들만을 생성하면 되지만, k가 커질수록 마이닝되는 패턴의 수가 기하급수적으로 상승한다. 예를 들어, 만약 k가 30이라고 한다면, 단 하나의 패턴 X와 관련하여 1,073,741,824개의 하위 패턴을 함께 생성해야만 하는 문제점이 있다. 이는 막대한 연산의 오버헤드를 초래함은 물론 결과 패턴에 대한 분석 작업을 어렵게 하는 요소로 작용한다. 이러한 문제점을 극복하고자 대표 패턴 마이닝의 개념이 등장했으며, 다양한 접근 방법을 통해 효율적으로 패턴을 추출하는 알고리즘들이 활발히 제안되어 왔다. 다음 장에서는 대표 패턴 마이닝에 대한 기법과 특징들의 자세한 사항들을 기술한다.

3. 대표 패턴 마이닝의 패턴 압축 기법

본 장에서는 전통적인 빈발 패턴 마이닝이 갖는 한계점인 과도한 패턴 생성 문제와 그에 따른 시간 및 메모리 자원의 낭비 등의 문제들을 해결하고자 고안된 대표 패턴 마이닝 방법에 대한 사항들을 기술하며, 특히 대표 패턴 마이닝의 대표적인 두 방법인, 패턴의 최대성 속성을 고려한 최대 빈발 패턴 마이닝과 패턴의 폐쇄성 속성을 고려한 닫힌 빈발 패턴 마이닝에 대한 사항들을 기술하며, 각 기법의 상이한 특성과 장단점을 논한다.

3.1. 최대성을 고려한 패턴 압축 기법

기존의 전통적인 접근방법들처럼 모든 빈발 패턴을 마이닝하는 대신, 패턴의 최대성을 고려한 대표 패턴들만을 선별적으로 마이닝함으로써, 상기 문제점들을 효과적으로 해결할 수 있는 방안을 고려할 수 있다. 여기서 패턴의 최대성은 다음과 같이 정의된다. 어떤 패턴, X와 이로부터 생성될 수 있는 상위 패턴 X'들의 집합 $\Gamma' = \{X'_1, X'_2, ..., X'_k\}$ 있다고 가정하자. 그러면, 다음과 같은 조건을 만족할 때, X는 최대 빈발 패턴으로 정의된다.

$$Support(X) \geq \delta,\ \Gamma = \{X' \mid Support(X') < \delta\}$$

이러한 제약조건을 통해 생성되는 패턴의 수를 획기적으로 줄일 수 있다. 즉, 기존에 단순히 설정된 최소 지지도 임계값 이상의 값을 갖는 모든 빈발 패턴을 추출하는 것이 아닌, 상기 수식의 조건을 추가적으로 만족시키는 소수의 패턴들만을 선별적으로 마이닝함으로써 생성되는 패턴의 수를 획기적으로 줄일 수 있다. 최상의 경우를 예로 들어보면, 위에서 어떤 빈발 패턴 X가 30개의 아이템들로 구성된다면, 1,073,741,824개의 패턴이 함께 생성되어야 함을 알 수 있었다. 하지만, 최대성을 고려한다면 최상의 경우에는 오직 1개의 패턴만을 최대 빈발 패턴으로 추출하면 되기 때문에 마이닝에 필요한 연산의 수를 획기적으로 줄일 수 있다. 한편으로, 패턴의 최대성을 판단하기 위해서는, 추가적인 자료구조가 필요하다. 그 이유는 임계값과 최대성 조건을 만족하는 어떤 패턴을 추출했다고 했을 때, 과연 이 패턴이 실제로 최대 빈발 패턴이 될 수 있는지, 아니면 기존에 추출된 최대 빈발 패턴의 하위집합일지에 대한 판단이 필요하기 때문이다. 따라서 최대 빈발 패턴 마이닝 알고리즘들은 이러한 역할을 하는 추가적인 자료구조를 운용해 최대 빈발 패턴들을 추출한다.

