韓國컴퓨터情報學會 論文誌
第14卷 第10號, 2009. 10.

2009-14-10-5-1

ㅎ그래프 마이닝에서 그래프 동형판단연산의 향상기법

노 영 상*, 윤 은 일**, 김 명 준***

Improved approach of calculating the same shape in graph mining

YoungSang No*, Unil Yun**, Myung Jun Kim***

요 약

그래프마이닝에서 그래프패턴의 동형판단문제는 지수함수적 계산시간을 요구하기 때문에 그래프마이닝의 전체 수행시간에서 동형판단 연산이 차지하는 비율이 매우 높다. 그러므로 그래프마이닝 알고리즘은 그래프동형판단을 최대한 효율적으로 할 필요가 있다. 본 논문은 그래프마이닝에서 빠른 수행시간을 보이는 gaston 알고리즘의 동형판단효율성을 증가시켜 수행시간을 평가해 보았으며, 제시한 방법으로 인해 더욱 향상된 성능을 보인다.

Abstract

Data mining is a method that extract useful knowledges from huge size of data. Recently, a focussing research part of data mining is to find interesting patterns in graph databases. More efficient methods have been proposed in graph mining. However, graph analysis methods are in NP-hard problem. Graph pattern mining based on pattern growth method is to find complete set of patterns satisfying certain property through extending graph pattern edge by edge with avoiding generation of duplicated patterns. This paper suggests an efficient approach of reducing computing time of pattern growth method through pattern growth's property that similar patterns cause similar tasks. we suggest pruning methods which reduce search space. Based on extensive performance study, we discuss the results and the future works.

▸ Keyword : 서브 그래프 마이닝 (Subgraph mining), 패턴 확장 (pattern extension), 동형 그래프 동형 판단 (the same graph shape estimation)

• 제1저자 : 노영상 교신저자 : 윤은일
• 투고일 : 2009. 07. 22, 심사일 : 2009. 08. 12, 게재확정일 : 2009. 09. 15.
* 충북대학교, 컴퓨터공학 석사 ** 충북대학교 전자정보대학 컴퓨터전공 조교수
*** 충북대학교 전자정보대학 컴퓨터전공 교수
※ "이 논문은 2009년도 충북대학교 학술연구지원사업의 연구비지원에 의하여 연구되었음 (This work was supported by the research grant of the Chungbuk National University in 2009)"

Ⅰ. 서 론

데이터마이닝 알고리즘들 [1, 3, 13, 14]은 비즈니스나 여러 과학분야 나 사회분야의 데이터들을 분석하는데 폭넓게 사용되고 있으며, 효과적인 사용으로 인해 데이터마이닝의 인지도는 점점 더 높아지고 있다. 데이터마이닝에서 그래프마이닝 [은 매우 관심있는 주제가 된다. 그래프는 현실세계의 많은 것들이 모델링 할 수 있기 때문이다. 그래프마이닝은 아이템셋 마이닝과는 다르게 그래프매칭기법의 효율성을 매우 중요하게 여긴다. 그래프의 동형판단이 지수수함수적 계산시간을 요구하기 때문이다. 그래프마이닝 작업의 전체 소요시간에서 그래프 동형판단이 차지하는 비율은 매우 높다. 그렇기 때문에 그래프마이닝 알고리즘은 그래프 동형판단의 문제를 최대한 효과적으로 해결해야 한다.

그래프마이닝의 효과적인 패턴매칭을 위하여 패턴확장기법(pattern-growth)을 사용한다[5, 7, 8, 10, 11, 12]. 패턴확장기법은 빈발하는 패턴으로부터 빈발하는 간선만을 하나씩 늘려가면서 모든 패턴을 찾아내는 방법이다. 하지만 단순이 빈도수에 의지한 확장은 중복된 그래프패턴을 생성해 낸다. 그렇기 때문에 패턴이 확장되었을 때, 확장된 패턴의 중복성 판단을 반드시 해주어야 한다. 그래프패턴의 중복성 판단은 아이템셋마이닝처럼 찾아진 패턴들과의 비교를 통하지는 않는다. 그래프마이닝에서는 먼저 모든 그래프를 고정된 방식으로 표현할 수 있는 정규형을 정해둔다. 그리고 패턴 생성된 패턴이 정규형일 때에만 이 패턴을 찾아진 패턴으로 인정하고 그 외의 경우에는 중복된 패턴으로 생각하여 이를 폐기함으로써 중복성 판단을 효율적으로 수행한다. 본 연구에서는 패턴확장 기법은 분석하여 유사한 작업들이나 좀 더 효율적인 운영이 필요한 부분을 발견하고 더 빠른 수행시간을 개선하였다.

