단일 스캔을 통한 웹 방문 패턴의 탐색 기법
(An Efficient Approach for Single-Pass Mining of Web Traversal Sequences)

김 낙 민 † 정 병 수 †† 아메드 파한 †
(Nakmin Kim) (Byeong-Soo Jeong) (Chowdhury Farhan Ahmed)

요 약 인터넷 사용의 급증과 더불어 보다 편리한 인터넷 서비스를 위한 여러 연구가 활발히 진행되어 왔다. 웹 로그 데이터로부터 빈번하게 발생되는 웹 페이지들의 방문 시퀀스를 탐색하는 기법 역시 효과적인 웹 사이트를 설계하기 위한 목적으로 많이 연구되어 왔다. 그러나 기존의 방법들은 모두 여러 번의 데이터베이스 스캔을 필요로 하는 방법으로 지속적으로 생성되는 웹 로그 데이터로부터 빠르게 실시간적으로 웹 페이지 방문 시퀀스를 탐색하기에는 많은 어려움이 있었다. 또한 점진적(incremental)이고 대화형식(interactive)의 탐색 기법 역시 지속적으로 생성되는 웹 로그 데이터를 처리하기 위하여 필요한 기능들이다. 본 논문에서는 지속적으로 생성되는 웹 로그 데이터로부터 단일 스캔을 통하여 빈번히 발생하는 웹 페이지 방문 시퀀스를 점진적이고 대화 형식적인 방법으로 탐색하는 방법을 제안한다. 제안하는 방법은 WTS(web traversal sequence)-트리 구조를 사용하며 다양한 실험을 통하여 기존의 방법들에 비해 성능적으로 우수하고 효과적인 방법임을 증명한다.

키워드 : 데이터 마이닝, 웹 마이닝, 웹 방문 시퀀스, 높은 유틸리티 패턴, 데이터 스트림

Abstract Web access sequence mining can discover the frequently accessed web pages pursued by users. Utility-based web access sequence mining handles non-binary occurrences of web pages and extracts more useful knowledge from web logs. However, the existing utility-based web access sequence mining approach considers web access sequences from the very beginning of web logs and therefore it is not suitable for mining data streams where the volume of data is huge and unbounded. At the same time, it cannot find the recent change of knowledge in data streams adaptively. The existing approach has many other limitations such as considering only forward references of web access sequences, suffers in the level-wise candidate generation-and-test methodology, needs several database scans, etc. In this paper, we propose a new approach for high utility web access sequence mining over data streams with a sliding window method. Our approach can not only handle large-scale data but also efficiently discover the recently generated information from data streams. Moreover, it can solve the other limitations of the existing algorithm over data streams. Extensive performance analyses show that our approach is very efficient and outperforms the existing algorithm.

Key words : Data mining, web mining, web access sequences, high utility patterns, data streams

† 비 회 원 : 경희대학교 컴퓨터공학과
ifridx@naver.com
farhan@khu.ac.kr
†† 종신회원 : 경희대학교 컴퓨터공학과 교수
jeong@khu.ac.kr
논문접수 : 2010년 8월 19일
심사완료 : 2010년 8월 26일

정보과학회논문지 : 데이타베이스 제37권 제5호(2010.10)

1. 서 론

인터넷의 사용이 급증하면서 웹을 활용하는 범위도 지속적으로 확대되고 있다. 이에 따라 보다 지능적인 웹 서비스를 위하여 웹 로그 데이터로부터 유용한 지식을 추출하는 데이터 마이닝 기술이 활발히 연구되어 왔다. 이러한 기술을 특히 웹 마이닝[1-4]이라 하며 세분하여 보면 웹의 내용(contents)에 대한 분석, 구조(structure)에 대한 분석, 그리고 웹의 사용(usage)에 대한 분석 부문으로 나누어 볼 수 있다[5]. 사용자들의 웹 페이지 방문 시퀀스에 대한 탐색은 웹의 사용에 대한 마이닝 기술로 효과적인 웹 사이트의 설계, 상호 관련도가 높은

웹 페이지의 탐지, 다음에 방문할 웹 페이지의 추천 등과 같은 응용에 중요하게 사용될 수 있다. 아울러 웹 페이지의 방문 경향이 유사한 사용자들을 그루핑 한다든지 사용자들을 방문 성향에 따라 적절히 분류하여 여러 다양한 목적으로 활용할 수도 있다.

한편 웹 로그 데이터는 매일 지속적으로 생성되는 대용량이 자료이므로 빠른 처리를 위하여 한번의 데이터베이스 스캔을 사용하는 탐색 방법이 요구되고 또한 최근 자료에 대한 정보가 중요하므로 점진적인 탐색 기법이 요구된다. 여기서 점진적인 탐색 기법이란 웹 로그 데이터가 증가될 경우 탐색을 위하여 전체 웹 로그 데이터를 다시 처리하는 것이 아니라 이미 만들어 놓은 자료 구조 위에 증가된 웹 로그 데이터를 점진적으로 추가하여 사용할 수 있는 탐색 기법을 말한다. 아울러 응용 목적에 따라 여러 다른 임계 값을 사용하여야 할 경우에도 기존 자료 구조를 그대로 사용할 수 있는 대화 형식적인 탐색 기법이 요구된다[1].