최대성을 고려한 대표적인 패턴 마이닝 기법인 FPmax*의 경우를 살펴보면, 이 알고리즘은 FP-Growth의 기본적인 마이닝 절차를 따르며 패턴 마이닝 작업을 수행한다. 재귀적 호출을 통해 조건적 트리 구조를 구축하는 중에 단일 경로를 갖는 트리가 생성되면 해당 트리와 현재까지 누적된 프리픽스간의 조합을 통해 다수의 패턴을 생성하는 FP-Growth와 달리, FPmax*는 단일 경로를 갖는 조건적 트리가 생성될 경우 현재까지 누적된 프리픽스와 해당 트리의 모든 아이템을 합친 결과를 후보 패턴으로 생성하며, 이후에 이를 기존에 유효한 최대 빈발 패턴으로 추출한 결과를 저장해 놓은 트리 구조와 비교하는 하위집합 확인 작업을 수행하여, 만약 해당 후보 패턴이 기존의 어떠한 패턴의 하위 집합에도 속하지 않을 경우 이를 새로운 유효 패턴으로 고려해 마이닝한다.

한편으로, 상기 조건을 적용해 대표패턴을 마이닝할 경우, 이후에 다시 원본 빈발 패턴 정보를 추출할 때, 일부 패턴 손실이 일어날 수 있기 때문에 패턴의 압축과 복원 과정이 동시에 필요한 환경에서는 이 기법이 부적합하다.

3.2. 폐쇄성을 고려한 패턴 압축 기법

패턴의 압축과 복원 과정에 대한 무결성을 보장함과 동시에 대표 패턴들을 마이닝하기 위한 방법으로 폐쇄성을 고려한 패턴 압축 기법이 사용될 수 있다. 폐쇄성의 정의는 다음과 같다. 어느 패턴, X와 이로부터 생성될 수

있는 상위 패턴 X'들의 집합 $\Lambda = \{X'_1, X'_2, ..., X'_k\}$ 있을 때, 다음과 같은 조건을 만족한다면, X는 폐쇄 빈발 패턴으로 정의된다.

$$Support(X) \geq \delta,\ \Lambda = \{ X' \mid Support(X') \neq Support(X)\}$$

상기 조건을 만족하는 폐쇄 빈발 패턴들만을 선별하여 추출함으로써, 전통적인 빈발 패턴 마이닝에 비해 더 적은 수의 대표패턴들을 추출할 수 있다. 최대 빈발 패턴 마이닝의 경우와 마찬가지로, 이 폐쇄 빈발 패턴 마이닝을 수행하는데 역시 추가적인 자료구조가 필요하다. 마찬가지로, 그 이유는 현재 상태에서는 임계값과 폐쇄성 조건을 만족하는 패턴일지라도 기존에 추출되었던 폐쇄 빈발 패턴들과 비교 시 폐쇄성 속성을 잃을 수 있기 때문이다.

폐쇄성을 고려한 대표적인 패턴 마이닝 기법인 FPclose 역시 FPmax*와 같이 FP-Growth의 기본적인 마이닝 절차를 따른다. FPclose와 FPmax* 간의 가장 큰 차이점은 재귀 호출을 통한 분할 정복 방식의 마이닝 과정을 수행하면서 단일 경로를 갖는 조건적 트리를 생성했을 경우 FPclose는 FPmax*에서 고려하는 최대성 대신 폐쇄성을 고려하여 후보 패턴의 유효성을 검사한다.

비록 최대성을 고려하는 기법에 비해 이 기법의 패턴 압축 효과는 떨어지지만, 압축된 패턴을 다시 원래의 빈발 패턴으로 복원하는 과정에서 어떠한 패턴 손실도 없이 해당 과정을 완벽히 수행할 수 있기 때문에, 마이닝의 효율성과 패턴의 무결성을 동시에 요구하는 마이닝 환경에서는 이 방법이 더욱 효과적이다. 표 1은 상기 패턴 압축 기법들 간의 비교 내용을 정리한 결과이다.