Ⅱ. 관련 연구

처음 그래프마이닝에서도 아이템셋마이닝처럼 후보자패턴를 생성한 뒤 그 빈도수를 계산하는 방법을 사용했다[6]. 그러나 그래프동형판단문제는 NP-hard문제에 속하기 때문에 후보자를 만드는 일이나, 만들어진 후보자패턴의 빈도수를 계산하는 일이 매우 비효율적으로 진행된다. 그래프동형판단을 효과적으로 하는 방법은 패턴확장기법 [5, 7, 8, 10, 11, 12]을 활용하는 것이다. 패턴확장기법은 패턴을 확장해가면서 확장된 패턴으로 생성되는 패턴을 먼저 찾는 방식인데, 이렇게 하면 확장된 패턴의 동형판단문제를 많이 생략할 수 있기 때문에 그래프마이닝에 매우 적합한 방법이라 할 수 있다. 하지만 패턴확장기법은 빈도수에 의지하여 모든 가능한 간선들을 모두 확장하기 때문에 중복된 패턴을 반복적으로 생성하게 된다. 이를 효과적으로 그래프마이닝에 적용한 최초의 연구는 gspan 알고리즘[10]이다. gspan은 그래프의 모든 간선들에 깊이우선탐색의 우선순위를 정함으로써, 우선순위의 코드의 생성 및 비교를 통하여 이를 실현했다. 그래프 마이닝에 대한 성능을 향상시키는 연구가 지속적으로 되어왔고 좋은 성능을 가진 알고리즘들 [2, 4, 8, 9]이 제안되어 왔다. 이중 대표적인 그래프 패턴확장 알고리즘이 gaston [8] 이다. gaston은 패턴확장도중 생겨나는 모든 패턴들을 경로그래프(path), 자유트리그래프(tree), 순환그래프(cyclic graph)로 나누어 중복된 패턴의 확장을 최소화하는 좋은 성능을 보이는 대표적인 그래프 마이닝 알고리즘이다. 본 논문에서는 그래프동형판단을 최소화하는 방법 및 egaston 알고리즘을 제안하고 gaston[8]과 성능을 평가 및 분석한다.

Ⅲ. 문제 정의

3.1 그래프 및 서브그래프의 개념정의

그래프란 노드(vertices)와 간선(edges)으로 구성되어 있는 구조체를 말한다. 한 노드의 차수(degree)란 그 노드에 접한 간선의 개수를 말한다. 경로그래프(path)는 모든 노드들이 2차수를 가지며 양 끝의 두 노드만이 1차수를 가지는 그래프를 말한다. 즉 순환을 이루지 않는 간선들의 연결이다. 트리그래프란 순환을 이루지 않으면서 적어도 한 개 이상의 노드의 차수가 3차수 이상인 그래프를 말한다. 트리그래프는 개념적으로 루트트리와 자유트리로 부를 수 있는데, 루트트리는 어떤 특정 노드를 루트로 여기지만, 자유트리는 어떤 특정 노드를 루트로 여기지 않다. 그러므로 트리구조를 자유트리의 개념으로 다루는 것은 루트트리 다루는 것 보다 중복성 줄일 수 있다.

그래프는 간선의 방향성이 있느냐 없느냐에 따라서 유향그래프(directed graph)나 무향그래프(directed graph)로 구분될 수 있으며, 두 노드간의 간선이 한 개뿐인가 여러 개가 허용되는 가에 따라서 단순그래프(simple graph)나 다중그래프(multigraph)로 나눌 수 있다. 또한 각 요소(노드, 간선)들이 유일한 레이블이 유일한가 아니면 동일한 레이블을 여러 요소들이 가질 수 있는가, 노드들은 회기간선(self loop)을 허용 하는 가 안하는 가 등 여러 가지 방법으로 그래프의 종류를 구분 지을 수 있다. 본 논문에서는 여러 다른 노

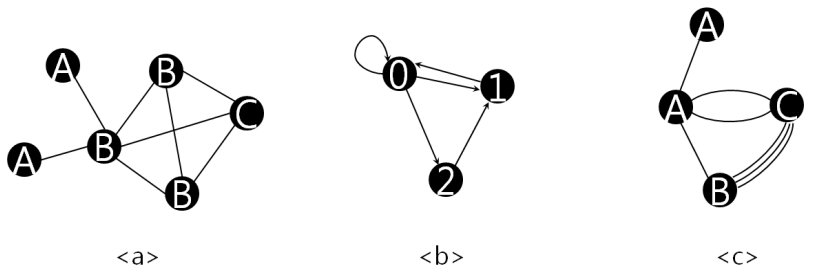

그림 1. 여러 그래프의 종류
Fig. Several types of graphs

드들이 동일한 레이블을 가질 수 있으며, 회기간선을 허용하지 않는 무향단순그래프를 대상으로 한다. 그림 1. 〈a〉를 말하는 것이며, 그림 1. 〈b〉는 유향그래프 이고, 그림 1. 〈c〉는 다중그래프이므로 그 대상이 아니지만, 조금의 변형으로 이러한 문제들도 유사하게 해결할 수 있다.

