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

길이에 따라 감소하는 빈도수 제한조건을 고려한 가중화 그래프 패턴 마이닝 기법☆

A Weighted Frequent Graph Pattern Mining Approach considering Length-Decreasing Support Constraints

윤 은 일[1*] 이 강 인[1]
Unil Yun Gangin Lee

요 약

대규모의 데이터베이스로부터 숨겨진 유용한 패턴 정보를 찾기 위해 빈발 패턴 마이닝이 제안된 이래로, 다양한 종류의 접근 방법들과 어플리케이션들이 연구되어 왔다. 특히, 빈발 그래프 패턴 마이닝은 계속해서 복잡해져 가는 최근의 데이터들을 효과적으로 다루기 위해 제안되었고, 이와 관련한 다양한 효율적인 알고리즘들이 연구되어 왔다. 그래프 데이터베이스로부터 얻을 수 있는 그래프 패턴들은 이를 구성하는 요소들에 따라 다른 중요도를 가지며 길이에 따라 다른 특성을 갖는다. 하지만, 전통적인 빈발 그래프 패턴 마이닝 접근 방법들은 이러한 문제들을 고려할 수 없다는 한계점을 지닌다. 즉, 기존의 방법들은 마이닝 과정에서 추출되는 그래프 패턴들의 길이에 상관없이 오직 하나의 최소 지지도 임계값만을 고려하고 이들의 가중치 요소들을 사용하지 않기 때문에, 실제적으로 쓸모없는 그래프 패턴들이 상당량 생성될 수 있다. 작은 수의 정점과 간선을 갖는 작은 그래프 패턴들은 이들에 대한 가중화 지지도 값이 상대적으로 높을 때 흥미로운 특성을 갖는 경향이 있는 반면, 많은 정점과 간선을 갖는 큰 그래프 패턴들은 비록 가중화 지지도 값이 상대적으로 낮을지라도 흥미로운 특성을 가질 수 있다. 이러한 이유로, 본 논문에서는 길이에 따라 감소하는 지지도 제한조건을 고려한 가중치 기반의 빈발 그래프 패턴 마이닝 알고리즘을 제안한다. 본 논문에서 제공되는 총체적인 실험 결과들은 제안되는 방법이 기존의 최신 그래프 마이닝 알고리즘과 비교하여 패턴 생성, 수행시간, 그리고 메모리 사용량 측면에서 더욱 뛰어난 성능을 보장함을 보인다.

☞ 주제어 : 길이 감소 지지도 제한조건, 가중화 빈발 패턴 마이닝, 그래프 패턴, 데이터 마이닝, 빈발 패턴 마이닝

ABSTRACT

Since frequent pattern mining was proposed in order to search for hidden, useful pattern information from large-scale databases, various types of mining approaches and applications have been researched. Especially, frequent graph pattern mining was suggested to effectively deal with recent data that have been complicated continually, and a variety of efficient graph mining algorithms have been studied. Graph patterns obtained from graph databases have their own importance and characteristics different from one another according to the elements composing them and their lengths. However, traditional frequent graph pattern mining approaches have the limitations that do not consider such problems. That is, the existing methods consider only one minimum support threshold regardless of the lengths of graph patterns extracted from their mining operations and do not use any of the patterns' weight factors; therefore, a large number of actually useless graph patterns may be generated. Small graph patterns with a few vertices and edges tend to be interesting when their weighted supports are relatively high, while large ones with many elements can be useful even if their weighted supports are relatively low. For this reason, we propose a weight-based frequent graph pattern mining algorithm considering length-decreasing support constraints. Comprehensive experimental results provided in this paper show that the proposed method guarantees more outstanding performance compared to a state-of-the-art graph mining algorithm in terms of pattern generation, runtime, and memory usage.

☞ keyword : Length-decreasing support constraint, weighted frequent pattern mining, graph pattern, data mining, frequent pattern mining

[1] Dept. of Computer Engineering, Sejong University, Seoul, 143-747, Korea
* Corresponding author (yunei@sejong.ac.kr)
[Received 11 July 2014, Reviewed 23 July 2014, Accepted 22 September 2014]
☆ 이 논문은 2014년도 세종대학교 교내연구비 지원에 의한 논문임.

1. 서 론

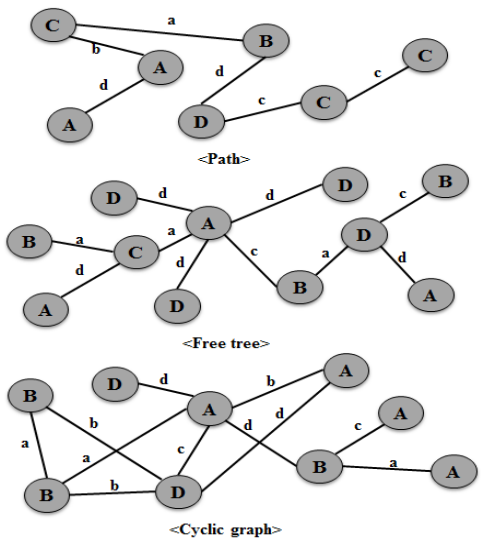
시간이 지남에 따라 점점 더 복잡해지고 대규모화 되어가는 데이터로부터 숨겨진 유용한 지식 또는 정보를 찾기 위해 데이터 마이닝의 개념이 제안된 이래로, 데이터 마이닝에 관한 다양한 접근 방법과 응용들이 연구되어 왔다. 또한 데이터 마이닝의 한 분야인 빈발 패턴 마이닝 [1, 3, 5]은 대규모의 데이터베이스로부터 유용한 정보를 패턴 형태로 제공하기 위한 기술이다. 최근에는 실세계의 다양하고 복잡한 데이터들을 효과적으로 표현 할 수 있는 그래프 데이터 구조를 효과적으로 마이닝하기 위한 방법으로써, 빈발 그래프 패턴 마이닝 [2, 4, 7, 8, 11]에 대한 연구가 활발히 진행되고 있다.