(표 1) 패턴 압축 기법의 분석 방식별 비교

(Table 1) Comparison of pattern condensing techniques

패턴 압축 기법	패턴 압축 효과	패턴복원
최대성을 고려한 패턴 압축 기법	높음	손실 가능성 존재
폐쇄성을 고려한 패턴 압축 기법	비교적 낮음	보장

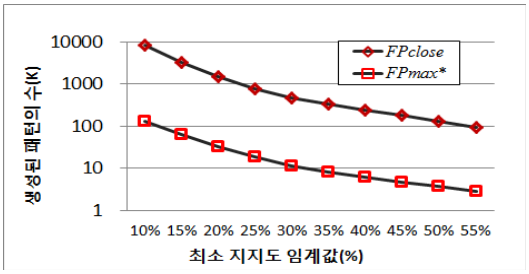
3.3. 마이닝 환경에 따른 대표 패턴 마이닝 기법의 적용

패턴의 최대성 속성을 고려한 최대 빈발 패턴 마이닝은 패턴의 압축 효율을 극대화 한 방법으로써, 일반적으로 패턴의 폐쇄성 속성을 고려한 닫힌 빈발 패턴 마이닝 기법에 비해 더욱 적은 메모리 소모량을 가지고 더욱 빠르게 마이닝 작업을 수행 할 수 있는 장점이 있다. 하지만, 이런 방법으로 생성된 최대 빈발 패턴을 다시 원래의 빈발 패턴 형태로 전환시킬 경우, 최대 빈발 패턴의 특성으로 인해 상황에 따라 패턴 정보의 손실이 발생할 수 있다. 반면에, 닫힌 빈발 패턴 마이닝은 비록 패턴의 압축 효율은 최대 빈발 패턴 마이닝의 효율보다 떨어지지만 이로부터 얻어진 닫힌 빈발 패턴을 다시 원래의 빈발 패턴 형태로 되돌리는 과정에서 어떠한 패턴 손실도 일어나지 않는다는 특징이 있다. 따라서 상기 두 기법은 마이닝 환경에 따라 다음과 같이 활용될 수 있다. 먼저 최대 빈발 패턴 마이닝의 경우에는, 마이닝의 효율성을 중요시 여기는 데이터 스트림 패턴 마이닝 환경에 효과적으로 적용될 수 있다. 데이터 스트림 환경에서는 수많은 데이터들이 끊임없이 축적되기 때문에, 패턴의 복원성이나 정확성도 중요하지만 특히 축적된 데이터로부터 즉각적으로 마이닝 작업을 하는 것이 중요한 요소로 여겨진다. 따라서 이러한 작업 환경에서는 최대 빈발 패턴 마이닝 기법의 적용이 효과적이다. 반면에, 작은 오차나 손실에도 막대한 영향이나 손해를 끼칠 수 있는 마이닝 환경에서는 최대 빈발 패턴마이닝 접근방법은 적합하지 않다. 이러한 경우에는 패턴의 압축효과는 최대 빈발 패턴 마이닝 기법에 비해 떨어지지만 패턴의 전환 과정에서 어떠한 손실도 야기하지 않는 닫힌 빈발 패턴 마이닝이 더욱 적합하다.

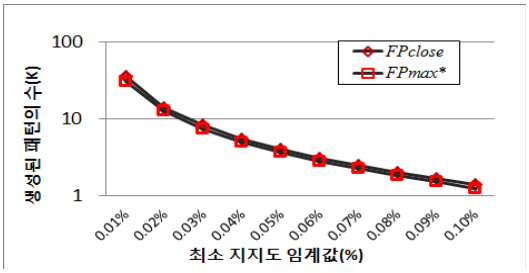
4. 성능 분석

4.1. 환경 설정

상기 패턴 압축 기법의 성능 평가 및 분석을 위해 동일한 구현 수준에서 C++로 작성된 두 알고리즘 FPclose와 FPmax* [4]에 대해 4GHz CPU, 16GB RAM, Windows 7 OS 환경에서 테스트를 진행했다. 또한 패턴 마이닝 분야에서 잘 알려진 실 데이터셋들 가운데 하나인 Connect 데이터셋과 Chain-store 데이터셋을 이용해 성능 평가를 진행했다. Connect 데이터셋은 밀집된 (dense) 특성을 갖는 데이터셋이며, 반면에 Chain-store는 희소한 (sparse) 특성을 갖는다. 상기 각각 상반된 특성을 갖는 실 데이터셋들에 대해 성능 평가를 진행함으로써, 각 패턴 압축 기법이 어떠한 특징과 효과를 보이는지 알 수 있다.