그래프는 G 또는 g로 표현하기로 정하며, 그래프를 구성하고 있는 노드의 집합을 V로 정하고, 간선의 집합을 $E = \{(v1,v2) \mid v1,v2 \in V \text{ and } v1 \neq v2\}$로 정한다. 간선에는 방향성이 없으므로, 두 간선 (v1,v2), (v2,v1)는 같은 간선으로 본다. 구성요소들이 가질 수 있는 레이블 집합은 L로 표현하며, 함수 l(V)나 l(E)는 각 간선이나 노드의 레이블을 반환한다. 두 개의 그래프 G1, G2가 있을 때, 그래프 G1의 내부구조 안에 그래프 G2의 구조와 동일한 구조의 그래프가 있을 때 그래프 G2를 G1의 서브그래프(subgraph)라 하고, G1 ⊃G2로 표현한다. 본 논문의 범위에서 동일한 구조란 노드와 간선의 구조가 정확하게 일대일로 상대되며, 또한 그 각각의 레이블이 동일한 경우를 말한다. 수식으로 표현했을 때, 그래프 G1=(V1, E1, L1), G2=(V2, E2, L2)에 대하여 G1 ⊃G2 라면, 다음의

두 식을 만족하는 일대일 함수 f(V1) → V2 가 존재한다.

⑴. $\forall v \in V1 \Rightarrow l(v) = l(f(v))$

⑵. $\forall (v1,v2) \in E1 \Rightarrow (f(v1),f(v2)) \in E2$ and $l1(v1,v2) = l2(f(v1),f(v2))$

G1⊃G2이고 G2⊃G1인 그래프를 동형그래프라고 한다. 경로(path)란 두 노드 사이에서 간선들을 통한 연결을 말하며, 하나의 그래프내에서 모든 노드들 사이에 경로가 존재할 경우 이를 연결그래프(connected graph)라고 한다.

3.2 문제의 정의

그래프 트랜잭션 데이터베이스는 하나의 트랜잭션의 하나의 연결그래프로 구성되어있다. 어떤 그래프 구조가 여러 개의 트랜잭션에서 나타난다면, 이 구조는 관심의 대상이 된다. 마이닝 작업은 일정 빈도수 이상의 이러한 패턴들을 모두 찾아내는 작업이다. 사용자가 관심을 가지는 일정한 빈도수(minimum frequency threshold)를 제시하면, 그것과 같거나 그 이상의 빈도수를 가지는 서브그래프패턴들을 모두 찾아내게 된다. 한 가지 주의 할 점은 어떤 서브그래프는 하나의 트랜잭션에서 여러 번에 걸쳐서 나타날 수 있으나 빈도수에는 단 한 번만 계산된다는 것이다.

IV. 동형 판단을 위한 성능 개선 기법

gaston알고리즘은 확장탐색영역을 위와 같이 3부분으로 나누어 다룸으로써 수행시간을 효과적으로 단축시켰다. 확장된 패턴이 간단한 구조일수록 중복성 판단을 효과적으로 할 수 있기 때문이다. 또한 트리그래프 패턴들은 그 패턴에 포함된 경로그래프들에 의해 정규화 될 수 있기 때문에, 이를 토대로 중복판단 없이도 중복생성회피를 할 수 있으므로 더욱 효율적이다. 순환그래프 역시도 그 패턴에 포함된 자유트리그래프로서 정규화 시킬 수 있다. 순환그래프들은 최소신장스패닝트리를 기준으로 정규형을 이룬다. 또한 최소신장스패닝트리는 레이블의 우선순위에 따른 깊이우선탐색의 정규형을 따른다. 어떤 순환그래프의 깊이우선순위에 따르는 정규형 최소신장스패닝트리는 하나만이 존재한다. 그러므로 순환그래프 내에 정규형 최소신장스패닝트리가 변하지 않도록 확장을 한다면 중복생성을 억제할 수 있다. 하지만 순환그래프를 중복판단 없이 중복생성회피할 수 있는 방법은 아직 없다. 그러므로 패턴을 확장할 때 마다 반복적으로 스패닝 트리 나열 연산을 하게 되는데 이는 매우 많은 시간을 소요한다.

