가중치 순회로부터 빈발 순회패턴의 탐사 및 순회분할을 통한 성능향상

이성대* · 박휴찬**

Discovery of Frequent Traversal Patterns from Weighted Traversals and Performance Enhancement by Traversal Split

Seong Dae Lee* · Hyu Chan Park**

요 약

실세계의 많은 문제는 그래프와 그 그래프를 순회하는 트랜잭션으로 모델링될 수 있다. 예를 들면, 웹페이지의 연결구조는 그래프로 표현될 수 있고, 사용자의 웹페이지 방문경로는 그 그래프를 순회하는 트랜잭션으로 모델링 될 수 있다. 이와 같이 그래프를 순회하는 트랜잭션들로부터 빈발 패턴과 같이 중요한 패턴을 찾아내는 것은 의미 있는 일이다. 본 논문에서는, 방향 그래프와 그 그래프를 순회하는 가중치가 있는 트랜잭션들이 주어졌을 때, 빈발한 순회패턴을 탐사하는 알고리즘을 제안한다. 또한, 이 알고리즘의 성능향상을 위하여 순회를 분할하는 방법을 제안하고 실험을 통하여 검증한다.

ABSTRACT

Many real world problems can be modeled as a graph and traversals on the graph. The structure of Web pages can be represented as a graph, for example, and user's navigation paths on the Web pages can be model as a traversal on the graph. It is interesting to discover valuable patterns, such as frequent patterns, from such traversals. In this paper, we propose an algorithm to discover frequent traversal patterns when a directed graph and weighted traversals on the graph are given. Furthermore, we propose a performance enhancement by traversal split, and then verify it through experiments.

키워드

데이터 마이닝, 그래프, 가중치 순회

I. 서 론

데이터 마이닝(data mining)은 대용량 데이터베이스로부터 유용한 정보를 추출하는 기술이라고 정의할 수 있다[1]. 이러한 데이터 마이닝을 위하여 다양한 자료구조(data structure)와 알고리즘(algorithm)이 제안되었고 많은 분야에서 성공적으로 활용되고 있다[2]. 최근에는 그래프(graph)를 기반으로 하는 데이터 마이닝이 주요 관심사가 되고 있다[3,4]. 그래프는 실세계의 많은 문제를 모델링하기에 적합한 데이터 모델로 알려져 있다. 예를 들면, 웹사이트의 구조는 그래프로 쉽게 표현될 수 있다. 즉, 웹 사이트에 존재하는 각각의 웹 페이지(web page)는 그래프의 정점(vertex)으로, 웹 페이지 사이에 존재하는 하이퍼링크(hyperlink)는 그래프의 간선(edge)으

* 한국해양대학교 대학원 컴퓨터공학과
** 한국해양대학교 IT공학부 (교신저자)

접수일자 : 2006. 12. 15

로 표현될 수 있다. 나아가, 사용자가 방문한 웹 페이지 접근 경로(access path)는 그래프를 순회(traversal)하는 트랜잭션(transaction)으로 표현될 수 있다. 또한, 각 페이지에 머문 사용자 시간은 순회의 가중치(weight)로 모델링될 수 있다.

이와 같은 그래프를 활용한 대표적인 데이터 마이닝 문제로 웹로그 마이닝이 있다. 웹로그 마이닝은 웹 페이지의 구조를 그래프로 표현하고, 사용자들의 웹 페이지 접근 기록인 웹로그를 그 그래프를 순회하는 트랜잭션으로 가공하여, 가장 빈발하게 발생하는 페이지 접근 경로를 찾는 문제이다. 이러한 웹로그 마이닝은 그래프를 순회하는 트랜잭션으로부터 빈발 순회패턴(frequent traversal pattern)을 찾는 문제로 치환하여 해결할 수 있다[3,5]. 하지만, 기존의 방법들은 순회에 부여될 수 있는 가중치 정보를 탐사 과정이나 결과에 반영하지 않는다.

본 논문에서는 기반 그래프(base graph)와 그 그래프를 순회하는 가중치가 부여된 트랜잭션들이 주어졌을 때, 빈발 순회패턴을 탐사하는 알고리즘을 제안한다. 먼저, 주어진 순회로부터 각 간선의 가중치의 평균(average)과 표준편차(stan- dard deviation)를 구하여 기반 그래프의 간선 가중치로 부여한다. 그 다음, 수정된 기반 그래프와 순회 데이터베이스로부터 간선의 가중치를 고려하여 빈발 순회패턴을 탐사한다.

특히, 탐사 결과의 신뢰도를 높이기 위하여 순회 트랜잭션의 노이즈(noise)를 제거하는 방법을 제안한다. 어떤 순회의 특정 간선의 가중치가 다른 값에 비하여 현격히 크거나 작으면 노이즈로 간주될 수 있다. 예를 들면, 어떤 사용자가 특정 웹페이지에 머문 시간이 현격히 큰 경우는 자리를 비운 경우로, 현격히 작은 경우는 내용을 보지 않고 그냥 지나간 경우로 볼 수 있다. 따라서 이러한 페이지 방문은 정상적인 페이지 방문과 구별하여 노이즈로 간주하여 최대한 탐사 결과에 반영하지 않는 것이 바람직하다. 본 논문에서는 먼저 순회들로부터 각 간선의 가중치의 평균과 표준편차를 구하여 기반 그래프의 간선에 부여한다. 이 평균과 표준편차로부터 신뢰구간(confidence interval)을 구하여, 순회에 포함된 간선의 가중치가 이 신뢰구간 내에 있으면 정상적인 간선으로 간주하고, 신뢰구간 외에 있으면 노이즈로 간주한다.