그러나 기존의 전통적인 빈발 그래프 패턴 마이닝은 다음과 같은 한계점을 갖는다. 그래프를 구성하는 각 요소는 서로 간에 각기 다른 중요도를 갖지만, 일반적인 빈발 그래프 패턴 마이닝 접근 방법은 그래프의 각 요소가 어떤 중요도를 갖는지에 관계없이 모든 요소가 동일한 중요도를 갖고 있다고 가정한 채로 마이닝 연산을 수행하기 때문에, 이를 통해 얻어진 마이닝 결과가 실세계의 특성을 효과적으로 반영하지 못하는 문제가 있다. 더욱이, 생성되는 그래프 패턴들이 비록 주어진 최소 지지도 임계값 이상의 지지도 값을 갖는다 할지라도, 각 패턴의 길이 특성에 따라 서로 다른 특성을 가질 수 있다. 즉, 작은 수의 정점과 간선을 포함하는 작은 규모의 그래프 패턴들은 이들에 대한 가중화 지지도 값이 상대적으로 높을 때 흥미로운 특성을 갖는 경향이 있는 반면에, 많은 수의 정점과 간선을 갖는 큰 규모의 그래프 패턴들은 비록 가중화 지지도 값이 상대적으로 작아도 흥미로운 특성을 가질 수 있다. 이러한 문제점들을 고려하기 위해, 본 논문에서는 길이에 따라 감소하는 지지도 제한조건을 고려한 가중치 기반의 빈발 그래프 패턴 마이닝 방법, *WEL-GMiner* (WEight and Length-decreasing support constraint-based frequent Graph pattern Miner) 을 제안함으로써, 기존의 접근방법보다 더욱 효율적으로 유용한 그래프 패턴 정보를 마이닝할 수 있다. 그래프 데이터베이스로부터 패턴을 마이닝하기 위해서는 전통적인 빈발 패턴 마이닝보다 더욱 복잡한 연산 과정이 필요하다. 특히 그래프의 중복 마이닝을 막는 동형판단 연산은 NP-hard 문제로써 연산의 오버헤드를 발생시킬 수 있다. 하지만, 본 연구에서는 생성되는 그래프 패턴의 수를 효과적으로 감소시킴으로써 이러한 연산의 수를 감소시킬 수 있으며, 이는 또한 수행시간과 메모리 자원 소모의 감소에 기여한다. 따라서 인터넷 컴퓨팅 환경과 같은 실시간의 즉각적인 처리를 필요로 하는 환경에서 더욱 효과적으로 활용될 수 있다.

2. 관련 연구

2.1. 빈발 그래프 패턴 마이닝

빈발 그래프 패턴 마이닝의 주요 목적은 그래프 데이터베이스로부터 최소 지지도 임계값 조건을 만족하는 모든 빈발 그래프 패턴을 마이닝 하는 것이다. 전통적인 빈발 패턴 마이닝과 빈발 그래프 패턴 마이닝의 가장 큰 차이점 중 하나는 오직 아이템만을 고려하는 전통적인 방법과 달리, 그래프를 구성하는 정점과 간선과 같은 더욱 복잡한 요소들을 고려하며, 유효한 그래프 패턴을 마이닝하기 위해서는 그래프의 동형판단 과정을 필요로 한다.



(그림 1) 그래프 형태의 예
(Figure 1) Example of graph types

gSpan [12], FFSM [6], Gaston [10] 등은 빈발 그래프 패턴 마이닝의 대표적인 알고리즘들로써, 주어진 그래프 데이터베이스들로부터 모든 빈발한 그래프 패턴을 마이닝한다. 특히 Gaston은 이러한 알고리즘들 가운데 비록 메모리를 더 소모하지만 가장 빠른 마이닝 수행 속도를 보장한다. 그림 1은 그래프 데이터베이스로부터 생성될 수 있는 그래프 패턴의 예를 보인다. 모든 그래프의 형태는 그림에서 보이는 경로 (path), 자유 트리 (free tree), 순환

그래프 (cyclic graph) 중 하나의 형태를 따르게 된다. Gaston은 이러한 그래프 형태 각각에 적합한 마이닝 기법을 상황에 따라 선택 해 그래프 마이닝 연산을 수행한다. 또한 그래프 데이터베이스 스캔 횟수를 줄이고, 더욱 빠른 속도로 마이닝을 수행하기 위해 제안된 자료구조인 embedding list를 사용한다. 그러나 Gaston과 같은 전통적인 빈발 그래프 마이닝 기법들은 그래프를 구성하는 요소들 각각의 중요도를 고려하지 못하는 한계점이 있고, 생성되는 그래프 패턴들의 길이에 따른 특성들을 고려하지 못하는 문제점을 갖고 있다.