본문에서는 이 순환그래프영역에서의 중복패턴으로의 확장을 좀 더 회피할 수 있는 방법에 대하여 설명하며, 스패닝트리나열연산을 좀 더 효과적으로 운영할 수 있는 방법을 제시한다.

4.1 효과적인 스패닝 트리 나열 연산

gaston 알고리즘은 모든 그래프들을 경로그래프 영역(path), 자유트리그래프영역(free tree), 순환그래프영역(cyclic graph)으로 나눈다. 자유트리는 특정 노드를 루트로 여기지 않는 트리구조를 말한다. 자유트리를 구성하는 경로들 중에 최대 길이

경로로써, 자유트리의 크기를 결정할 수 있다. 또한 이 최대 길이 경로의 우선순위를 정하여 가장 높은 우선순위를 가지는 경로를 유일하게 선정할 수 있다. 모든 자유트리들을 최대우선순위 최고길이경로와 동일한 경로그래프들로부터 확장함으로써, 자유트리의 중복생성을 억제할 수 있다. 경로그래프가 자유트리그래프로 확장되면 자유트리그래프에서는 더 이상 최대우선순위 최대경로가 변하지 않도록 제한함으로써 중복생성을 억제하게 된다. 이는 경로그래프들로 자유트리그래프 영역을 분할한 것이라고 할 수 있다. 자유트리에서 순환간선이 확장되면서부터 그래프는 순환그래프 영역으로 들어가게 된다. 순환그래프로 확장된 패턴은 더 이상 노드의 확장을 허용하지 않는다. 순환그래프에서 노드가 확장된 그래프는 그 순환그래프의 기본이 되는 트리에서 노드가 먼저 확장되고 난 뒤 순환간선이 확장되어 생기는 그래프와 동일하기 때문이다.

자유트리에서 순환간선이 확장되면 생성된 그래프의 최소신장스패닝트리는 처음의 자유트리와 동일해야 한다. 그리고 추가적으로 순환간선 확장되어도 역시 최소신장스패닝트리는 변하지 않아야 한다. 이는 경로그래프들이 자유트리영역을 분할하는 것처럼, 자유트리그래프들로서 순환그래프영역을 분할하는 것이다.

새롭게 생성된 순환그래프의 최소신장스패닝트리가 변하였는지 알기 위해서, 그래프의 모든 스패닝트리들을 모두 나열하면서 기본이 되었던 자유트리 모두 비교해 본다. 그러기 위해서는 모든 스패닝트리를 중복 없이 그리고 빠짐없이 나열할 수 있는 방법이 필요하다. 스패닝트리를 중복 없이 모두 나열하기 위하여 그래프 내의 모든 순환간선들의 조합을 고려해 본다. 중복된 조합을 피하기 위해서 모든 순환간선들의 순서를 정하여 하향식의 조합만을 허락한다. 순환간선의 우선순위는 패턴확장기법에 의해 생겨나는 노드의 순서를 통하여 생성될 수 있다. 간선의 양 끝단의 노드의 생성순서를 작은 값부터 순차로 연결하면 모든 간선의 우선순위 값을 정할 수 있다.

4.2 알고리즘

알고리즘 1은 모든 순환간선을 중복 없이 나열하는 방법이다. 스패닝트리의 정규형 우선순위를 비교하는 방법은 이 논문과 직적적인 관련이 없으므로 생략한다.

알고리즘 1. FindingHighOrderSpanningTree Algorithm

```
1: 입력 : G
2: 출력 : 더 높은 우선순위의 스패닝트리가 찾아졌으면 참
          그 외에는 거짓 값을 반환한다.
3: FindingHighOrderSpanningTree( G ) {
4:     if G가 트리라면 than
5:         G의 정규형트리표현을 기본자유트리와 비교한다;
6:         if G가 더 높은 우선순위를 가지는 트리라면 than
7:             return 참;
8:         else return 거짓;
9:     else
10:        G의 모든 간선에 순환간선인지 여부를 기록한다;
11:        for 모든 노드들에 대하여 do
12:            for 모든 간선들에 대하여 do
13:                if 마지막으로 지워진 간선보다 작으면 than
14:                    G에서 간선을 지운다;
15:                    결과 ← FindingHighOrderSpanningTree(G');
16:                    G에 간선을 넣는다;
17:                    if 결과 값이 참이면 then return 참;
```