이를 바탕으로 후보 순회패턴(candidate tra- versal patterns)을 생성하는 각 단계에서 노이즈를 제거하는 기본적인 알고리즘을 제안한다. 이러한 기본 알고리즘은 노이즈 판별을 후보 생성의 매 단계마다 반복적으로 수행해야 하기 때문에 성능 저하를 초래할 수 있다. 따라서 성능을 향상시키기 위한 방안으로 각 순회의 노이즈 간선을 사전에 판별하여 이 간선을 제거하여 순회를 분할하는 방안을 제안한다. 이 두 가지 방법의 성능을 비교하기 위한 실험과 그 결과를 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해서 살펴보고, 3장에서는 순회패턴 탐사를 위한 가중치 그래프에 대하여 설명한다. 4장에서는 가중치를 고려한 빈발 순회패턴을 탐사하는 기본 알고리즘을 제안하고, 5장에서는 순회분할을 통한 성능향상 방안을 제안한다. 6장에서는 제안한 알고리즘을 구현하여 실험한 후 결과를 분석한다. 마지막으로 7장에서는 결론과 향후 연구과제에 대하여 논한다.

II. 관련 연구

데이터 마이닝은 대규모 데이터베이스에 내재해 있는 고급 정보나 패턴을 추출해서 의사 결정(decision)이나 예측(prediction) 등에 활용하고자 하는 기술이다. 기존의 데이터 마이닝 연구를 본 논문과 관련하여 그래프와 가중치를 기준으로 분류하면, 기반이 되는 그래프는 고려치 않고 사용자의 트랜잭션만 고려한 경우, 기반 그래프와 트랜잭션을 동시에 고려한 경우, 탐사 과정에 가중치를 고려한 연구로 나누어 볼 수 있다.

첫 번째, 사용자의 트랜잭션만 고려한 경우는 연관 규칙 탐사나 순차 패턴 탐사 등이 있다. 연관 규칙 탐사는 데이터베이스에 존재하는 항목(item)들의 친화도나 패턴을 찾아내는 방법으로서 "빵을 구매하는 고객의 40%는 우유도 함께 구매한다."와 같이 트랜잭션에 있는 항목간의 연관성을 찾아내는 방법으로 Apriori 알고리즘이 대표적이다[1]. 이 알고리즘은 두 단계로 구성된다. 우선, 각 아이템의 빈도수를 계산하여 최소 지지도(minimum support) 이상을 만족하는 항목들의 집합인 빈발 항목집합(frequent itemsets)을 찾는다. 그 다음, 빈발 항목집합으로부터 최소 신뢰도(minimum confidence) 이상을 만족하는 연관규칙을 구한다. 이러한 알고리즘을 개선한 것으로는 DHP(Direct Hashing and Pruning) 알고리즘[6]과 Partition- ing 알고리즘[7] 등이 있다. 순차 패턴 탐사는 연관 규칙 탐사를 확장한 것으로서 트랜잭션

항목들 간의 시간적인 순서를 고려한 것이다[8]. 또한, 순차 패턴 탐사의 응용으로 기반 그래프는 고려치 않고 순회들만 고려하여 빈발 패턴을 탐사하는 연구가 있었다[5].

두 번째는 본 논문에서와 같이 기반 그래프가 주어지고, 기반 그래프의 간선을 따라서 순회하는 트랜잭션들로부터 빈발 경로(frequent path)를 탐사하는 것이다. 이는 순차 패턴과 유사하지만, 기반 그래프의 정점과 간선의 연결 관계를 고려하여 빈발 경로 여부를 판별한다는 점이 다르다. 예를 들면, 웹 로그 마이닝에서 웹의 구조는 기반 그래프로, 웹 로그는 전처리(pre-processing) 과정을 거친 후 순회 트랜잭션으로 변환하여 빈발하는 방문 경로를 탐사할 수 있다[3,4].

세 번째, 가중치를 고려한 데이터 마이닝으로 각 항목(item)의 수익이나 판매량을 항목의 가중치로 부여하여 탐사 결과에 반영하는 방법이다. Cai[9]는 각 항목에 가중치가 부여되어 있는 경우 빈발 항목집합(frequent itemsets)을 탐사하는 일반적인 방법을 제안하였다. 즉, 가중치 지지도(weighted support)와 지지도 한계값(support bound)이라는 개념을 도입하여 가중치가 있는 항목들의 패턴을 탐사하였다.

하지만, 본 논문과 같이 기반 그래프와 순회 트랜잭션을 동시에 고려하면서 가중치도 함께 고려한 연구는 우리의 이전 연구[10]외에는 없었다.