단일 스캔을 통하여 웹 페이지 방문 시퀀스를 탐색하는 방법으로 DSM-PLW[3]이 제안되었으나 이 방법의 제약 사항은 웹 페이지 방문 시퀀스에서 순방향 방문(forward reference)만을 고려하고 역방향 방문(backward reference)을 처리할 수 없다는 것이다. 역방향 방문이란 $W_1W_3W_5W_1W_2..(W_i$는 웹 페이지)와 같은 웹 페이지 방문에서 W_1을 방문하고 다시 W_1을 방문하는 것을 말한다. DSM-PLW 알고리즘은 일련의 웹 페이지 방문 순서에서 역방향 방문을 제거한 순방향 방문 순서들만을 대상으로 하여 빈번히 발생하는 웹 페이지 방문 순서 패턴을 찾는다.

본 논문에서는 기존의 WAP(web access pattern)-트리를 확장하여 역방향 방문 순서도 포함될 수 있는 단일 스캔 웹 페이지 방문 순서 탐색 기법을 제안한다. 제안하는 탐색 기법은 웹 로그 데이터를 단일 스캔하여 WTS(web traversal sequence)-트리 구조를 만들고 이 WTS 트리 구조를 통하여 빈번하게 발생하는 웹 페이지 방문 순서를 찾는다. 제안하는 탐색 기법은 또한 점진적이고 대화 형식적인 탐색 기법으로도 활용될 수 있다. 제안하는 기법의 성능적 우수성을 입증하기 위하여 최근 기법인 IncWTP[1] 알고리즘과 다양한 데이터 집합에 대하여 성능 실험을 하였다.

본 논문의 구성은 2장에서 관련된 기존의 연구 결과를 요약하고 3장에서는 문제 정의와 함께 제안한 기법을 소개한다. 4장에서는 성능 실험에 대한 결과를 요약하고 5장에서 결론을 맺는다.

2. 관련 연구

WWW에서 이루어지는 여러 형태의 트랜잭션들에서 수집되는 데이터들에 대하여 데이터 마이닝 기술을 활용하여 보다 편리한 웹 서비스를 제공하기 위한 연구가 오래 전부터 활발히 진행되어 왔다. WEBMINER[6] 시스템은 웹 로그 데이터로부터 추출한 정보(연관 규칙, 빈번한 방문 패턴)들을 SQL 형태의 질의로 검색할 수 있는 웹 분석 도구이다. WEBMINER에서는 웹 분석 도구로서 전체적으로 갖추어야 할 기능들과 이들을 구현하는 소프트웨어 구조를 소개하고 있다. 유사한 연구로 WUM[10]에서는 사용자가 관심이 있는 웹 사용 패턴을 MINT라는 마이닝 언어로 명시하면 이를 바탕으로 동적으로 유용한 웹 사용 패턴을 검색할 수 있도록 하는 웹 분석 도구이다.

위에 언급한 연구들은 웹 로그 데이터로부터 특정 패턴을 효율적으로 탐색하는 기법에 대한 것이라기 보다는 웹 마이닝 도구의 프로토타입 형태로 웹 환경에서 마이닝 기술이 활용되기 위한 필요 기능과 앞으로 해결해야 할 연구 과제들을 정리한 것이라 할 수 있다. 웹 페이지의 방문 패턴에 대한 본격적인 연구 결과로는 Chen[7]이 제안한 두 알고리즘, 완전 스캔(full-scan) 알고리즘과 선택적-스캔(selective-scan)으로 선택적 스캔 방법에서 불필요한 DB 스캔을 제거하여 성능을 개선할 수 있는 방안을 제시하고 있다. 그러나 Chen의 방법은 순방향 방문만을 고려하는 것으로 방문 시퀀스에 역방향 방문이 포함된 시퀀스의 처리에는 적합하지 못하다.

빈번한 시퀀스 패턴의 탐색은 Apiriori-원칙을 이용하여 단계별로 후보 시퀀스를 생성하고 이를 시퀀스 DB의 스캔을 통하여 검증하는 방법이 많이 제안되었으나 이러한 방법은 여러 번의 DB 스캔을 필요로 하기 때문에 DB의 크기가 커지면 성능이 떨어지는 문제점을 가지고 있다. 이러한 문제를 해결한 기법으로는 WAP-트리[9] 구조를 이용하여 빈번한 시퀀스의 탐색을 순차적 패턴 확장(sequential pattern growth)으로 처리하는 WAP-mine 알고리즘이 있다. 그러나 WAP-mine 알고리즘 역시 두 번의 DB 스캔이 요구되는 방법으로 정적인(static) DB를 가정하고 있기 때문에 웹 로그와 같이 지속적으로 생성되는 동적인 DB를 처리하기에는 적합하지 않다.