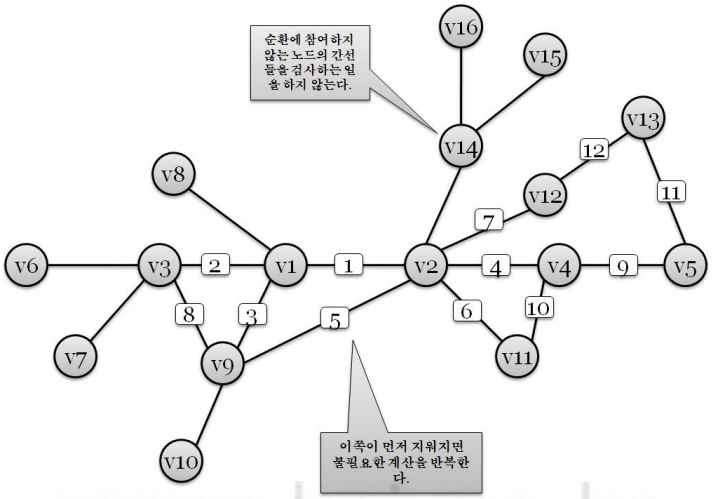

그림 2. 순환간선의 우선순위
Fig. 2. The priority order of cycle edge

알고리즘은 그래프를 입력 받으면, 더 높은 정규형 우선순위의 스패닝트리가 찾아질 때 까지 반복적으로 그리고 재귀적으로 모든 계속적으로 스패닝트리를 만들어 낸다. 만일 모든 스패닝트리를 나열할 때 까지 더 높은우선순위의 스패닝트리가 찾아지지 않았다면, 알고리즘은 거짓 값을 반환한다. 이는 주어진 그래프가 중복된 그래프가 아니라는 것을 의미 하게 된다. 알고리즘은 먼저 입력받은 그래프가 트리인지를 먼저 판단한다. 만일 트리라면 주어진 순환그래프의 기본이 되었던 자유트리와 우선순위를 비교한 뒤, 더 높은 우선순위의 트리라면 참값을 반환한다.(line 4-8) 만일 트리가 아니라면 모든 간선들을 탐색하여 간선의 순환여부를 확인한 뒤, 모든 가능한 스패닝트리를 다시 나열하기 위하여 순환간선을 차례대로 지우면서 알고리즘을 재귀적으로 호출한다.(line 9-17) 이때 중복없는 순환간선의 조합을 위하여, 가장 마지막에 지워진 간선보다 작은 순서값을 가지는 간선만을 선별적을 제거한다. (line 13)

4.3 알고리즘 성능 개선 예제

알고리즘 1은 순환간선의 중복된 조합을 생성하지 않기 위하여 더 낮은 순서 값을 가지는 간선만을 지우도록 했는데 이는 알고리즘이 재귀적인 방식으로 운영된 다는 것을 생각하면 매우 비효율적이다. 알고리즘의 운영 도중 어떠한 시점에, 마지막에 지워진 순환간선보다 더 높은 값을 가지는 순환간선들로 이루어진 순환이 존재한다면 이 순환은 그래프에서 지워지지 않을 것이고 이는 알고리즘이 최종적으로 트리를 만들지 못한다는 것이다. 그럼에도 알고리즘은 이와 같은 사실을 판단하지 못하므로, 마지막에 지워진 간선보다 작은 순환간선들의 모든 조합을 계속해서 만들어보고 결국에 모두 트리를 만들지 못하고 되돌아오게 된다. 각 단계에서 모든 순환간선을 판단하는 작업 또한 그래프에 남은 모든 간선의 수에 비례한 수행시간을 필요로 하며, 모든 간선들을 돌면서 지워야 하는 간선을 찾는 작업 또한 도든 간선의 수에 비례하기 때문이 매우 많은 계산의 낭비를 초례하게 된다. 그림2는 이와 같은 경우의 예를 보여준다. 노드의 생성순서에 조합을 통해 만들어진 간선의 순서는 하얀색 사각형 안에 값으로 표현되어 있다. 이 그래프에서 가장먼저 5번순환간선이 지워졌다고 가정하면 순환간선 6,7,12,11,9,10 으로 이루어진 순환은 끝까지 지워지지 않는다. 그럼에도 1,2,3,4 의 순환간선이 존재하므로, 이 순환간선들을 모두 조합하면서 계속 반복된다. 가장 높은 값을 가지는 순환간선을 가장 먼저 제거한다면 이와 같은 경우는 발생하지 않을 것이다. 가장 큰 값의 순환간선을 먼저찾기 위하여 알고리즘 1의 line 11, 12의 반복문의 노드나 간선의 탐색순서를 거꾸로 탐색할 수 있다. 물론 이는 완전히 순환간선의 순서와는 부합하지 않기 때문에 실험을 통하여 효과를 확인해 본다.

