연관규칙을 이용한 적응형 학교 웹사이트 구축 알고리즘

이정민°, 전우천
용인고기초등학교°, 서울교육대학교 컴퓨터교육과
sutch24@hanmir.com°, wocjun@snue.ac.kr

An Algorithm for Adaptive School Web Site Construction Using Association Rules

Jeong-Min Lee°, Woo-Chun Jun
Yongin Kogi Elementary School°, Dept. of Computer Education, Seoul National University of Education

요 약

최근에 학교 현장에서 제공하는 홈페이지는 학교의 정보화 능력을 가늠하는 척도가 되고 있으며 학생과 학부모 그리고 학교가 상호 의사소통 할 수 있는 좋은 장을 마련해주고 있다. 그러나 끊임없이 변화하는 학생들의 검색 패턴에 대해서 학교 홈페이지가 적절히 대처하지 못하고 있으며, 그들의 방문 목적 달성을 위한 충분한 안내를 제공함에 있어 한계를 가지고 있는 것이 사실이다. 본 논문에서는 사이트 접속자들의 행동 패턴 분석을 위해 웹서버 로그 데이터를 이용하고, 데이터 마이닝의 한 기법인 연관규칙을 적용하여 로그 데이터를 분석함으로써 사용자들의 의미 있는 행동패턴을 추출하는 알고리즘을 제안하였다. 이렇게 추출된 행동패턴을 기반으로 하이퍼링크가 자동으로 생성되어 해당 웹페이지에 삽입됨으로써 특정 개인뿐만 아니라 공통의 다수가 편리하게 이용할 수 있는 적응형 학교 웹사이트 구축 방안을 제시한다.

1. 서 론

최근 학교 홈페이지는 학교의 또 다른 얼굴로 간주되며, 학교의 정보화 수준을 가늠하는 중요한 척도가 되고 있다. 그러나 학교 홈페이지는 제작 당시 네비게이션을 충분히 고려했다 하더라도 해당 웹페이지에 익숙하지 못한 사용자나, 길을 잃기 쉬운 학생들에게 친절한 안내를 제공하는 것에 있어 한계를 가진다. 이것은 사용자들마다 각기 다른 목적을 갖고 사이트에 접근하며, 동일 사용자라 하더라도 접속 때마다 다른 서비스를 원할 수 있기 때문이다[1]. 더구나 홈페이지 담당 교사에게 학교 홈페이지 접속자들의 행동패턴을 일일이 분석해서 그때마다 수작업으로 웹사이트 갱신을 요구하는 것은 힘든 일이다.

만일 학교 홈페이지가 사이트 접속자들의 행동 패턴을 분석하여 자동적으로 편리한 네비게이션을 만들어 준다면, 접속자들에게는 친절하고 효과적인 웹사용 경험을 제공할 수 있으며, 홈페이지 담당자에게는 지속적인 갱신의 부담을 경감시켜 줄 수 있을 것이다. 이렇게 사용자들의 접근 패턴을 학습하여 외형이나 조직을 자동적으로 개선시키는 웹사이트를 적응형 웹 사이트 (Adaptive Web Site)라고 한다[2].

본 연구에서는 사이트 접속자들의 행동패턴을 분석하여 웹사이트의 구조나 외형을 자동적으로 개선시켜 나가는 적응형 웹사이트 구축 방안을 제시한다. 즉, 사이트 접속자들의 행동 패턴 분석을 위해 웹서버 로그 데이터를 이용하고, 데이터 마이닝의 한 기법인 연관규칙을 적용하여 로그 데이터를 분석함으로써 사용자들의 의미 있는 행동패턴을 추출하는 알고리즘을 제안하였다. 이렇게 추출된 행동패턴을 기반으로 하이퍼링크가 자동으로 생성되어 해당 웹페이지에 삽입됨으로써 특정 개인뿐만 아니라 공통의 다수가 편리하게 이용할 수 있는 적응형 웹사이트가 구축된다.