2.2. 빈발 패턴 마이닝의 길이 감소 지지도 제한조건

전통적인 빈발 패턴 마이닝 분야에서 생성되는 패턴들은 단순히 주어진 최소 지지도 임계값보다 크거나 갖은 지지도 값을 갖는 것들이다. 하지만, 실제적으로는, 작은 수의 아이템들을 갖는 소규모의 패턴들이 지지도 값이 비교적 높을 때 의미 있는 정보를 포함하는 경향을 보이며, 반면에 다수의 아이템들을 포함하는 대규모의 패턴들은 해당하는 지지도 값이 비교적 작을지라도 의미 있는 정보를 내포하고 있을 수 있다. 따라서 빈발 패턴 마이닝 분야에서 이러한 특성을 고려하면서 마이닝 연산을 수행하기 위해 다양한 연구들 [13, 14]이 수행되어 왔다. LPMiner/SLPMiner [13]는 길이에 따라 감소하는 지지도 제한조건을 고려해 트렌젝션 또는 순차 트렌젝션 데이터베이스에서 의미 있는 패턴을 추출하기 위한 기법들이다. 하지만, 상기 알고리즘들은 모두 단순하게 아이템들로 구성된 데이터베이스를 대상으로 하는 마이닝 방법들이기 때문에, 그래프 데이터베이스를 대상으로는 마이닝 연산을 수행할 수 없다는 한계점을 갖는다. FGM-LDSC [9]는 길이 감소 지지도 제한조건을 그래프 패턴 마이닝에 적용한 알고리즘이다. 하지만 FGM-LDSC는 그래프의 빈도수 요소만을 고려할 뿐, 생성되는 그래프 패턴 내의 요소 각각에 대한 중요도를 고려할 수 없다는 점에서 한계점을 갖는다.

3. 가중치 요소와 길이에 따라 감소하는 지지도 제한 조건을 고려한 빈발 그래프 패턴 마이닝

3.1. 사전 지식

본 절에서는, 제안되는 알고리즘과 관련 기법들을 기술하기에 앞서, 제안되는 사항들에 대한 이해를 돕기 위해 그래프 패턴에 대한 정의와 개념을 포함하는 사전 지식에 대해 기술한다. 그래프는 다수의 정점과 간선들로 이루어진 집합을 말하며, 하나의 간선에는 두 개의 정점이 연결된다.

정의 1. (빈발 그래프 패턴) 어떤 그래프 패턴, G가 있고, 그래프 데이터베이스, $GDB = \{T_1, T_2, ..., T_k\}$ (각 T는 하나의 그래프 트렌젝션) 가 있을 때, 이 패턴의 지지도는 다음과 같은 수식을 통해 구할 수 있다.

$$E(G, T_i) = \begin{cases} 1, if\ G \in T_i \\ 0, otherwise \end{cases}$$

$$SUP(G) = \Sigma_{Gi \in GDB}\ E(G, T_i)$$

사용자로부터 특정된 최소 지지도 임계값을 δ 라고 하자. 그러면, G는 다음과 같은 조건을 만족할 때, 빈발 그래프 패턴으로 정의된다.

$$SUP(G) \geq \delta$$

결과적으로, 빈발 패턴 마이닝이란 위와 같은 조건을 만족하는 모든 그래프 패턴들을 추출하는 일련의 작업이라 할 수 있다.

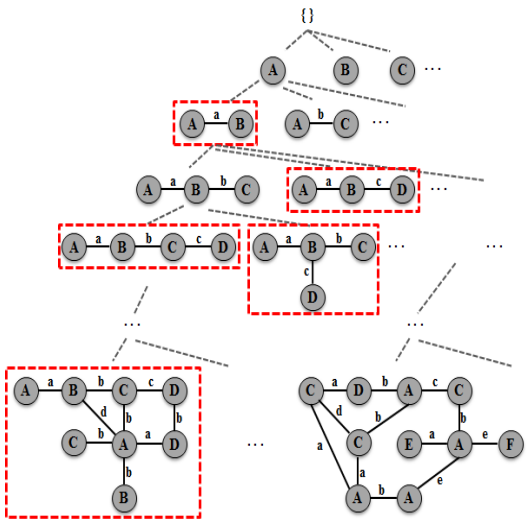
3.2. 그래프 패턴의 길이에 따른 가중화 빈발 그래프 패턴 생성

전통적인 빈발 패턴 마이닝의 상기 패턴 길이에 따른 특성은 빈발 그래프 패턴 마이닝에서도 동일하게 적용될 수 있다. 더욱이, 실세계로부터 얻어지는 그래프 데이터베이스 상에서는, 그래프를 구성하는 요소들은 각기 다른 중요도를 가지기 때문에 이에 대한 고려 역시 필요하다. 특히, 간선은 그래프 패턴의 특성을 파악하고 동형판단을 하는데 이용되는 중요한 요소로써, 본 논문에서는 그래프의 요소들 가운데 간선의 가중치를 고려함으로써, 정점 간에 상호 연결 정도에 있어서 각기 다른 중요도를 고려 할 수 있도록 한다. 가중치를 고려한 가중화 빈발 그래프 패턴의 정의는 다음과 같다.