V. 성능평가

5.1 실험환경

성능의 변화를 알아보기 위하여 기본이 되는 gaston 알고리즘에 우선순위 하향식 방법을 적용하여 egaston-S를 만들었다. 실험환경은 core2X64, 2GB memory, Opensolarise10 에서 실험했다. 실험에 쓰인 그래프데이터 들은 PTE, DTP로서, PTE는 340개의 그래프 트랜잭션으로 이루어져 있으며, DTP는 422개의 트랜잭션으로 구성되어 있다. 두 데이터에서 최소빈도수임계값에 따라 찾아지는 패턴의 수는 표1, 표2와 같다.

표1. 패턴의 개수 (PTE)
Table 1. Number of patterns (PTS)

support / patterns	1.5%	1.8%	2.1%	2.4%	2.6%	2.9%
Path	2817	2234	1724	1381	1065	903
FreeTree	586148	284294	117654	47496	27809	19578
Cyclic Graph	132248	57951	17571	5743	3164	2277
Total	721213	344479	136949	54620	32038	22758

표2. 패턴의 개수 (DTP)

suppport / patterns	3.6%	4.3%	5.0%	5.7%	6.4%
Path	4383	3474	2894	2289	1904
FreeTree	22821527	1470771	672979	260874	75046
Cyclic Graph	9037845	434760	229419	66090	14436
Total	31863755	1909005	905292	329253	91386

5.2 성능분석

빈도수의 값이 작아질수록 찾아지는 패턴의 개수는 기하급수적으로 늘어난다. 빈도수가 낮아질수록 찾아지는 패턴에서 순환그래프의 비율이 증가는 것을 볼 수 있다. 그림3, 그림4는 이를 잘 보여주며, DTP데이터셋에서는 PTE데이터 셋 보다 순환그래프의 비율이 훨씬 높은 것을 볼 수 있다. 그림5, 그림6를 보면 찾아지는 패턴의 평균 크기 또한 훨씬 커지는 것을 볼 수 있다. 이는 더 작은 빈도수에서 제시된 방법의 효율성이 더욱 잘 나타날 것을 말해주며, PTE 데이터셋에서 보

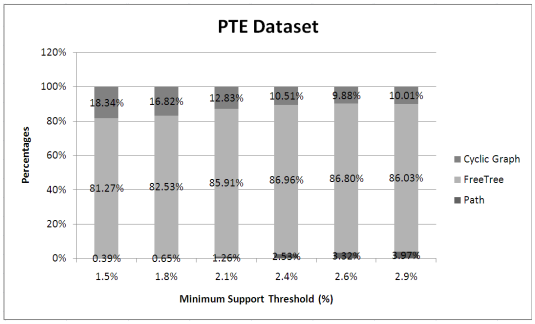

그림 3. 전체 패턴의 비율 (PTE)
Fig. 3. The ratio of total patterns (PTE)

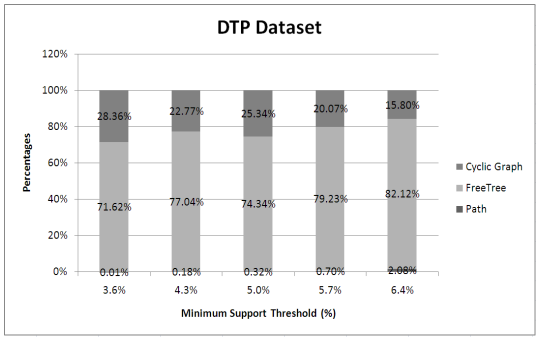

그림 4. 전체 패턴의 비율 (DTP)
Fig. 4. The ratio of total patterns (DTP)

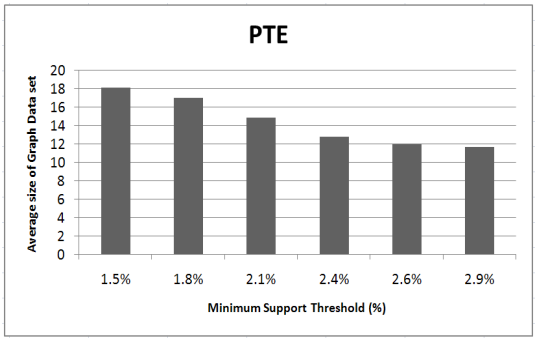

그림 5. 빈발패턴그래프의 평균 크기 (PTE)
Fig. 3. The average size of frequent graph patterns (PTE)