웹 로그와 같은 동적인 DB를 고려한 점진적이고 대화 형식적인 웹 페이지 방문 시퀀스 탐색 기법으로 최근 발표된 IncWTP[1] 알고리즘을 들 수 있다. 그러나 이 방법 역시 기본적으로는 단계별로 후보군을 생성-검증하는 방식을 취하고 있기 때문에 시퀀스 DB의 크기와 후보군의 크기가 커지게 되면 성능적인 문제를 나타낼 수 있다. 위에 언급된 여러 문제점들을 해결할 수 있는 탐색 기법으로 본 논문에서는 동적인 웹 로그 데이

터로부터 시퀀스 DB의 단일 스캔을 통한 효과적인 점진적 그리고 대화 형식적인 빈번한 시퀀스 탐색 기법을 제안한다.

3. 단일 스캔 웹 방문 패턴 탐색 기법

3.1 문제 정의

웹 페이지 방문 패턴 탐색의 문제는 순차적 패턴의 탐색과 같은 문제로 다음과 같이 정의될 수 있다. 우선 W (= {w_1, w_2, w_3, ···, w_n})를 방문 가능한 웹 페이지들의 집합이라 정의하고 S_i 를 웹 페이지의 방문 순서로 이루어진 시퀀스(sequence)라 하면 S_i 는 [$w_1w_2\cdots w_k$] ($w_j \in W$, $1 \le j \le k$)로 나타낼 수 있다. 방문 시퀀스 S_i 에는 같은 웹 페이지가 여러 번 나타날 수 있으며(그림 1의 S_1과 S_3 시퀀스에서 웹 페이지 a의 경우), $|S_i|$는 방문 시퀀스의 길이(length)라 하고 방문 시퀀스에 나타난 웹 페이지의 전체 개수로 나타낸다. 두 개의 방문 시퀀스 α = [$a_1a_2a_3\cdots\cdots a_p$], β = [$b_1b_2b_3\cdots\cdots b_q$] ($p \le q$)에 대하여 $b_{i1} = a_1$, $b_{i2} = a_2$, ……, $b_{im} = a_m$ ($i_1 \le i_2 \le \cdots\cdots \le i_p$)을 만족하는 경우 α를 의 서브 시퀀스(sub-sequence), β를 α의 슈퍼 시퀀스(super-sequence)라 하며, 또한 β가 α를 포함(contain)한다고 정의한다. 예를 들면 그림 1의 S_3에서 *'ae'*는 *'abea'*의 서브 시퀀스이다.

시퀀스 데이터베이스(SDB)는 웹 로그 데이터로부터 생성한 대용량의 웹 페이지 방문 시퀀스(WAS: Web Access Sequence)들로 구성된다. 시퀀스 S_i의 빈도수(frequency)는 SDB에서 S_i를 포함하고 있는 WAS의 개수를 의미한다. 예를 들면 그림 1의 SDB에서 S_1, S_3, S_4, S_5, 그리고 S_6가 시퀀스 *'ab'*를 포함하므로 시퀀스 *ab*의 빈도수는 *5*가 된다. 하나의 WAS에 해당 시퀀스가 여러 번 나타나는 경우에도 시퀀스의 빈도수는 하나로 계산한다. 예를 들면, 시퀀스 S_1에서 *'a'* 시퀀스는 2번 나타나지만 *'a'* 시퀀스의 빈도수 계산시 한번만 적용한다. 시퀀스 최소 임계 값(minimum threshold) δ는 전체 WAS 개수(즉 |SDB|)에서 특정 시퀀스가 나타나는 비중(percentage)을 나타내며 최소 빈도수(minimum support)는 최소 임계 값에 전체 WAS의 수를 곱한 값의 상한 정수 값(ceiling integer)으로 나타낸다. 예를 들면 δ를 50%로 하면 최소 빈도수는 (50% × 7 = 3.5) 4가 된다. SDB에서 최소 빈도수 보다 많이 나타나는 시퀀스를 δ-시퀀스라 하며, 빈번한 웹 페이지 방문 시퀀스의 탐색은 주어진 SDB와 δ값에 대하여 모든 δ-시퀀스들을 찾는 문제로 정의할 수 있다.