정의 2. (가중화 빈발 그래프 패턴) 어느 한 그래프 패턴, G에 대해 정점, 간선, 간선의 가중치 정보를 각각 $V = \{v_1, v_2, ..., v_n\}$, $E = \{e_1, e_2, ..., e_m\}$, $W = \{w_1, w_2, ..., w_m\}$ 로 나타내면, G에 대한 가중화 지지도 값은 다음과 같은 수식에 의해 계산될 수 있다.

$$W(G) = \frac{\sum_{i=1}^{n} w_i}{m}$$

$$WSUP(G) = SUP(G) * W(G)$$



(그림 2) 그래프 확장의 예
(Figure 2) Example of graph extension

여기서 W의 각 요소, 즉, 각 간선 e_i에 대한 가중치 w_i는 0과 1사이의 값으로 정규화 된 값이다. 정규화의 범위는 사용자에 요구에 의해 자유롭게 바뀔 수 있다. 정규화를 하는 이유는 실세계의 다양한 데이터는 얻어지는 환경에 따라 다양한 가중치 값을 가질 수 있으며, 이 값들은 경우에 따라 아주 작을 수도, 혹은 클 수도 있고, 각기 다른 가중치 범위를 가질 수 있기 때문에 정규화를 통해 이러한 문제를 해결하고 더욱 정확한 가중치 값을 얻고자 함이다. 이 값이 최소 지지도 임계값, δ보다 크거나 같을 경우, 가중화 빈발 그래프 패턴이 된다.

그림 2는 패턴 확장 과정을 통해 가중화 그래프 패턴이 생성되는 예를 보여준다. 그래프 패턴은 그림에서 보이는 바와 같이 최초에 하나의 정점을 기준으로 하여 이에 새로운 정점과 간선이 결합하면서 경로 형태의 그래프가 만들어진다. 경로는 또한 같은 경로 형태로 확장될 수도 있고, 자유 트리 형태나 순환 그래프 형태로도 확장 가능하다. 일단 한번 확장되는 그래프 패턴이 자유트리의 형태를 가지면, 그 후에 가능한 확장은 같은 자유트리 구조나 순환 그래프 형태로 한정된다. 마찬가지로, 순환 그래프로 일단 한번 확장이 끝난 패턴은 같은 순환 그래프 형태의 그래프 패턴으로만 확장이 가능하다. 그림 2의 붉은 점선으로 둘러싸인 그래프 패턴들을 실제로 유용한 패턴으로 가정하자. 그러면 길이에 따른 그래프 패턴의 특성을 고려하지 않고 하나의 최소 지지도 제한조건만을 사용하는 전통적인 마이닝 기법은 다음과 같은 문제점을 갖는다. 첫째, 최소 지지도 임계값이 비교적 높게 설정되어 있는 경우, 짧은 길이의 그래프 패턴은 추출이 가능하지만, 길이가 긴 그래프 패턴은 일반적으로 짧은 길이의 패턴보다 낮은 지지도 값을 갖기 때문에 마이닝이 불가능하다. 반면에 최소 지지도 임계값이 낮게 설정될 경우, 비록 실제적으로 필요한 그래프 패턴 정보는 모두 추출할 수 있지만, 낮아진 임계값 때문에 그와 더불어 쓸모없는 패턴들까지 마이닝된다는 문제점이 있다.

정의 3. (그래프의 길이-감소 지지도 임계값) 어떤 그래프 데이터베이스, GDB가 있을 때, 이로부터 생성될 수 있는 모든 가능한 그래프 패턴의 길이 집합을 $L = \{l_1, l_2, ..., l_n\}$ $(l_i \geq l_j, 1 \leq i < j \leq n)$ 이라고 하자. 그러면 각 길이 l에 대한 길이-감소 지지도 임계값의 집합은 $MINSUP = \{minsup_1, minsup_2, ..., minsup_n\}$ $(minsup_i \geq minsup_j, 1 \leq i < j \leq n)$ 으로 표현된다.

3.3. 알고리즘의 정확성과 효율성을 고려한 그래프 패턴 프루닝

본 연구의 마이닝 과정을 통해 생성되는 그래프 패턴은 해당하는 길이의 $minsup$ 값보다 크거나 같은 가중화 지지도를 갖는다. 하지만 임계값 조건과 가중치에 의한 패턴의 사전 프루닝 작업은 안티모노톤 속성 (anti-monotone property)을 위배하여 심각한 패턴 손실을 유발할 수 있다. 즉, 길이가 k인 어떤 그래프 패턴 G가 있을 때, 비록 $WSUP(G) < minsup_k$로 G가 유효하지 않은 패턴이 될지라도, G의 다음 확장 그래프인 G'에 대해서는 G'의 가중화 지지도가 추가되는 가중치 요소에 따라 기존보다 더욱 커질 가능성이 있고 또한 G'의 길이에 해당하는 최소 지지도 임계값은 G의 값보다 같거나 작아지기 때문에 $WSUP(G') \geq minsup_{(k+1)}$의 조건을 만족할 수 있다. 따라서, 만약 현재는 유효하지 않은 패턴 G를 사전에 제거해버린다면 이로부터 생성되는 수많은 유효 패턴들 역시 애초에 마이닝될 수 없는 상태가 되어버리기 때문에 다음과 같은 추가적인 고려사항이 반드시 필요하다.