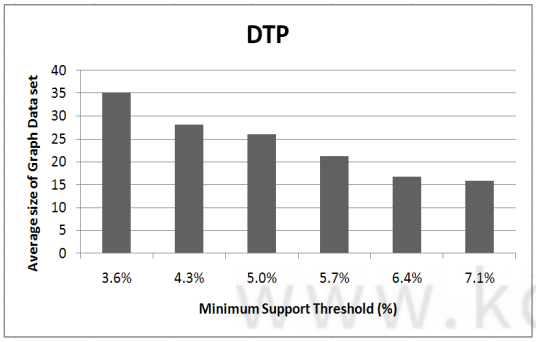

그림 6. 빈발패턴그래프의 평균 크기 (DTP)
Fig. 6. The average size of frequent graph patterns (DTP)

다 DTP데이터 셋에서 효과가 더욱 잘 나타날 것이라는 것을 짐작할 수 있다.

egaston-S도 역시 그래프 동형판단연산의 효율성을 높인 작업이라 할 수 있다. 그러므로 적은수의 데이터셋에 대해서는 대부분의 최소빈도수임계값에서 더 빠른 수행시간을 보인는 것을 알 수 있다. 스패닝 트리를 나열하는데 불필요한 많은 순환에 참여하지 않는 간선들을 확인하는 작업이 사라졌고 또한 모든 순환간선을 제거할 수 없는 순환간선이 먼저 지워짐으로 인해 반복되는 재귀호출이 사라짐으로서 계산상의 많은 이득을 본 것이라 판단된다. 빈도수가 작아질수록 향상되는 성능을 보이는 것은 역시 그래프동형판단의 좋은 성능을 보인다는 것을 보여준다. 그래프마이닝에서는 그래프 동형판단이 매우 많은 비중을 차지한다는 점에서 이러한 작은 변화가 전체적인 성능개선에 많은 도움을 준다.

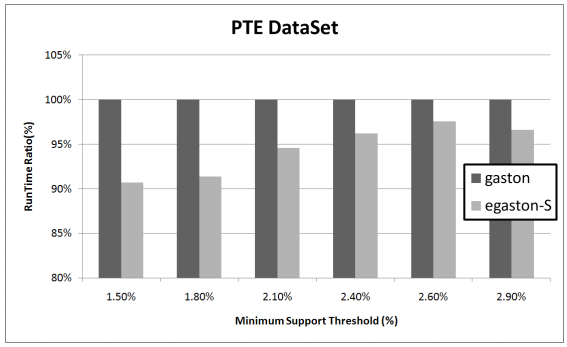

그림 7. egaston-S 수행시간 (PTE)
Fig. 7. Runtime of egaston-S (PTE)

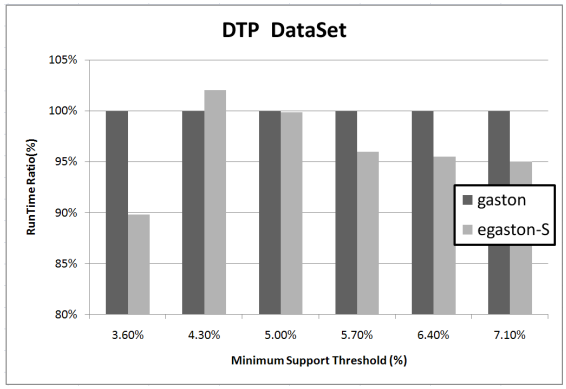

그림 8. egaston-S 수행시간 - DTP
Fig. 8. Runtime of egaston-S (DTP)

그림 7과 8은 egaston-S의 수행시간을 기본 gaston 알고리즘과 비교하여 나타낸다. 차트의 세로축은 기본알고리즘

에 적용된 방법의 수행시간의 백분율 값이다. 제안된 방법에 의해 대부분의 임계값에서 더 빠른 수행시간을 보여주며, 제안된 방법의 효율성을 보여주지만 DTP 데이터 셋의 일부 빈도수에서는 개선되지 않은 성능을 보인다.

기존알고리즘의 문제점이 모든 경우에 해당하지 않는다는 것을 보여준다. 극단적인 하나의 예를 상상해보자. 그래프 내의 하나의 순환만이 존재하고 가장 높은 우선순위를 가지는 순환간선 하나만이 더 높은 우선순위의 스패닝트리를 만들어 낸다면, 기존 알고리즘은 단 한번 만에 스패닝트리 나열작업을 끝내지만 변형된 알고리즘은 모든 순환간선을 거꾸로 모두 차례대로 제거하며 스패닝트리의 우선순위를 판별하는 작업을 한 뒤 마지막에서야 더 높은 우선순위의 스패닝트리가 있음을 알고 작업을 종료하게 된다. 이와 유사한 경우가 많다면 기존 알고리즘의 성능이 변형된 알고리즘 보다 더 좋게 나타날 것이다. 하지만 그래프 내에 순환이 고르게 분포하고, 또 순환에 포함되고 있지 않은 노드들이 많다면 변형된 알고리즘이 더 좋은 성능을 보일 것이고, 이러한 경우가 더 일반적이라는 것을 실험결과들이 보여준다.