III. 순회패턴 탐사를 위한 가중치 그래프

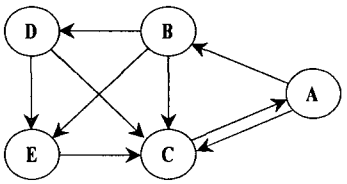
본 논문에서는 기반 그래프와 그 그래프를 순회하는 가중치가 부여된 트랜잭션들이 주어졌을 때, 가중치를 고려하여 빈발하게 발생하는 순회패턴을 탐사하는 알고리즘을 제안하고자 한다. 이를 위해 필요한 기반 그래프, 순회, 신뢰구간 등에 대한 정의는 다음과 같다.

[정의 1] *기반 그래프(base graph)*는 단순 방향 그래프(simple directed graph)로서 유한한 정점(vertex)과 간선(edge)의 집합이며, 셀프루프(self loop)가 없고 간선은 방향성을 가진다.

정의 1은 순회의 기반이 되는 그래프를 정의한 것으로서, 본 논문에서는 셀프루프가 없는 단순 그래프와 간선이 방향성을 가지고 있는 방향 그래프를 가정한다. 본 논문에서는 인접 리스트(adjacent list)를 사용하여 구현하였다.

[정의 2] *순회(traversal)*는 기반 그래프에 존재하는 간선들의 연속적인 순차이다. 그래프에서 간선은 정점의 쌍으로 표현되므로, 순회를 정점들의 순차인 $t = <v_1, v_2, ..., v_n>$로 나타낼 수 있다. *가중치 순회(weighted traversal)*는 순회에 포함된 각 간선이 가중치를 지니는 것을 의미한다. 따라서 가중치 w가 부여된 순회 t는 $(t, w) = (<v_1, v_2, ..., v_n>, <w_1, w_2, ..., w_{n-1}>)$로 나타낼 수 있고, w_i는 간선 $<v_i, v_{i+1}>$의 가중치를 의미한다. *순회 데이터베이스(traversal database)*는 이러한 가중치 순회들의 집합을 나타낸다.

본 논문에서는 기본적으로 가중치 순회를 대상으로 하므로, 이하에서는 가중치 순회를 단순히 순회라고 언급한다. 정의 1과 2를 따르는 기반 그래프와 순회 데이터베이스의 예제는 그림 1과 같다. 그림에서 알 수 있듯이, 모든 순회는 기반 그래프에 존재하는 간선을 따라서 정점을 순회하면서 생성된다. 예를 들면, 순회 #1 <*A, B, C*>는 정점 *A, B, C*를 간선 <*A, B*>, <*B, C*>의 순서대로 방문하였으며, 각 간선의 가중치는 각각 2.2, 2.0이라는 것을 의미한다.



(a) 기반 그래프

ID	*Traversal*	*Weight*
1	<*A B C*>	<2.2 2.0>
2	<*B D E C A*>	<3.0 4.3 3.5 3.1>
3	<*C A B D*>	<2.9 2.0 4.0>
4	<*D C A*>	<4.0 3.0>
5	<*B C A*>	<2.2 2.9>
6	<*A B E C*>	<2.1 3.4 3.2>
7	<*A B D E C*>	<1.4 3.9 4.4 3.2>
8	<*B E C*>	<2.3 3.4>
9	<*B D C*>	<3.8 3.1>
10	<*C A B D*>	<2.5 2.2 4.1>

(b) 순회 데이터베이스

그림 1. 기반 그래프와 순회 데이터베이스의 예제
Fig. 1. An example of base graph and traversal database

[정의 3] *부분순회(sub-traversal)*는 순회에 포함된 연속된 정점의 부분 집합을 의미한다. 즉, 순회 $t = <v_1, v_2, ..., v_n>$의 부분순회는 $s = <s_1, s_2, ..., s_m>$로 정의할 수 있으며, 여기서 $s_j = v_{j+k}, k \geq 0, 1 \leq j \leq m, j + k \leq n$ 이다.

예를 들면, 그림 1(b)에서 순회 #7 <*A B D E C*>의 경우, 길이가 4가 되는 부분순회는 <*A B D E*>와 <*B D E C*>의 2개만 존재한다. 길이가 3이 되는 부분순회는 <*A B D*>, <*B D E*>, <*D E C*>의 3개만 존재할 수 있다. 이러한 부분순회에 대한 정의는 순회패턴이 어떤 순회에 포함되는지를 판별하기 위하여 사용된다.

[정의 4] *가중치 기반 그래프(weighted base graph)*는 기반 그래프 $G = (V, E)$의 각 간선에 가중치 정보가 부여된 것이다. 여기서, 기반 그래프의 간선 $<v_i, v_j> \in V(G)$의 가중치 정보는 순회 데이터베이스에 존재하는 모든 간선 $<v_i, v_j>$의 가중치 평균(average, μ)과 표준편차(standard deviation, σ)를 계산하여 (μ_{ij}, σ_{ij})의 쌍으로 부여한다.

정의 4를 따르는 가중치 기반 그래프는 다음과 같이 구할 수 있다. 기반 그래프의 각 간선에 대하여 순회 데이터베이스에서 동일한 간선의 가중치를 모두 찾아 그 평균과 표준편차를 계산한다. 예를 들면, 그림 1(a)의 그래프의 간선 <*A B*>의 가중치는 다음과 같이 부여된다. 먼저, 순회 데이터베이스로부터 간선 <*A B*>를 포함하는 순회 #1, #3, #6, #7, #10을 탐색한 후, 각각의 가중치 2.2, 2.0, 2.1, 1.4, 2.2로부터 평균 2.0과 표준편차 0.3을 계산하여, 그들의 쌍 (2.0, 0.3)을 그래프의 간선 <*A B*>에 부여한다. 그림 2는 그림 1(a)의 기반 그래프에 가중치가 부여된 가중치 기반 그래프이다.