Seq. ID	웹 페이지 방문 시퀀스
S_1	a b f a c
S_2	e d a
S_3	a b e a c e
S_4	a b c b
S_5	a b e
S_6	e d a e a b
S_7	a c e

그림 1 웹 페이지 방문 시퀀스 DB의 예

3.2 WTS-트리의 구조 및 생성 방법

WTS-트리는 Prefix-트리의 형태로 트리의 각 노드는 웹 페이지의 번호(id)와 SDB의 WAS에 나타난 웹 페이지의 빈도수를 나타낸다. 또한 각 노드에 저장된 빈도수 값은 루트 노드로부터 해당 노드까지의 시퀀스 빈도수를 나타낼 수 있다. 즉, 그림 2(c)의 맨 왼쪽 가지의 경우, 시퀀스 *'abcb'*의 빈도수는 1, *'abc'*의 빈도수는 1, *'ab'*의 빈도수는 4, *'a'*의 빈도수는 5를 각각 나타낸다. 그림 1의 SDB에 나타난 각각의 웹 페이지 방문 시퀀스를 차례로 읽어 WTS-트리에 삽입하면 그림 2에서와 같은 순서로 WTS-트리가 만들어진다. 첫 번째로 S_1 *('abfac')*를 읽어 WTS-트리에 삽입하면 그림 2(a)와 같은 WTS-트리의 모습이 된다. 두 번째로 S_2 *('eda')*를 삽입하면 루트 노드로부터 기존의 웹 페이지 번호를 비교하면서 같은 방법으로 삽입된다. 트리의 각 노드에는 웹 페이지의 번호(id)와 방문된 빈도수가 저장된다. 그리고 왼쪽의 헤더-테이블(H-table)에는 각 웹 페이지

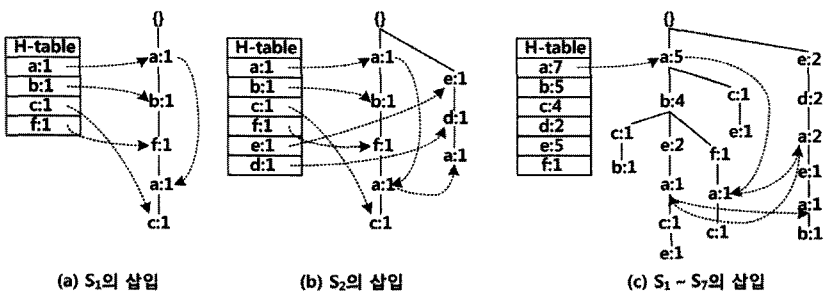


그림 2 WTS-트리의 생성 과정

번호(id)와 해당 페이지의 누적된 방문 빈도수를 기록한다. 이때 한 시퀀스에 여러 번 나타난 웹 페이지의 빈도수는 한 번만 계산한다. 즉 *'abfac'*에서 a는 한 번만 계산된다. 또한, WTS-트리에서 같은 웹 페이지 번호(id)들은 헤더 테이블의 웹 페이지 번호(id)를 시작으로 하여 나타난 순서대로 포인터로 연결이 된다. 그림 2(c)는 그림 1의 시퀀스 데이터베이스를 WTS-트리에 삽입한 최종 모습이다(노드들 간의 포인터는 편의상 a에 대한 것만 표시함).

WTS-트리의 구조는 [8]에서 제안한 WAP-트리와 유사하나 기본적인 차이점은 WAP-트리는 트리를 구성하기 전에 먼저 시퀀스 DB를 한 번 스캔하여 방문 빈도수가 높은 웹 페이지를 파악한 다음 빈도수가 높은 웹 페이지들로만 구성된 시퀀스들을 트리로 삽입하는데 반하여 WTS-트리는 모든 웹 페이지 시퀀스들을 한 번에 트리로 삽입한다는 것이다. WAP-트리는 빈도수가 높은 웹 페이지들만을 고려하므로 트리의 크기가 작아지는 장점은 있는 반면, 시퀀스 DB를 두 번 스캔하여야 하는 문제점이 있고 또한 빈도수에 대한 임계값이 변경될 경우 트리를 재구성해야 하는 문제점이 있다.

이와는 반대로 본 논문에서 제안하는 WTS-트리는 시퀀스 DB에 대한 단일 스캔을 통하여 트리를 구성할 수 있고 다양한 빈도수 임계값에 대해서도 하나의 트리 구조를 가지고 방문 패턴을 탐색할 수 있는 장점을 지니고 있다. 제안한 트리 구조의 문제점으로는 트리의 크기가 발생 가능한 모든 방문 시퀀스의 수에 비례하여 커지는 것을 들 수 있는 데, 실제 환경에서는 모든 방문 시퀀스가 발생할 가능성이 극히 작고 또 방문 시퀀스의 길이를 10~20 페이지 수로 제한한다면 GB급의 메모리로 충분히 처리될 수 있음이 실험 과정에서 확인되었다.

위에서 설명한 WTS-트리의 생성 과정을 알고리즘으로 나타내면 다음과 같다.