(그림 1) 패턴 생성 결과 (Connect)
(Figure 1) Pattern generation result (Connect)



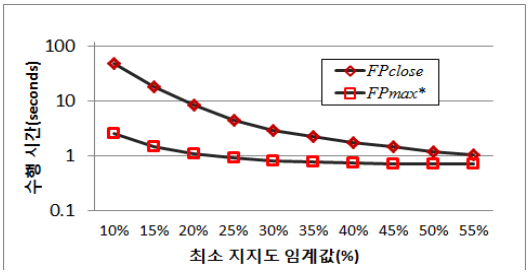
(그림 2) 패턴 생성 결과 (Chain-store)
(Figure 2) Pattern generation result (Chain-store)

4.2. 패턴 생성 결과 분석

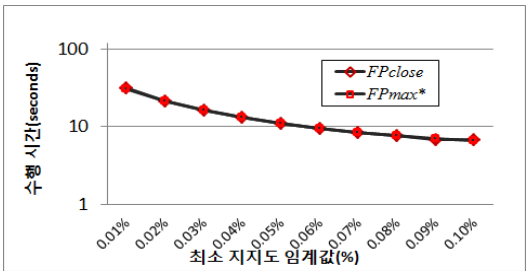
그림 1과 2는 Connect와 Chain-store 데이터셋에 대해 두 알고리즘이 각각 생성한 패턴의 수를 나타낸다. 그림 1에서 FPmax*의 경우, 최대성 속성에 의해 FPclose의 경우보다 일반적으로 더 적은 수의 패턴을 생성하기 때문에, 실 데이터 Connect에 대한 결과에서도 마찬가지로 모든 경우에서 더 적은 수의 패턴을 생성함을 알 수 있다. 더욱이, 주어진 최소 지지도 임계값이 낮아질수록, 그 차이는 더욱 벌어진다. 반면에, 그림 2에서는 이러한 최대성 속성에 의한 효과가 폐쇄성 속성과 비교해 별 차이가 없음을 볼 수 있다. 그 이유는 Chain-store와 같은 희소한 특성을 갖는 데이터셋에 대해서는 최대성에 의한 패턴 압축 효과가 상대적으로 떨어지기 때문이다. 위의 실험 결과들을 통해, 최대성에 의한 패턴 압축 효과는 Connect와 같은 밀집 데이터셋에 대해 더욱 좋은 성능을 보임을 알 수 있다.

4.3. 수행시간 결과 분석

상기 패턴 생성에 따른 결과 특성은 그림 3과 4에도

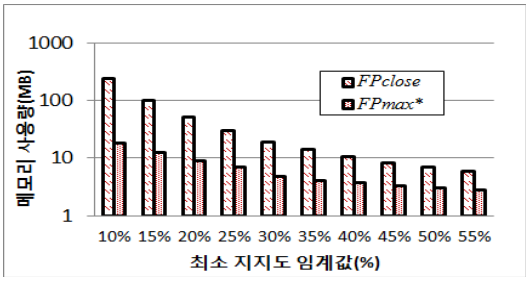


(그림 3) 수행 시간 결과 (Connect)
(Figure 3) Runtime result (Connect)

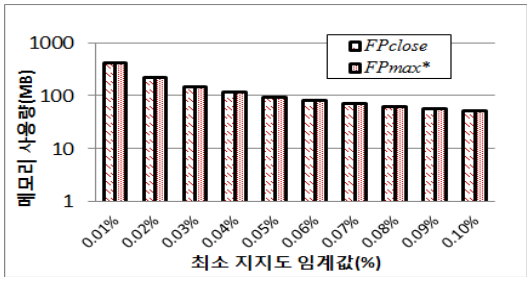


(그림 4) 수행 시간 결과 (Chain-store)
(Figure 4) Runtime result (Chain-store)