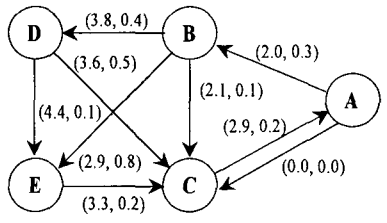


그림 2. 가중치 기반 그래프 예제
Fig. 2. An example of weighted base graph

[정의 5] *신뢰구간(confidence interval)*은 주어진 신뢰수준(confidence level)으로 모수(random variable) x가 포함될 구간을 의미한다.

예를 들면, 정규분포인 경우, 신뢰수준이 95%일 때 신뢰구간은 수식 1과 같이 설정된다.

$$P(\mu - 1.96\sigma \leq x \leq \mu + 1.96\sigma) = 0.95 \quad \text{(수식 1)}$$

수식 1에서 알 수 있듯이, 신뢰구간의 상한 값(upper bound)은 $(\mu + 1.96\sigma)$이고 하한 값(lower bound)은 $(\mu - 1.96\sigma)$이다. 여기서 μ는 간선의 평균을, σ는 간선의 표준편차를 의미한다. 신뢰수준 95%의 의미는 모수 x가 상한 값과 하한 값 사이에 있을 확률을 의미하며, 나머지 5%는 상한 값과 하한 값의 외부에 존재할 확률을 의미한다.

본 논문에서, 정의 5는 순회의 어떤 간선이 신뢰할 수 있는 지 여부를 판별하기 위하여 사용된다. 즉, 순회의 어떤 간선의 가중치가 신뢰구간 내에 있으면 이 간선을 신뢰할 수 있는 것으로 인정하고, 신뢰구간 외에 있으면 노이즈로 간주한다. 노이즈로 간주되는 간선은 탐사 과정에서 존재하지 않는 간선과 유사하게 취급한다. 따라서 신뢰할 수 있는 간선은 빈발 순회패턴을 탐사하는 과정에 포함시키고, 노이즈로 간주된 간선은 무시한다.

Ⅳ. 빈발 순회패턴 탐사

본 장에서는 3장에서 구한 가중치 기반 그래프와 순회 데이터베이스로부터 가중치를 고려하여 빈발하는 순회패턴을 탐사하는 알고리즘을 제안한다. 먼저, 이러한 빈발 순회패턴의 주요한 성질을 살펴보기로 한다. 길이가 k인 임의의 순회패턴 $p = <p_1, p_2, ..., p_k>$를 생각하자. 순회패턴 p에서 길이가 k-1인 부분 순회패턴(sub-traversal pattern)은 $<p_1, p_2, ..., p_{k-1}>$과 $<p_2, p_3, ..., p_k>$ 2개가 존재한다. 따라서 길이가 k인 순회패턴은 길이가 k-1인 2개의 순회패턴의 결합으로 생성할 수 있으며, 길이가 k인 순회패턴이 빈발하다는 것은 길이가 k-1인 2개의 부분 순회패턴이 모두 빈발한 경우에만 성립된다. 이러한 성질은 후보 순회패턴의 생성 시 핵심적으로 활용된다.

그림 3은 본 논문에서 제안하는 *IMTP* (In-Mining

Traversal Patterns) 알고리즘으로서 빈발 순회패턴을 탐사하는 각 단계에서 간선의 노이즈 여부를 판별한다.

Algorithm *IMTP*()

```
input: weighted base graph Gw, traversal database T,
       minimum support minSup, confidence level confLev
output: frequent patterns Lk

begin
  C1 ← set of all vertices
  k = 1

  // while candidates exist
  while (|Ck| > 0) {
    // count supports for candidate patterns
    for each traversal t ∈ T {
      P = {p | p ∈ Ck, p is sub-traversal of t}
      ∀p∈P p.count++
    }

    // prune candidate patterns w.r.t confidence interval
    if (k ≥ 2) Ck ← pruneCandidates(Ck, Gw, confLev)

    // obtain frequent patterns
    Lk = {p | p ∈ Ck, p.count ≥ minSup}

    // generate candidate patterns for next step
    Ck+1 ← genCandidates(Lk, Gw)
    k++
  }
end
```

그림 3. 빈발 순회패턴 탐사 알고리즘(IMTP)
Fig. 3. Algorithm for discovering frequent traversal patterns (IMTP)