알고리즘 1: WTS-트리의 생성
입력: 시퀀스 DB (SDB), *SDB*
출력: WTS-트리, T
단계작업:

1. T의 루트 노드를 생성하고, *current_node*를 루트 노드로 지정한다.
2. *SDB*의 모든 시퀀스, S (= $s_1s_2s_3...s_n$)의 s_i에 대하여 다음 작업을 수행
   FOR i = *1* to n DO
   IF *current_node*가 s_i에 해당하는 *child* 노드를 가지고 있다면
   THEN *current_node*를 s_i 노드로 지정하고 s_i 노드의 *count* 를 *1* 증가시킨다.
   ELSE (s_i:1) 노드를 새로 생성하고 (s_i:1) 노드를 *current_node*로 지정한다.
   (s_i : 1) 노드를 (s_i) 리스트에 연결한다.
3. WTS-트리, T를 반환한다. ■

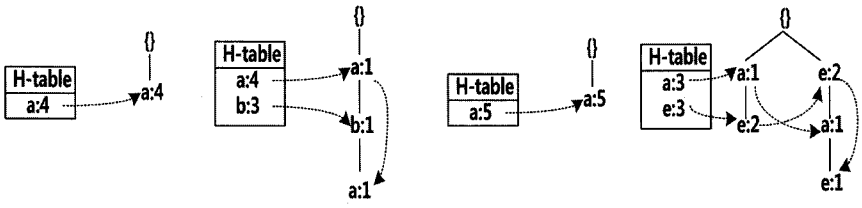
3.3 빈번한 웹 페이지 방문 패턴의 탐색 과정

제안하는 기법을 사용하여 빈번히 발생하는 WTS(web traversal sequence)의 탐색을 위해서는 우선 최소 임계값(δ)이 주어져야 하고 이에 따라 최소 빈도수(minimum support)가 정해져야 한다. 예를 들면, 그림 1의 시퀀스 DB를 가지고 생성한 그림 2(c)의 WTS-트리에 대하여 최소 임계값(δ=40%)에 대한 δ-시퀀스(즉 δ 값 이상의 빈번한 시퀀스)는 δ = 40%이므로 발생 빈도수가 3 (40% × 7 = 2.8) 이상인 시퀀스가 된다.

빈번한 시퀀스 패턴의 탐색은 Apiriori-원리[7]에 따른 순차적 패턴 확장(sequential pattern growth) 방식을 사용하며 헤더 테이블 하단부터 최소 빈도수 이상의 발생 빈도를 갖는 웹 페이지 번호에 대하여 차례대로 순차적 패턴 확장 기법을 적용하게 된다. 그림 2(c)의 WTS-트리의 경우 헤더 테이블에서 해당 페이지는 'e'가 되고 WTS-트리를 참조하여 'e'의 모든 이전 시퀀스(prefix sequence)들의 빈도수를 계산하고 'e'에 대한 조건적(conditional) WTS-트리를 구성하게 된다.

그림 2(c)에서 헤더 테이블의 'e'에 대한 포인터를 활용하여 이전 시퀀스(prefix sequence)를 찾으면 <ab: 2, abeac: 1, ac: 1, eda:1, ab: -1>가 된다. 여기서 유의할 것은 'ab'와 'abeac'에서 'ab'가 두 번 계산되었으므로 마지막에 'ab: -1'을 추가해야 한다는 것이다. 이렇게 되면 'e'의 이전 시퀀스(prefix sequence)들 중에서 최소 빈도수 3 이상인 웹 페이지는 <a: 4>만 남아 그림 3(a)와 같은 조건적 WTS-트리가 되고 이 과정에서 <e: 5, ae: 4>가 빈번한 시퀀스로 탐색된다. 계속해서 'c'를 적용하면('f'와 'd'는 최소 빈도수 3보다 작으므로 생략), <ab: 1, abea: 1, abfa: 1, a: 1>가 'c'에 대한 이전 시퀀스(prefix sequence)로 파악되고 이들 중 최소 빈도수 3보다 큰 것은 'a'와 'b'로 이들로 구성된 조건적 WTS-트리가 그림 3(b)과 같이 만들어질 수 있다. 마찬가지 방법으로 'b'와 'a'에 대해서도 적용하면 각각 그림 3(c)와 그림 3(d)와 같이 되고 최종적으로 δ = 40%에 대한 δ-시퀀스는 <e: 5, ae: 4, c: 4, ac: 4, bc: 3, abc: 3, b: 5, ab: 5, a: 7, aa: 3, ea:3>가 탐색된다. 조건적 WTS-트리를 만드는 과정은 순환적으로(recursively) 더 이상 조건적 WTS-트리가 나오지 않을 때까지 계속된다. 알고리즘 2는 빈번한 시퀀스의 탐색 과정을 나타낸 것이다.