프루닝 조건 1. (최대 가중치에 의한 과추정 프루닝) 주어진 GDB에 속하는 모든 그래프의 요소들 가운데 가장 큰 가중치를 최대 가중치, $MaxW$로 설정 후 추출되는 그래프 패턴의 지지도와 이 값의 곱이 최소 지지도 임계값보다 작을 경우 영구적으로 프루닝한다.

프루닝 조건 2. (길이-감소 지지도 임계값의 최소치 기반 프루닝) $MINSUP$의 값들 중 가장 작은 값인 $minsup_n$

을 프루닝의 기준으로 하여, 만약 어떤 그래프 패턴의 지지도가 해당하는 길이의 $minsup_k$ ($1 \le k \le n$)보다 작을지라도 $minsup_n$보다 작지 않으면 프루닝하지 않는다.

프루닝 조건 3. (실제 가중화 지지도에 의한 프루닝) *MaxW*를 고려해 생성된 그래프 패턴은 과추정된 가중화 지지도 값을 갖기 때문에, 최종적으로 해당 패턴의 실제 가중화 지지도 계산 결과가 최소 지지도 임계값보다 작지 않을 경우에만 유효 패턴으로 추출된다.

위와 같은 프루닝 조건들을 고려함으로써, 마이닝과정에서의 효율성을 높일 수 있고 생성되는 패턴의 정확성을 높일 수 있다. 그 이유는 위의 조건들 모두가 안티모노톤의 속성을 위배하지 않는 범위 내에서 불필요하게 생성되는 패턴의 수를 최소로 줄여주기 때문이다. 즉, 프루닝 조건 1과 2를 통해 안티모노톤의 속성을 위배하지 않는 수준에서 불필요한 후보패턴의 수를 최소로 줄임으로써 마이닝 탐색공간을 최소화 해 효율성을 높이고, 이렇게 생성된 패턴들에 대해 프루닝 조건 3을 통해 실제 가중화 지지도를 판단함으로써, 실제적으로 유효한 가중화 빈발 그래프 패턴만을 선별하여 마이닝의 정확성을 보장한다.

```
Input: a graph database, GDB,
       a set of length-decreasing supports, MINSUP
Output: a set of weighted frequent graph patterns, S
Mine_graphs(GDB)
1. δ ⇐ calculate minsup_n from MINSUP ;
2. V ⇐ find all vertices such that support ≥ δ;
3. E ⇐ find all edges such that support * MaxW ≥ δ;
4. for each vertex, v, in V
5.    g ⇐ v   //g는 현재 확장되는 그래프 패턴
6.    E' ⇐ find edges that can be attached to v among E;
7.    Stmt ⇐ "path"  // Stmt는 현재 그래프의 상태
8.    S = S ∪ Expand_graphs(g, E', Stmt);
9. return S;
----------------------------
Extending_graph(g, E, Stmt)
1.   for each edge, e in E
2.      if Stmt = "path" or "free tree"
3.         g' ⇐ g ∪ e ∪ v;  // v는 현재 e에 함께
                              연결되어 있는 정점
4.      else g' ⇐ g ∪ e;  // 여기서 e는 순환 간선
     //end for
5.   l ⇐ length of g';
6.   minsup_l ⇐ MINSUP(l);
7.   if SUP(g') * MaxW ≥ δ
8.      if  WSUP(g') ≥ minsup_l
9.         S = S ∪ g' ;
10.     else delete g' ;
11. else delete g' and goto line 1 with the next e ;
12. E' ⇐ find edges that can be attached to g' ;
13. Stmt ⇐ current graph state of g' ;
14. S = S ∪ Expand_graphs(g', E', Stmt);
15. return S;
```

(그림 3) WEL-GMiner 알고리즘
(Figure 3) WEL-GMiner algorithm

3.4. *WEL-GMiner* 알고리즘

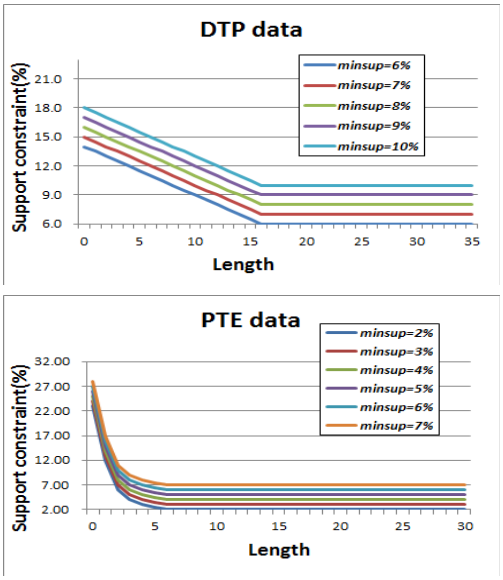
제안되는 알고리즘은 그림 3에서 보이는 바와 같이 수행된다. 먼저 주어진 그래프 데이터베이스로부터 유효한 간선과 정점을 찾은 후, 각 정점을 시작으로 그래프 확장 연산을 통해 길이에 따라 감소하는 지지도 제한조건을 고려한 가중화 빈발 그래프 패턴을 마이닝한다.