이 알고리즘은 길이가 1인 후보 순회패턴으로 기반 그래프의 각 정점을 초기화하여 시작한다. 이 후보 순회패턴은 1개의 정점으로만 구성되어 있어 간선이 존재하지 않으므로 가중치는 고려치 않고 발생 횟수만을 고려한다. 즉, 순회 데이터베이스에서 후보 순회패턴을 포함하는 순회의 갯수인 지지도(support)를 구한다. 이렇게 구한 후보 순회패턴의 지지도가 설정한 최소 지지도 이상이면 빈발 순회패턴이 된다. 이러한 빈발 순회패턴을 조인하여 길이가 2인 후보 순회패턴을 생성한다. 이후, 후보 순회패턴의 길이가 2 이상이 되는 각 단계에서는 가중치와 신뢰구간을 적용하여 지지도를 구한다. 이때, 후보 순회패턴의 지지도는 각 후보 순회패턴에 포함된 모든 부분 간선이 신뢰구간 내에 존재하는 경우에만 후보 순회패턴의 지지도 계산에 참여한다. 역시 후보 순회패턴의 지지도가 최소 지지도 이상이면 빈발 순회패턴이 된다. 이렇게 찾은 빈발 순회패턴을 조인하여 길이가 1 증가된 후보 순회패턴을 생성한다. 이 과정을 반복하다가 더 이상 후보 순회패턴을 생성할 수 없는 경우에 알고리즘은 종료된다.

그림 3의 *IMTP* 알고리즘에서 함수 *prune- Candidate*()는 가중치 기반 그래프의 각 간선에 부여되어 있는 가중치 평균과 표준편차를 이용하여 신뢰구간을 설정한 후, 설정된 신뢰구간 외부의 가중치를 갖는 순회의 간선은 노이즈로 간주하여 지지도 계산에서 제외한다. 즉, 순회 $(t, w) = (\langle v_1, v_2, ..., v_n\rangle, \langle w_1, w_2, ..., w_n\rangle)$가 후보 순회패턴 $p = \langle p_1, p_2, ..., p_k\rangle$를 포함하더라도, 만약 (t, w)에 있는 간선 중 p와 겹치는 어떤 간선의 가중치가 신뢰구간 외부에 존재할 경우 (t, w)는 후보패턴 p의 지지도 계산에서 제외한다. 예를 들어, 그림 2의 가중치 기반 그래프의 간선 <*A B*>의 신뢰구간은 다음과 같이 설정된다. 그림 2에서 볼 수 있듯이, 간선 <*A B*>의 가중치 평균이 2.0이고 표준편차가 0.3이다. 신뢰수준이 95%라고 가정하였을 때, 이들을 수식 1에 대입하면 신뢰구간은 $(2.0 - 1.96 \times 0.3) \sim (2.0 + 1.96 \times 0.3) \equiv 1.41 \sim 2.59$로 계산된다. 이 신뢰구간을 이용하여 순회의 간선이 노이즈인지를 판별하는 것은 다음과 같이 한다. 예를 들면, 그림 1(b)에서 순회 #7 (<*A B D E C*>, <1.4, 2.3, 4.4, 3.2>)은 후보 순회패턴 <*A B D*>를 포함하고 있다. 그러나 순회 #7의 간선 <*A B*>의 가중치는 1.4로서 가중치 기반 그래프의 간선 <*A B*>의 신뢰구간 1.41 ~ 2.59의 외부에 존재하므로 후보 순회패턴 <*A B D*>의 지지도 계산에서 제외된다.

그림 3의 *IMTP* 알고리즘의 *genCandidate()*는 다음 단계의 새로운 후보 순회패턴을 생성하는 함수이다. 길이가 k인 빈발 순회패턴은 다음 단계인 길이 k+1의 후보 순회패턴을 생성하기 위해 조인하게 된다. 즉, 길이가 k인 2개의 빈발 순회패턴 $\langle p_1, p_2, ..., p_k\rangle$와 $\langle p_2, p_3, ..., p_{k+1}\rangle$가 존재한다면, 조인을 통해 길이 k+1인 새로운 후보 순회패턴 $\langle p_1, p_2, ..., p_{k+1}\rangle$을 생성할 수 있다. 예를 들면, <*A B C*>와 <*B C D*>를 조인하면 <*A B C D*>를 구할 수 있다.

그림 4는 그림 1(b)의 순회 데이터베이스와 그림 2의 가중치 기반 그래프에 *IMTP* 알고리즘을 적용한 예이다. 예제에서의 최소 지지도는 2, 신뢰수준은 95%라고 가정하였다.

C_1

Candidate Pattern	Pruned Support
<A>	8
<B>	9
<C>	10
<D>	6
<E>	4

L_1

Frequent Pattern	Pruned Support
<A>	8
<B>	9
<C>	10
<D>	6
<E>	4

C_2

Candidate Pattern	Initial Support	Pruned Support
<A B>	5	4
<A C>	0	0
<B C>	2	2
<B D>	5	4
<B E>	2	2
<C A>	5	4
<D C>	2	2
<D E>	2	2
<E C>	4	4

L_2

Frequent Pattern	Pruned Support
<A B>	4
<B C>	2
<B D>	4
<B E>	2
<C A>	4
<D C>	2
<D E>	2
<E C>	4

C_3

Candidate Pattern	Initial Support	Pruned Support
<A B C>	1	1
<A B D>	3	2
<A B E>	1	1
<B C A>	1	1
<B D C>	1	1
<B D E>	2	1
<B E C>	2	2
<C A B>	2	1
<D C A>	1	1
<D E C>	2	2
<E C A>	1	1

L_3

Frequent Pattern	Pruned Support
<A B D>	2
<B E C>	2
<D E C>	2

그림 4. 빈발 순회패턴 탐사 예제
Fig. 4. An example of discovering frequent traversal patterns