알고리즘 2: δ-시퀀스의 탐색
입력: WTS-트리, T와 최소 임계값(δ)
출력: δ-시퀀스의 집합, £
단계작업:



(a) "e"의 조건적 WTS-트리 (b) "c"의 조건적 WTS-트리 (c) "b"의 조건적 WTS-트리 (d) "a"의 조건적 WTS-트리

그림 3 탐색 과정에서의 조건적 WTS-트리 생성

1. δ-시퀀스의 집합, £ = ϕ로 초기화한다.
2. WTS-트리, T의 헤더 테이블에서 최소 임계값(δ) 이상의 빈도수를 갖는 모든 s_i를 £에 삽입한다.
3. FOR each s_i in £ DO
   A. 조건적 시퀀스 기반, $\theta \mid s_i$을 구성한다.
   B. $\theta \mid s_i$을 대상으로 알고리즘 1을 사용하여 s_i의 조건적 WTS-트리를 생성한다.
   C. s_i의 조건적 WTS-트리를 가지고 알고리즘 2를 순환적으로 수행한다.
   s_i의 조건적 WTS-트리가 empty이면 리턴한다.
   D. s_i의 조건적 WTS-트리를 가지고 탐색한 모든 결과에 s_i를 연결하여 £에 삽입한다.
4. £를 반환한다. ■

3.4 점진적 탐색 및 대화 형식적 탐색

웹 방문 패턴 탐색에 사용되는 웹 로그 데이터는 시간에 따라 점진적으로 증가하게 되는 동적인 특성을 갖고 있기 때문에 웹 방문 패턴 탐색 기법 역시 점진적인 탐색 기능을 제공할 수 있는 기법이 되어야 할 것이다. 기존의 WAP-트리를 이용하는 탐색 기법은 먼저 빈번히 방문되는 웹 페이지들을 추출하기 위하여 시퀀스 DB에 대한 스캔이 요구되고 다시 WAP-트리를 구성하기 위하여 또 한 번의 DB 스캔이 필요하므로 점진적인 DB 증가가 이루어지는 환경에서는 사용할 수 없게 된다. 이에 반하여 제안하는 기법에서는 한 번의 DB 스캔으로 WTS-트리가 구성되므로 시퀀스 DB가 증가하는 경우에도 기존의 WTS-트리 위에 새로운 DB의 시퀀스들을 쉽게 삽입할 수 있게 된다.

대화 형식적 탐색 기능이란 "*build once mine many*" 성질을 갖고 있는 탐색 기법을 의미하는 것으로 한 번의 WTS-트리 생성으로 트리의 재구성 없이 다른 최소 임계값에 대하여 여러 번의 탐색 작업을 대화 형식적으로 할 수 있는 기능을 말한다. 제안하는 기법은 한 번의 DB 스캔으로 DB의 모든 시퀀스 정보가 WTS-트리에 저장되므로 여러 δ 값에 대한 반복적인 시퀀스 탐색이 가능할 뿐만 아니라 이전에 수행된 탐색 결과를 이용하여 탐색 시간을 빠르게 할 수 있는 기능도 제공할 수 있다.

예를 들면, δ = 70% 최소 임계값(최소 빈도수는 70% * 7 = 4.9 이므로 5)에 대한 δ-시퀀스의 탐색이 끝난 후 δ = 70% 보다 크거나 작은 두 개의 다른 δ' 값에 대하여 δ'- 시퀀스를 탐색하는 경우, 만약 δ' 값이 δ = 70% 보다 크다면 δ'-시퀀스는 모두 δ-시퀀스에 속해 있으므로 WTS-트리를 재 탐색할 필요 없이 δ-시퀀스에서 δ'의 최소 빈도수 값 이상의 빈도수를 나타내는 시퀀스들만을 선택하면 될 것이다. 이와 반대로 δ' 값이 δ = 70% 보다 작다면 δ-시퀀스는 모두 δ'-시퀀스가 될 수 있다.

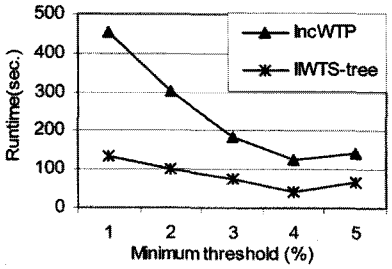
웹 로그 데이터로부터 탐색되는 웹 페이지 방문 패턴의 정보는 특정 시간대 혹은 최근 시점의 패턴 정보들이 유용하게 사용될 수 있다. 최근 시점에 대한 패턴 탐색을 위해서는 슬라이딩 윈도우 모델을 기반으로 하는 탐색 기법이 요구되는데, WTS-트리에 대한 슬라이딩 윈도우 탐색 모델은 [9]에서 제안된 슬라이딩 윈도우 탐색 모델을 그대로 적용하여 사용할 수 있다.