예제 1. 3개의 그래프 트렌젝션 {A-a-B}(path), {A-a-B-b-C}(path), {A-a-B-b-C (B-c-C)} (free tree)로 구성된 GDB가 있다고 가정하자. 여기서 세 번째 트렌젝션은 정점 B가 두 개의 정점 C와 연결되어 있는 형태이다. 각 정점의 지지도는 A:3, B:3, C:2, 간선의 지지도와 가중치는 a:3:0.8, b:2:1.0, c:1:0.5, *MINSUP* = {1.8(60%), 1.5(50%), 1.2(40%)}, $minsup_n$ = 1.2, *MaxW* = 1.0이라고 하자. 그러면 간선 c의 최대 가중화 지지도는 $minsup_n$ 보다 작기 때문에 사전에 프루닝된다. 현재의 *GDB*에서 정점 A로부터 생성될 수 있는 그래프 패턴은 A-a-B가 있고 이 패턴의 *WSUP*은 2.4로 길이 1에 해당하는 최소 지지도인 1.8보다 높기 때문에 유효한 패턴으로 마이닝되고 최대 가중화 지지도는 3이기 때문에 프루닝 또한 되지 않는다. A-a-B로부터 생성될 수 있는 패턴은 A-a-B-b-C와 A-a-B-c-C가 있지만 이 중 A-a-B-c-C은 간선 c가 프루닝되었기 때문에 생성되지 않는다. A-a-B-b-C의 최대 가중화 지지도는 2로 $minsup_n$보다 작지 않기 때문에 프루닝되지 않고 WSUP 역시 해당하는 길이의 최소 지지도 임계값 1.5보다 크기 때문에 유효한 패턴 결과로써 마이닝된다. 같은 방법으로 모든 패턴 확장에 대해 위와 같은 연산을 진행하면 최종적으로 길이에 따라 감소하는 빈도수 제한조건을 고려한 가중화 그래프 패턴들을 얻을 수 있다.

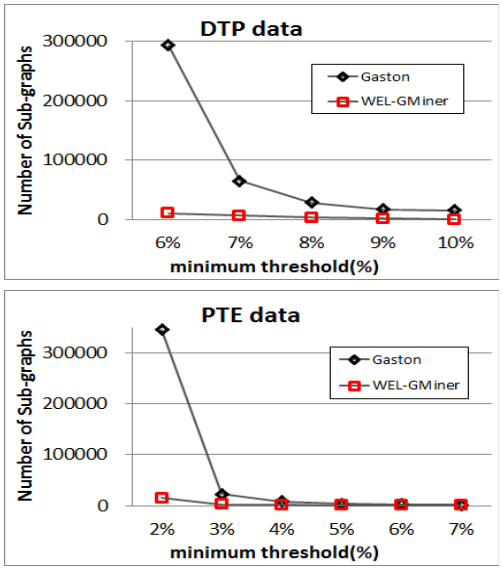
4. 성능평가

4.1. 환경 설정

본 절에서는, 제안되는 알고리즘, WEL-GMiner와 최신의 그래프 마이닝 알고리즘의 비교 및 분석을 통해 본 알



(그림 4) 길이 감소 지지도 임계값
(Figure 4) Length-decreasing support threshold



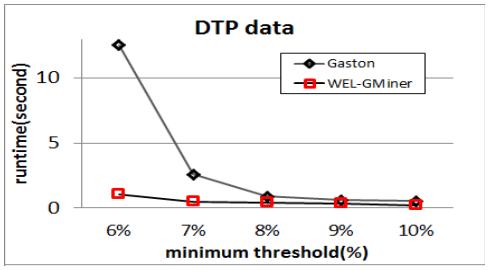
(그림 5) 패턴 생성 결과
(Figure 5) Pattern generation result