그림 3의 알고리즘은 먼저 길이가 1인 후보 순회패턴 C_1을 그래프에 있는 모든 정점으로 초기화한다. 이후 순회 데이터베이스를 탐색하여 C_1에 포함된 각 후보 순회패턴의 지지도를 구한다. 만약 각 후보 순회패턴의 지지도가 최소 지지도 2 이상을 만족할 경우 빈발 순회패턴 L_1에 포함시킨다. L_1에 포함된 빈발 순회패턴을 상호 조인하여 길이가 2인 새로운 후보 순회패턴 C_2를 생성한다. 이때 기반 그래프에 존재하지 않는 간선은 C_2에서 제외한다. 길이가 2인 후보 순회패턴 C_2의 지지도를 구하기 위해 다시 순회 데이터베이스를 탐색한다. 이때의 지지도는 신뢰구간을 적용하여 구하게 된다. 예를 들면, 후보 순회패턴 <*A B*>를 포함하는 순회가 5개 있으므로, 초기 지지도는 5이다. 그러나 순회 #7 (<*A B D E C*>, <1.4 2.3 4.4 3.2>)의 간선 <*A B*>의 가중치가 1.4로서 신뢰구간 1.41 ~ 2.59의 외부에 존재하므로, 이를 제외하면 실제 지지도는 4가 된다. 이러한 알고리즘을 적용시켜 생성된 C_2의 지지도가 최소 지지도 이상이면 빈발 순회패턴 L_2에 포함시킨다. C_3 역시 L_2를 조인하면 생성할 수 있다. 예를 들면, <*A B*>와 <*B C*>의 조인을 통해 <*A B C*>를 구할 수 있다. 이상과 같이 알고리즘은 L_3까지 진행되며, 길이가 4인 C_4는 더 이상 생성될 수 없으므로 알고리즘은 종료된다.

V. 순회분할을 통한 성능향상

4장의 *IMTP* 알고리즘은 후보 순회패턴 C_k의 지지도를 구하는 매 단계마다 신뢰구간을 검사하였다. 따라서 신뢰구간의 반복적인 적용으로 인해 계산량이 증가한다는 문제점이 발생한다. 이를 개선하기 위하여 본 장에서는 *PMTP*(Pre-Mining Traversal Patterns) 알고리즘을 제안한다. 이 알고리즘은 전처리 단계에서 노이즈를 포함하고 있는 순회를 2개 이상의 부분순회로 분할하여 계산량을 감소시키는 방법이다.

[정의 6] 어떤 순회 (t, w) = (<$v_1, v_2, ..., v_n$>, <$w_1, w_2, ..., w_{n-1}$>)에 신뢰구간을 벗어나는 간선 <v_i, v_{i+1}>이 포함되어 있으면, 이 순회는 2개의 부분순회 $(t, w)'$ = (<$v_1, v_2, ..., v_i$>, <$w_1, w_2, ..., w_{i-1}$>)과 $(t, w)''$ = (<$v_{i+1}, v_{i+2}, ..., v_n$>, <$w_{i+1}, w_{i+2}, ..., w_{n-1}$>)로 분할된다. 이렇게 분할된 순회의 집합을 *분할된 순회 데이터베이스(splitted traversal database)*라고 한다.

그림 5는 그림 1(b)의 순회 데이터베이스를 그림 2의 가중치 기반 그래프의 각 간선의 신뢰구간을 계산한 후 정의 6을 기반으로 각 순회들을 분할한 예제이다. 그림 1(b)의 순회 #2 (t, w) = (<*B D E C A*>, <3.0 4.3 3.5 3.1>)에서 간선 <*B D*>의 경우 신뢰구간 3.02 ~ 4.58의 범위를 벗어나므로 2개의 부분순회 <*B*>와 <*D E C A*>로 분할된다.

ID	Traversal	Weight
1	<A B C>	<2.2 2.0>
2′	<B>	<0.0>
2″	<D E C A>	<4.3 3.5 3.1>
3	<C A B D>	<2.9 2.0 4.0>
4	<D C A>	<4.0 3.0>
5	<B C A>	<2.2 2.9>
6	<A B E C>	<2.1 3.4 3.2>
7′	<A>	<0.0>
7″	<B D E C>	<3.9 4.4 3.2>
8	<B E C>	<2.3 3.4>
9	<B D C>	<3.8 3.1>
10′	<C>	<0.0>
10″	<A B D>	<2.2 4.1>

그림 5. 분할된 순회 데이터베이스 예제
Fig. 5. An example of splitted traversal database

그림 6은 분할된 순회 데이터베이스를 기반으로 빈발 순회패턴을 찾는 *PMPT* 알고리즘을 보여주고 있다. 그림 6에서 *SplitTraversals*() 함수는 정의 6에 따라 순회 데이터베이스를 분할하는 함수이다. 예를 들면, 그림 1(b)의 각 순회는 이 함수를 통하여 그림 5의 분할된 순회 데이터베이스로 변환된다. 분할된 순회는 노이즈가 제거된 상태이므로 알고리즘의 각 단계에서 신뢰구간을 적용할 필요가 없어 성능 개선이 기대된다. 하지만 사전에 모든 순회에 대하여 순회분할 여부를 판정해야 하고 순회분할에 따른 순회 데이터베이스의 크기가 증가하는 등 성능 저하 요인도 발생한다. 따라서 전체적인 성능의 개선은 실험을 통하여 확인할 수 있을 것이다. 분할된 순회 데이터베이스에 *PMTP* 알고리즘을 적용하여 얻게 되는 빈발 순회패턴은 *IMPT* 알고리즘의 결과와 같게 된다. 즉, 그림 5의 분할된 순회 데이터베이스로부터 탐사되는 빈발 순회패턴은 그림 4와 동일하게 된다.