밀접한 영향을 끼친다. 알고리즘의 전체 수행시간 중 대부분은 마이닝을 위한 트리 자료구조를 생성하고 각 자료구조로부터 유효한 패턴을 찾는데 소모된다. 따라서 일반적으로 마이닝되는 패턴의 수가 적을수록 해당하는 마이닝 수행 시간 역시 빨라진다. 그림 3에서, Connect 데이터셋은 밀집한 특성을 가지므로 최대성 속성에 의한 패턴 압축 효과가 더욱 현저하게 나타나고, 따라서 생성되는 패턴의 수 역시 폐쇄성 속성을 고려한 마이닝 작업과 비교해 현저하게 작아 수행시간 결과 역시 상당한 차이를 보이게 된다. 특히 설정된 최소 지지도 임계값이 낮아질수록 그 차이는 더욱 벌어지게 된다. 그림 4의 Chain-store의 경우, 각각 폐쇄성과 최대성 속성을 고려한 두 알고리즘 FPclose와 FPmax*간에 수행시간의 차이가 거의 없음을 볼 수 있다. 희소한 특성을 갖는 데이터셋에서는 트렌젝션간에 아이템이 비슷한 정도가 낮으며, 트렌젝션의 길이가 상대적으로 짧기 때문에 상대적으로 긴 길이의 패턴이 생성될 가능성이 희박해진다. 따라서 그림 2에서 보이는 바와 같이 최대성을 고려하는 경우와 폐쇄성을 고려하는 경우에 각각 생성되는 패턴 수의 차이가 상당히 작기 때문에 그에 따라 해당하는 마이닝 수행시간 역시 거의 차이가 없는 수준으로 나타난다.



(그림 5) 메모리 사용량 결과 (Connect)
(Figure 5) Memory usage result (Connect)



(그림 6) 메모리 사용량 결과 (Chain-store)
(Figure 6) Memory usage result (Chain-store)

4.4. 메모리 사용량 결과 분석

다음의 성능 평가는 알고리즘의 메모리 사용량에 관한 것으로써, 그림 5와 6에서 보이는 바와 같이 데이터셋의 특성에 따라 상이한 성향의 결과가 나타났다.

밀집 데이터셋인 Connect 데이터셋의 경우에는 전체적으로 최대성 속성을 적용한 알고리즘인 FPmax*가 더 적은 메모리를 소모함을 알 수 있다. 특히 그 차이는 설정된 최소 지지도 임계값이 낮아질수록 커진다. 이러한 결과가 나타나는 이유는 임계값이 낮아짐에 따라 두 알고리즘 모두 더 많은 수의 패턴들을 생성하는 것은 동일하지만 최대성을 고려하는 경우 최대성 속성이 가지는 뛰어난 패턴 압축 효과로 인해 폐쇄성을 고려하는 FPclose에 비해 더 적은 수의 패턴을 추출하기 때문이다.

앞서 기술한 바와 같이, 최대성과 폐쇄성 각각을 고려하기 위해서는 마이닝을 위한 트리 구조 외에도 추가적인 자료구조가 필요하며, 마이닝 되는 최대 빈발 패턴 혹은 닫힌 빈발 패턴들이 많아질수록 해당 자료구조에 패턴 정보가 저장되기 때문에 사용되는 메모리 역시 커지게 된다.

반면에, 희소한 특성을 갖는 Chain-store데이터셋에 대해서는 Connect와는 다른 경향의 결과를 보인다. 그림 2에서 보이는 바와 같이 Chain-store 데이터셋에서 생성되는 패턴의 수는 최대성 속성을 고려하는 알고리즘과 폐쇄성 속성을 고려하는 알고리즘 간에 차이가 거의 없음을 알 수 있다. 따라서 그에 따라 해당 마이닝 작업을 위해 필요로 하는 메모리 자원 역시 두 알고리즘들 간에 차이가 거의 없게 된다. 이런 이유로 그림 6의 그래프의 결과처럼 설정된 최소 지지도 임계값의 값에 상관없이 두 알고리즘들이 모두 거의 비슷한 정도의 메모리 사용량 성능을 보인다.

5. 결 론

본 논문에서는 기존의 전통적인 빈발 패턴 마이닝 접근방법들이 갖는 한계점들 가운데 하나인 과도한 패턴 생성에 따른 연산의 오버헤드와 패턴 분석의 어려움을 극복하기 위해 제안된 대표 패턴 마이닝에 대한 특성 및 성능에 대한 분석을 진행하였다. 최대성을 고려한 패턴 압축 기법은 전반적으로 더 적은 수의 패턴을 생성함은 물론 더욱 빠른 수행시간과 적은 메모리 사용량을 보이는 반면, 폐쇄성을 고려한 패턴 압축 기법의 성능은 대체로 최대성 기법에 비해 떨어지지만 압축된 패턴의 복원 과정에 대한 무결성을 보장하기 때문에, 사용자의 사용 환경에 따라 두 가지 기법 모두 유용하게 활용될 수 있는 특성이 있다.