VI. 결론 및 추후연구

본 논문에서는 그래프마이닝에서 패턴확장기법에 의한 패턴의 중복생성을 판단하기 위해 사용한 최소신장스패닝트리 나열연산을 효과적 개선한 마이닝 알고리즘을 제안하였다. 그래프마이닝에서 그래프 동형판단이 차지하는 비율이 크며 유사한 작업이 반복적으로 행해지기 때문에 제안한 기법의 성능이 효과적으로 향상됨을 실험을 통하여 그 성능의 개선됨을 보였다. 추후연구로 제안된 기법은 그래프 패턴중 대표 그래프 패턴을 마이닝하거나 제한조건 기반의 그래프 마이닝 기법에도 적용할 수 있을 것이다.

참고문헌

〔1〕 R. Agrawal, T. Imilienski, and A. Swami. "Mining association rules between sets of items in large datasets," In Proceedings of SIGMOD 1993.

〔2〕 T. Asai, K. Abe, S. Kawasoe, H. Arimura, H. Sakamoto, S. Arikawa "Efficient substructure discovery from large semi-structured data," In Proceedings of KDD'04 SIAM SDM'02, April 2002.

〔3〕 J. Han and M. Kamber, "*Data Mining : Concepts and Techniques*" Morgan Kaufmann. Publishers, 2005.

〔4〕 J. Huan, W. Wang, Jan Prins "Efficient mining of frequent subgraphs in the presence of isomorphism," In Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03)

〔5〕 J. Huan, W. Wang, J. Prins, J. Yang "SPIN: mining maximal frequent subgraphs from graph databases," KDD '04: In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining

〔6〕 A. Inokuchi, Takashi Washio, Hiroshi Motoda "An apriori-based algorithm for mining frequent substructures from graph data," In Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Disco (2000)

〔7〕 M. Kuramochi and G. Karypis. "An Efficient Algorithm for Discovering Frequent Subgraphs," In Proceedings of IEEE Trans. Knowl. Data Eng. 16(9): 1038-1051, 2004.

〔8〕 S. Nijssen, J. N. Kok "A quickstart in frequent structure mining can make a difference," In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'04)

〔9〕 N. Vanetik, E. Gudes. "Mining Frequent Labeled and Partially Labeled Graph Patterns," In Proceedings of the International Conference on Data Engineering 2004 (ICDE2004), 2004.

〔10〕 X. Yan, J. Han "gSpan: Graph-based substructure pattern mining," In Proceedings of the 2002 IEEE International Conference on Data Mining

〔11〕 X. Yan, J. Han "CloseGraph: Mining Closed Frequent Graph Patterns," In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. 2003.

〔12〕 F. Zhu, X. Yan, J. Han, P. S. Yu "gPrune: A Constraint Pushing Framework for Graph

Pattern Mining," In Proceedings of Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'07)

〔13〕 산사볼트가람라호차, 황영섭,"근사 알고리즘을 이용한 순차패턴 탐색," 한국컴퓨터정보학회 논문지, 제 14권, 제 5호, 29-36쪽, 2009년 5월.

〔14〕 오승준, "데이터마이닝의 자동 데이터 규칙 추출 방법론 개발: 계층적 클러스터링 알고리듬과 러프 셋 이론을 중심으로," 한국컴퓨터정보학회 논문지, 제 14권, 제 6호, 135-142쪽, 2009년 6월.

저 자 소 개

노 영 상

2007: 충북대학교 컴퓨터공학 학사.
2009: 충북대학교, 컴퓨터공학 석사
관심분야: 데이터마이닝, 데이터베이스

윤 은 일

1997: 고려대학교 이학석사.
1997 - 2006: 한국통신 멀티미디어 연구소 전임/선임연구원.
2005: Texas A&M Univ. 공학박사
2005 - 2006: Texas A&M Univ. 포스닥연구원.
2006 - 2007: 한국전자통신연구원, 선임연구원.
2007 - 현재: 충북대학교 전자정보대학 컴퓨터전공 조교수
관심분야: 데이터마이닝, 정보검색, 데이터베이스

김 명 준

1979: 서강대학교 학사.
1984: Florida Institute of Technology, 공학석사
1992: Texas A&\M Univ. 공학박사
1993 - 현재: 충북대학교 전자정보대학 컴퓨터전공 교수
관심분야: 실시간 시스템, 운영체제, 시스템 소프트웨어