Algorithm *PMTP*()

```
input:  weighted base graph Gw, traversal database T,
        minimum support minSup, confidence level confLev
output: frequent patterns Lk

begin
  // split traversals into sub-traversals
  T' = SplitTraversals(Gw, T, confLev)

  C1 ← set of all vertices
  k = 1

  // while candidates exist
  while (|Ck| > 0) {
    // count supports for candidate patterns
    for each traversal t' ∈ T' {
      P = {p | p ∈ Ck, p is sub-traversal of t'}
      ∀p∈P p.count++
    }

    // obtain frequent patterns
    Lk = {p | p ∈ Ck, p.count ≥ minSup}

    // generate candidate patterns for next step
    Ck+1 ← genCandidates(Lk, Gw)
    k++
  }
end
```

그림 6. 개선된 빈발 순회패턴 탐사 알고리즘(PMTP)
Fig. 6. Enhanced algorithm for discovering frequent traversal patterns (PMTP)

VI. 실험 및 평가

본 논문에서는 제안한 2개의 알고리즘 *IMTP*와 *PMTP*의 성능을 비교 평가하기 위하여 알고리즘을 구현하고 실험하였다. 구현은 Pentium IV 3.00GHz CPU, 1GB Memory, Windows XP Operating System 환경에서, 프로그래밍 언어로 Microsoft Visual C++ 6.0을, 그래프 및 순회 정보를 저장하기 위한 데이터베이스로 Microsoft SQL Server 2005 Standard Edition을 사용하였다.

실험에 사용된 그래프는 정점의 수와 각 정점 당 평균 간선의 수를 입력으로 하여 생성하였고, 순회 트랜잭션은 기반 그래프의 간선을 따라 순회하는 경로(path)의 각 간선에 가중치를 부여하여 생성하였다. 간선에 가중치를 부여함에 있어서, 대부분의 자연적인 측정치가 정규분포 형태를 지니므로, 가중치의 분포가 정규분포가 되도록 난수 발생기(random number generator)를 이용하여 생성하였다. 실제 실험에서 사용한 기반 그래프는 정점 수 100개, 간선 수 2,000개, 즉 각 정점 당 평균 간선 수가 20개인 그래프이다. 순회 데이터베이스의 순회 수는 100,000개, 순회의 최대 길이는 51이다. 실험에서의 신뢰수준은 본문 예제의 95%보다 더 많은 후보 및 빈발 순회패턴을 생성한 후의 결과를 비교하기위하여 98%로 설정하였다.

표 1과 그림 7은 본 논문에서 제안한 *IMTP*와 *PMTP* 알고리즘의 수행 결과와 수행 시간을 비교하였다. 이 실험에서는 최소 지지도에 따라 생성되는 후보 및 빈발 순회패턴의 수가 달라지므로 최소 지지도를 1%에서 10%까지 바꾸어 가며 비교하였다. *IMTP* 알고리즘의 경우, 빈발 순회패턴을 탐사하는 매 단계마다 신뢰구간을 검사하는 과정이 필요하다. 반면에, *PMTP* 알고리즘에서는

순회분할을 위한 전처리 과정이 필요하지만, 빈발 순회 패턴을 탐사하는 각 단계에서의 신뢰구간 검사는 생략할 수 있다. 표 1과 그림 7에서 확인할 수 있듯이, *PMTP* 알고리즘의 처리 속도가 *IMPT* 알고리즘에 비해 평균적으로 46.8% 정도 개선됨을 확인할 수 있다. 즉, *PMPT* 알고리즘에서 순회 데이터베이스를 분할하는 비용과 순회 데이터베이스의 크기 증가에 따른 비용이 *IMPT* 알고리즘에서 매 단계마다 신뢰구간을 검사하는 비용보다 더 작아 성능이 개선됨을 보여주고 있다.

표 1. 알고리즘의 성능 비교
(그래프 정점 100개, 간선 2000개; 순회 100,000개; 신뢰수준 98%)
Table 1. Comparisons of algorithm performance

최소 지지도(%)	패턴 길이	패턴 수	수행시간(초)		비교 (PMTP/IMTP)
			IMTP	PMTP	
1	51	3	7,864	3,727	47%
2	49	2	2,684	1,681	63%
3	42	1	2,548	1,218	48%
4	35	4	2,107	905	43%
5	31	2	1,628	712	44%
6	30	1	1,259	557	44%
7	26	1	993	441	44%
8	23	1	800	357	45%
9	21	1	649	292	45%
10	19	1	524	236	45%
수행시간비교평균					46.8%