4. 성능 실험

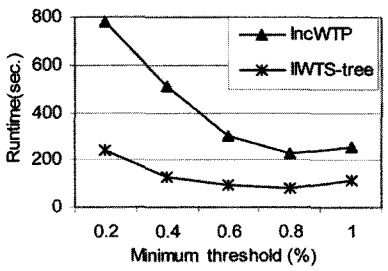
제안하는 기법의 성능 평가를 위해서 [7]에서 언급된 원칙에 따라 생성한 가상의 데이터 집합[10]을 사용하였다. 이와 같은 형태의 데이터 집합은 기존의 웹 방문 시퀀스 연구[2,11-13]에서도 사용되었던 것이다. 데이터 집합의 생성에 사용된 매개 변수의 내용은 표 1과 같다. 데이터 집합은 C10.S5.N50.D100K와 C15.S5.N100.D200K을 각각 생성하여 가장 최근에 발표된 IncWTP[1] 알고리즘과 제안된 기법의 처리 시간을 최소 임계값 혹은 시퀀스의 개수 별로 비교하여 보았다. 실험 환경은 2GB 메모리의 펜티움 듀얼 코어 2.13 GHz, Windows XP에서 Microsoft C++ 6.0을 사용하였다.

표 1 데이터 집합 생성을 위한 매개 변수

매개 변수	내 용
D	웹 시퀀스의 개수
C	시퀀스의 평균 길이
S	빈번한 시퀀스의 평균 길이
N	웹 페이지 개수

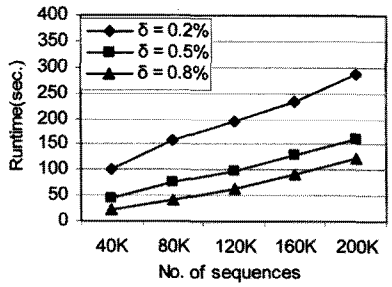


(a) 데이터[C10.S5.N50.D100K]의 처리 시간

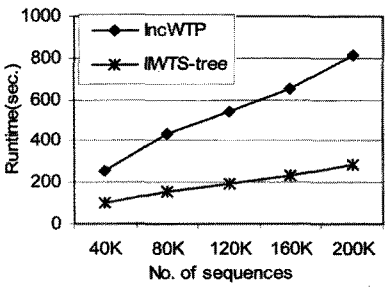


(b) 데이터[C10.S5.N50.D100K]의 처리 시간

그림 4 최소 임계값에 따른 처리 시간의 변화



(a) 데이터[C10.S5.N50.D100K]의 처리 시간



(b) 데이터[C10.S5.N50.D100K]의 처리 시간

그림 5 시퀀스 DB의 크기에 따른 처리 시간의 변화

그림 4(a)와 그림 4(b)는 최소 임계값을 변경하면서 IncWTP 알고리즘과 제안한 기법의 탐색 처리 시간을 두 데이터 집합에 대하여 각각 나타낸 것이다. 그림 4에서 알 수 있듯이 전체적으로 제안한 기법이 좋은 성능을 보이고 있다. 이러한 이유는 제안한 기법은 한 번의 시퀀스 DB 스캔을 통하여 탐색이 이루어지는 반면 IncWTP에서는 단계별로 후보군(candidates)을 생성하고 검증하는 과정에서 여러 번의 DB 스캔이 필요하기 때문인 것으로 파악된다. 이러한 성능의 차이는 최소 임계값이 적어 후보군이 많아질수록 더욱 두드러진다.

그림 5(a)는 시퀀스 DB의 크기가 점진적으로 증가하는 상황에서 제안한 기법의 처리 시간을 서로 다른 최소 임계값에 대하여 실험한 결과를 나타낸다. 전체적으로 제안한 기법은 시퀀스 DB의 크기가 점진적으로 증가하는 경우에도 처리 시간의 증가가 그림 5(b)의 IncWTP 알고리즘에 비해 비교적 크지 않은 것을 알 수 있다. 이는 전체 시퀀스 DB를 다시 스캔하여야 하는 IncWTP 알고리즘과는 달리 제안한 기법에서는 증가된 DB만을 기존 WTS-트리에 삽입함으로써 빈번한 시퀀스의 탐색이 가능하기 때문이다. 또한 최소 임계값에 따른 처리 시간의 차이는 WTS-트리에서 각각의 웹 페이지에 대한 조건적 WTS-트리를 순환적으로 생성하는 시간의 차이에 따른 것이다.

5. 결 론

본 논문은 웹 로그 데이터로부터 추출한 웹 페이지 방문 시퀀스 DB를 단 한번의 DB 스캔을 통하여 빈번히 발생하는 웹 페이지 방문 시퀀스에 대한 효율적인 탐색 기법을 제안하였다. 제안한 기법은 역방향 방문 및 순방향 방문 모두를 포함할 수 있는 시퀀스 탐색 기법으로 기존의 IncWTP 알고리즘에 비해 좋은 성능을 보일 수 있음을 실험을 통해 증명하였다. 제안한 기법은 또한 시퀀스 DB가 동적으로 변화하는 환경에서도 IncWTP 알고리즘에 비해 확장성이 좋은 것으로 나타났다. 향후 추가적으로 연구할 사항으로는 시간 정보를 함께 고려하는 시퀀스 패턴을 탐색하는 기법 및 웹 페이지의 중요도에 따라 시퀀스의 비중을 다르게 고려하는 탐색 모델을 정립하는 것 등을 들 수 있다.