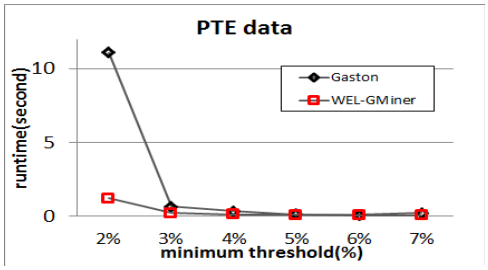
고리즘의 효율성을 패턴 생성, 수행시간, 메모리 소모량 측면에서 입증한다. 알고리즘들은 모두 C++로 구현되었으며, 3.33GHz CPU, 3GB RAM, Windows 7 OS의 PC환경에서 평가되었다. 사용되는 데이터셋은 화합물 정보를 담고 있는 DTP와 화학 정보를 담고 있는 PTE 데이터셋이며, 실세계로부터 얻어진 실제 데이터셋이다 [10]. 데이터셋의 가중치 설정은 기존의 가중치 기반 마이닝 방법 [14]과 마찬가지로 랜덤하게 설정되었으며, 설정 범위는 0.5~0.8이다. 그림 4는 각 데이터셋에 대한 길이 감소 지지도 임계값 설정을 나타낸다. 그림의 각 그래프의 minsup은 Gaston 알고리즘에 설정된 최소 지지도 임계값에 대한 WEL-GMiner의 길이 감소에 따른 지지도 임계값의 집합, MINSUP의 설정 상태를 말한다. 여기서 x축은 각 그래프 패턴의 길이를 말하고 y축은 길이에 해당하는 최소 지지도 설정 값을 말한다. 즉, 그림 4의 DTP에 대한 그래프에서 minsup=10%에 해당하는 그래프는 Gaston에 대한 최소 지지도 임계값이 10%로 설정되었을 때 WEL-GMiner는 길이가 1인 그래프에 대해서는 최소 지지도 임계값이 18%로 설정되고, 이후로 그림에서 보이는 바와 같이 길이가 늘어남에 따라 선형으로 감소하며 길이가 15 이후에는 일정한 최소 지지도 임계값 (10%)을 유지함을 보여주고 있다.

4.2. 패턴 생성과 수행시간

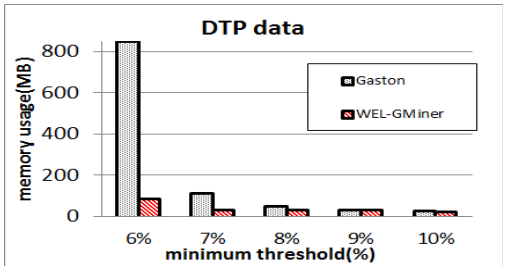
그림 5는 알고리즘들의 패턴 생성 결과를 나타낸다. 그림의 x축은 그림 4의 각 그래프에 대응되는 것으로써, 예를 들어 그림 5 DTP 데이터에서 최소 지지도 임계값이 10%일 때 Gaston은 모든 길이의 그래프 패턴을 마이닝함에 있어 10%의 동일한 최소 지지도 임계값을 가지고 마이닝하고, 제안되는 알고리즘은 그림 4에서 보이는 해당 *MINSUP* 정보를 바탕으로 마이닝을 수행함을 의미한다. 최소 지지도 임계값이 낮아질수록 두 데이터셋 모두에서 생성되는 패턴의 수는 증가한다. 그러나 본 연구의 알고리즘은 그림 4에서처럼 길이가 증가할수록 감소하는 지지도 임계값을 사용할 뿐만 아니라, 그래프 패턴의 가중치 요소를 고려함으로써, 더욱 큰 중요도를 갖는 가중화 빈발 그래프 패턴들만을 마이닝한다. 따라서 그림에서 보이는 바와 같이 비교대상 알고리즘에 비해 더 적은 수의 유용한 패턴들을 마이닝할 수 있게 된다.

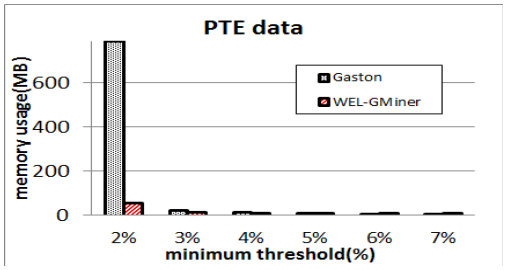
불필요하게 생성되는 패턴수를 감소시킴으로써, 그래프 패턴을 마이닝하는데 소요되는 그래프 동형판단과 같은 복잡한 연산에 대한 오버헤드를 줄일 수 있으며, 그 결과 그림 6에서 나타나는 바와 같이 필요로 하는 수행시간 역시 짧아짐을 알 수 있다. 특히 최소 지지도 임계값이 낮아질수록, 제안되는 알고리즘과 비교대상의 격차가 더욱 커짐을 알 수 있다.





(그림 6) 수행시간 결과
(Figure 6) Runtime result





(그림 7) 메모리 사용량 결과
(Figure 7 Memory usage result

4.3. 메모리 사용량

그래프 패턴의 가중치를 고려한 프루닝 효과와 길이에 따라 감소하는 지지도 임계값에 따른 프루닝 효과는 수행시간의 효율성 향상뿐만 아니라 알고리즘의 메모리 사용량에도 큰 기여를 한다. 모든 빈발 그래프 패턴을 마이닝하는 Gaston 알고리즘은 제안되는 WEL-GMiner에 비해 더 많은 그래프 확장 연산을 수행해야만 하고, 그 과정에서 그래프 확장에 필요한 정보를 유지해야 하며, 순환 그래프의 동형판단 과정에서는 추출되는 순환 그래프의 최소 스패닝 트리형태의 정보를 가지고 있어야 하기 때문에, 그림 7에서 보이는 바와 같이 전체적으로 더 많은 메모리를 소모한다. 반면에 본 연구의 알고리즘은 1차적으로 그래프 데이터베이스에서 가중화 빈발 그래프 패턴을 생성할 수 없는 요소를 미리 프루닝하며, 각 길이에 따라 감소하는 지지도 임계값 제한조건을 적용하기 때문에, 위와 같은 연산의 양이 Gaston에 비해 작다. 최소 지지도 임계값이 낮아질수록 두 알고리즘간의 메모리 사용량 격차는 더욱 급격히 커진다.