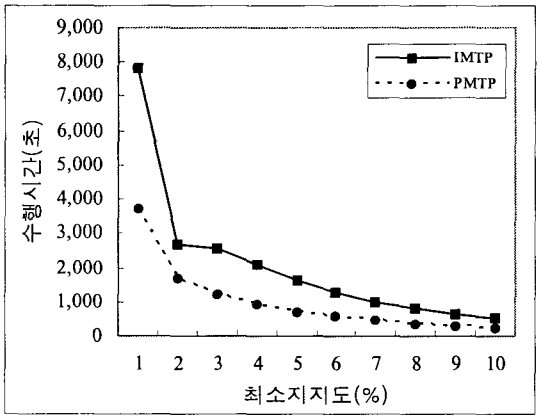


그림 7. 알고리즘 수행시간 비교
Fig. 7. Comparisons of algorithm runtime

Ⅶ. 결 론

본 논문에서는 기반 그래프를 순회하는 가중치가 있는 트랜잭션으로부터 빈발 순회패턴을 탐사하는 알고리즘을 제안하였다. 이 알고리즘에서는 간선의 가중치 평균과 표준편차를 이용하여 신뢰구간을 설정한 후, 신뢰구간을 벗어나는 노이즈 경로를 제거하였다. 이러한 방법을 통하여 최종 단계에서 찾아진 최대 빈발 순회패턴은 보다 나은 신뢰성을 확보할 수 있다. 또한 성능 개선을 위하여 순회 데이터베이스를 분할한 후 탐사하는 알고리즘도 제안하였다. 알고리즘의 구현과 실험을 통하여 성능이 개선됨을 확인하였다.

본 논문에서 제안하는 알고리즘에서 정점이나 간선에 부여되는 가중치는 그래프의 응용 분야에 따라 다양한 형태로 주어질 수 있다. 예를 들면, 웹 로그 마이닝의 경우 사용자의 웹 페이지 이동 시간은 간선의 가중치로, 웹 페이지의 정보량은 정점의 가중치로 부여될 수 있다. 이에 따라, 현재 본 논문의 연구결과를 확장하여 기반 그래프와 순회에 다양한 형태의 가중치가 있는 경우에 빈발 순회패턴을 탐사하는 알고리즘을 개발하고 있다. 또한, 이를 웹 마이닝과 같은 실제적인 분야에 응용하는 연구를 진행하고 있다.

참고문헌

[1] R. Agawal and R. Srikant, Fast Algorithms for Mining Association Rules, Proc. of the 20th Int. Conf. on Very Large Database (VLDB), pp.487-499, Chile, Sep. 1994.

[2] J. Han and M. Kamber, Data Mining Concepts and Techniques, Morgan Kaufman, 2001.

[3] A. Nanopoulos and Y. Manolopoulos, "Finding Generalized Path Patterns for Web Log Data Mining", Proc. of the 4th East-European Conf. on Advances in Databases and Information Systems (ADBIS), pp.215-225, Czech Republic, Sep. 2000.

[4] A. Nanopoulos and Y. Manolopoulos, "Mining Patterns from Graph Traversals", Data and Knowledge Engineering (DKE), vol.37, no.3, pp.243-266, Jun. 2001.

[5] M.S. Chen, J.S. Park, and P.S. Yu, "Efficient Data Mining for Path Traversal Patterns", IEEE Trans. on

Knowledge and Data Engineering, vol.10, no.2, pp.209-221, Mar. 1998.

[6] J.S. Park, M.S. Chen, and P.S. Yu, "An Effective Hash-Based Algorithm for Mining Association Rules", Proc. of ACM SIGMOD Int. Conf. of Management of Data, pp.175-186, USA, May 1995.

[7] A. Savasere, E. Omiecinski, and S.B. Navathe, "An Efficient Algorithm for Mining Association Rules in Large Databases", Proc. of 21st Int. Conf. on Very Large Database (VLDB), pp.432-444, Switzerland, Sep. 1995.

[8] R. Agrawal and R. Srikant, "Mining Sequential Patterns", Proc. of International Conference on Data Engineering, pp.3-14, Taiwan, Mar. 1995.

[9] C.H. Cai, W.C. Ada, W.C. Fu, C.H. Cheng, and W.W. Kwong, "Mining Association Rules with Weighted Items", Proc. of International Database Engineering and Applications Symposium (IDEAS), pp.68-77, UK, Aug. 1998.

[10] S.D. Lee and H.C. Park, "Mining Frequent Patterns from Weighted Traversals on Graph using Confidence Interval and Pattern Priority", International Journal of Computer Science and Network Security (IJCSNS), vol.6, no.5A, pp.136-141, May. 2006.

저자소개

이 성 대(Seong-Dae Lee)

1999 한국해양대학교 컴퓨터공학과 (공학사)
2001 한국해양대학교 컴퓨터공학과 (공학석사)
2001~현재 한국해양대학교 컴퓨터공학과 박사과정
1995~1996 미래정보CIM
※관심분야 : 데이터베이스, 해양정보시스템, 데이터마이닝, XML

박 휴 찬(Hyu-Chan Park)

1985 서울대학교 전자공학과 (공학사)
1987 한국과학기술원 전기및전자공학과 (공학석사)
1995 한국과학기술원 전기및전자공학과 (공학박사)
1987~1990 금성반도체
1997~현재 한국해양대학교 부교수
※관심분야 : 데이터베이스, 해양정보시스템, 데이터마이닝, XML