참 고 문 헌

[1] Y.-S. Lee, S.-J. Yen, "Incremental and interactive mining of web traversal patterns," In *Information Sciences*, vol.178, pp.287-306, 2008.

[2] Y.-S. Lee, S.-J. Yen, G.H. Tu and M.C. Hsieh, "Web usage mining: Integrating path traversal patterns and association rules," In *International Conference on Informatics, Cybernetics, and Sys-*

terns, pp.1464-1469, 2003.
[3] H.-F. Li, S.-Y. Lee and M.-K. Shen, "DSM-PLW: Single-pass mining of path traversal patterns over streaming web click-sequences," In *Computer Networks*, vol.50, pp.1474-1487, 2006.
[4] B. Mobasher, N. Jain, E.-H. Han, J. Srivastava, "Web mining: Pattern discovery from World Wide Web transactions," In *Tech Rep*: TR96-050, 1996.
[5] R. Cooley, B. Mobasher and J. Srivastava, "Web mining: Information and pattern discovery on the world wide web," In *IEEE International Conference on Tools with Artificial Intelligence*, pp.558-567, 1997.
[6] M. Spiliopoulou, and L. C. Faulstich, "Wum: A web utilization miner," In *EDBT Workshop WebDB98*, Springer Verlag, pp.109-115, 1996.
[7] R. Agrawal, R. Srikant, "Mining Sequential Patterns," In *IEEE International Conference on Data Engineering (ICDE)*, pp.3-14, 1995.
[8] J. Pei, J. Han, B. Mortazavi-asl and H. Zhu, "Mining access patterns efficiently from web logs," In *Pacific-Asia Conf. on Knowledge Discovery and Data Mining* (PAKDD), pp.396-407, 2000.
[9] Syed Khairuzzaman Tanbeer, Chowdhury Farhan Ahmed, Byeong-Soo Jeong, and Young-Koo Lee, "Sliding Window-based Frequent Pattern Mining over Data Streams," *Information Sciences*, vol.179, Issue 22, pp.3843-3865, 2009.
[10] http://www.almaden.ibm.com/cs/projects/iis/hdb/Projects/data mining/datasets/syndata.html
[11] C.I. Ezeife, Y. Lu, "Mining web log sequential patterns with position coded pre-order linked WAP-tree," In *Data Mining and Knowledge Discovery*, vol.10, pp.53-87, 2005.
[12] S. Yang, J. Guo and Y. Zhu, "An efficient algorithm for web access pattern mining," In *International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp.726-729, 2007.
[13] S.J. Yen, Y.S. Lee, C.W. Cho, "Efficient approach for the maintenance of path traversal patterns," In *IEEE International Conference on e-Technology, e-Commerce and e-Service*, pp.207-214, 2004.
[14] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan, "Web usage mining: discovery and applications of usage patterns from web data," In *SIGKDD Explorations*, vol.1, no.2, pp.12-23, 2000.
[15] W. Wang and P. T. Cao-Thai, "Novel position-coded methods for mining web access patterns," In *IEEE International Conference on Intelligence and Security Informatics(ISI)*, pp.194-196, 2008.
[16] B. Zhou, S. C. Hui and A. Fong, "CS-Mine: An efficient wap-tree mining for web access patterns," In *International Asia-Pacific Web Conference (APWeb)*, pp.523-532, 2004.

김 낙 민
2009년 경희대학교 전자정보대학 컴퓨터공학과 졸업(학사). 2009년~현재 경희대학교 전자정보대학 컴퓨터공학과 석사과정. 관심분야는 데이터베이스, 데이터 마이닝

정 병 수
1983년 서울대학교 공과대학 컴퓨터공학과 졸업(학사). 1985년 한국과학기술원 전산학과(석사). 1995년 Georgia Institute of Technology, College of Computing (박사). 1985년~1989년 한국 데이터통신㈜ 선임연구원. 1995년~1996년 Georgia Institute of Technology, PostDoc. 1996년~현재 경희대학교 전자정보대학 컴퓨터공학과 교수. 관심분야는 데이터베이스, 데이터 마이닝, 모바일 컴퓨팅

아메드 파한
2004년 방글라데시 다카 대학 컴퓨터공학과 졸업(학사). 2006년 방글라데시 다카 대학 컴퓨터공학과(석사). 2006년~2007년 방글라데시 다카 대학 컴퓨터공학과 전임강사. 2007년~현재 경희대학교 전자정보대학 컴퓨터공학과 박사과정. 관심분야는 데이터 마이닝, 패턴 탐색, 모바일 컴퓨팅