참 고 문 헌 (Reference)

[1] R. Agrawal, T. Imilienski, and A, Swami, “Mining association rules between set of items in large databases”, ACM SIGMOD, Vol.40, No.2, pp.207-216, 1993. http://dx.doi.org/10.1145/170036.170072

[2] J. Cai, X. Zhao, and Y. Xun, “Association rule mining method based on weighted frequent pattern tree in mobile computing environment”, International Journal of Wireless and Mobile Computing, Vol. 6, No. 2, pp. 193-199, 2013. http://dx.doi.org/10.1504/IJWMC.2013.054047

[3] G. Fang, Z. Deng, and H. Ma, “Network Traffic Monitoring Based on Mining Frequent Patterns”, Fuzzy Systems and Knowledge Discovery, Vol. 7, pp. 571-575, 2009. http://dx.doi.org/10.1109/FSKD.2009.444

[4] G. Granhne and J. Zhu, “Fast algorithms for frequent itemset mining using fp-trees”, IEEE Transactions on

Knowledge and Data Engineering, Vol.17, No.10, pp.1347-1362, 2005.
http://dx.doi.org/10.1109/TKDE.2005.166

[5] 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

[6] G. Lee, U. Yun, and K. Ryu, ″Sliding window based weighted maximal frequent pattern mining over data streams″. Expert Systems with Applications, Vol. 41, No. 2, pp. 694-708, 2014.
http://dx.doi.org/10.1016/j.eswa.2013.07.094

[7] G. Pyun, U. Yun, and K. Ryu, “Efficient frequent pattern mining based on Linear Prefix tree”, Knowledge Based Systems, Vol.55, pp.125-139, 2014.
http://dx.doi.org/10.1016/j.knosys.2013.10.013

[8] 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

[9] H. Ryang and U. Yun, ″Performance Analysis of Frequent Pattern Mining with Multiple Minimum Supports″, Journal of Internet Computing and Services, Vol. 14, No. 6, pp. 1-8, 2013.
http://dx.doi.org/10.7472/jksii.2013.14.6.01

[10] A. Sallaberry, N. Pecheur, S. Bringay, M. roche, and M. Teisseire, “Sequential patterns mining and gene sequence visualization to discover novelty from microarray data”, Journal of Biomedical Informatics, Vol.44, pp. 760-774, 2011.
http://dx.doi.org/10.1016/j.jbi.2011.04.002

[11] M.Y. Su, G.J. Yu, and C.Y. Lin, “A real-time network intrusion detection system for large-scale attacks based on an incremental mining approach”, Computers & Security, Vol. 28, No. 5, pp. 301-309, 2009.
http://dx.doi.org/10.1016/j.cose.2008.12.001

[12] U. Yun and E. Yoon, ″An Efficient Approach for Mining Weighted Approximate Closed Frequent Patterns Considering Noise Constraints″, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol. 22, No. 6, pp. 879-912, 2014.
http://www.worldscientific.com/doi/abs/10.1142/S0218488514500470

[13] U. Yun and G. Lee, ″A Weighted Frequent Graph Pattern Mining Approach considering Length-Decreasing Support Constraints″, Journal of Internet Computing and Services, Vol. 15, No. 6, pp. 125-132, 2014.
http://dx.doi.org/10.7472/jksii.2014.15.6.125

◐ 저 자 소 개 ◑

이 강 인 (Gang-in Lee)
2012년 충북대학교 컴퓨터공학전공 학사. (공학사)
2014년 세종대학교 대학원 컴퓨터공학 석사. (공학석사)
2014년~현재 세종대학교 대학원 컴퓨터공학 박사과정. (공학박사)
관심분야 : 데이터마이닝, 정보검색, 데이터베이스.
E-mail : ganginlee@sju.ac.kr

윤 은 일 (Un-il 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