5. 결 론

본 논문에서는 전통적인 빈발 그래프 패턴이 갖는 문제인 그래프 패턴을 구성하는 요소들의 각기 다른 중요도를 고려하지 못하는 문제와 더불어, 그래프 패턴의 길이에 따라 상이해지는 특성을 고려하지 못하는 문제를 해결하고자, 길이에 따라 감소하는 지지도 제한조건과 그래프 패턴 내의 가중치 요소를 고려해 가중화 빈발 그래프 패턴을 마이닝하기 위한 방법을 제안했다. 본 연구의 제안되는 기법은 상기 제한조건을 통해 기존보다 더욱 유용한 그래프 패턴 결과를 얻을 수 있을 뿐만 아니라, 강력한 프루닝 효과로 인해 마이닝 연산의 효율성을 향상시킬 수 있었다. 추후 연구로써, 본 연구의 기법은 또한 동적인 데이터베이스를 대상으로 하는 실시간 마이닝이나 스트림 마이닝 분야에도 효과적으로 접목될 수 있으며, 이를 통해 동적 그래프 마이닝 관련 기법의 성능을 향상시킬 수 있을 것으로 예상한다. 또한 본 연구의 한계점으로써, 유용한 그래프 패턴 결과를 얻기 위해서는 유저가 각 그래프의 길이에 대해 적절한 임계값 설정을 찾아야 하는 문제점이 있는데, 추후 연구로써 어떠한 임계값 설정도 없이 상위 k개의 패턴 결과를 마이닝할 수 있는 상위-k 패턴 마이닝의 특성을 본 연구에 효과적으로 적용하는 방안에 대한 연구를 진행하여 이러한 문제점을 해결하고자 한다.

참 고 문 헌 (Reference)

[1] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," in Proc. of the 20th Int'l Conf. on Very Large Data Bases (VLDB), pp. 487-499, 1994.

[2] A. Bifet, G. Holmes, B. Pfahringer, and R. Gavalda, “Mining Frequent Closed Graphs on Evolving Data Streams”, in Proc. of the 17th ACM SIGKDD Int’l Conf. on Knowledge Discovery and Data Mining, pp. 591-599, 2011.

[3] A.Y.R. González, J.F.M. Trinidad, J.A. Carrasco-Ochoa, and J. Ruiz-Shulcloper, ″Mining frequent patterns and association rules using similarities,″ Expert Systems with Applications, Vol. 40, No. 17, pp. 6823-6836, 2013.

[4] S. Gunnemann and T. Seidl, “Subgraph Mining on Directed and Weighted Graphs”, in Pro. of the 14th Pacific-Asia Conf. on Knowledge Discovery and Data Mining, pp. 133-146, 2010.

[5] J. Han, J. Pei, and Y. Yin, ″Mining Frequent Patterns without Candidate Generation,″ in Proc. of the 2000 ACM SIGMOD Int’l Conf. on Management of Data, pp. 1-12, 2000.

[6] J. Huan, W. Wang, and J. Prins, “Efficient Mining of Frequent Subgraphs in the Presence of Isomorphism”, in Proc. of the 3rd IEEE International Conf. on Data Mining, pp. 549-552, 2003.

[7] Y. Jia, J. Zhang, and J. Huan, “An efficient graph-mining method for complicated and noisy data with real-world applications”, Knowledge Information Systems, vol. 28, no. 2, pp 423-447, 2011.

[8] C. Jiang, F. Coenen, and M. Zito, “Frequent Sub-graph Mining on Edge Weighted Graphs” in Proc. of the 12th int’l conf. on Data warehousing and knowledge discovery, pp. 77-88, 2010.

[9] G. Lee and U. Yun, “Frequent Graph Pattern Mining with Length-Decreasing Support Constraints”, Multimedia and Ubiquitous Engineering, pp. 185-192, 2013.

[10] S. Nijssen and J.N. Kok, “The Gaston Tool for Frequent Subgraph Mining”, Electronic Notes in Theoretical Computer Science, vol. 127, no. 1 pp. 77-87, 2005.

[11] L.T. Thomas, S.R. Valluri, and K. Karlapalem, “MARGIN: Maximal frequent subgraph mining”, Transactions on Knowledge Discovery from Data. vol. 4, no. 3, pp. 10:1-42, 2010.

[12] X. Yan and J. Han, “gSpan: graph-based substructure pattern mining”, in Proc. of the 2002 IEEE Int’l Conf. on Data Mining, pp. 721-724, 2002.

[13] M. Seno, and G. Karypis, “Finding frequent patterns using length-decreasing support constraints”, Data Mining and Knowledge Discovery, vol. 10, no. 3, pp. 197-228, 2005.

◐ 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

이 강 인 (Gangin Lee)

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