본 논문의 구성은 다음과 같다. 2 장에서는 데이터 마이닝과 관련된 개념 및 대표적인 연관규칙 알고리즘들을 살펴본다. 3 장에서는 연

관규칙을 이용한 적응형 학교 웹사이트를 구축하기 위한 방안 및 적용의 예를 제시하며, 4장에서는 결론 및 향후 연구 과제를 제시한다.

2. 이론적 배경

2.1 데이터마이닝과 웹마이닝

1) 데이터 마이닝 (Data Mining)

데이터 마이닝은 지식발견 (Knowledge Discovery)의 과정으로 언급되기도 하는데 일반적으로 많은 양의 데이터에 함축적으로 들어 있는 지식이나 패턴을 찾아내는 기술로 정의된다[3,4].

대형 할인점이나 백화점에서는 데이터 마이닝 기술을 이용하여 고객의 이동경로를 줄일 수 있도록 상품을 진열하고 있으며, 신용카드 회사에서는 우수 고객을 선별하여 안내장을 발송하는 것에, 또는 불량 회원의 카드 사용 패턴을 알아내어 카드 부정사용을 막거나 카드 발급을 제한하는 분야 등에 데이터 마이닝 기술이 폭넓게 활용되고 있다.

2) 웹 마이닝 (Web Mining)

데이터 마이닝이 많은 양의 데이터, 즉 데이터베이스 내에 존재하는 의미 있고 유용한 정보를 캐내는 과정이라면 웹 마이닝은 웹 상에 존재하는 문서나 서비스로부터 의미 있고 유용한 정보를 자동적으로 발견하고 추출하기 위해 데이터 마이닝 기술을 사용하는 과정으로 볼 수 있다[5].

웹 마이닝은 크게 웹컨텐cm 마이닝 (Web Content Mining), 웹구조 마이닝 (Web Structure Mining), 그리고 웹사용 마이닝 (Web Usage Mining)으로 나뉜다. 웹 컨텐츠 마이닝이 웹 컨텐츠, 자료, 문서로부터 유용한 정보를 얻어내는 것이라면, 웹 구조 마이닝은 웹 문서 내의 하이퍼링크 구조에 초점을 두고 웹에 링크된 하위 구조를 발견하기 위함이다. 또한 웹사용 마이닝은 사용자가 웹과 상호작용하는 동안 사용자의 행동을 예측하는 기술에 초점을 두고 있다[5].

웹 사용 마이닝을 위해 최근의 연구는 웹로그를 분석하고 있으며, 그 이유는 사용자와 웹 간의 상호작용의 모든 기록이 웹서버의 로그 파일에 기록되기 때문이다. 본 연구에서 구현하고자 하는 적응형 학교 웹사이트의 경우, 작게는 웹사용 마이닝의 관점에서 크게는 웹마이닝의 관점으로 구분할 수 있다. 이것은 누적된 웹로그 파일의 분석을 통해 사용자들의 행동 패턴을 학습하고, 그 결과를 토대로 웹페이지를 자동으로 개선시켜 주도록 적응형 학교 웹사이트가 구현되기 때문이다.

2.2 연관규칙 탐사를 위한 알고리즘

1) 연관규칙 (Association Rules)

연관규칙은 데이터 마이닝 기술 중에 가장 대표적인 기술로, 장바구니 분석 (Market-Basket-Analysis)으로 불리기도 한다. 발견되었던 연관 규칙 중에 하나는, '일회용 아기 기저귀를 사는 사람은 맥주도 같이 산다.'라는 규칙이었다. 이것은 미국에 대형 편의점의 소비자 장바구니에 담겨진 물품들의 구매 데이터를 마이닝한 결과였다[4].

<표1>과 같은 구매 트랜잭션 데이터베이스가 존재하고, 우리는 한 트랜잭션 내에서 어떤 상품의 존재가 다른 상품의 구매에 영향을 미치는 상품간 연관성을 찾고자 한다고 가정하자.

<표 1 간단한 트랜잭션 데이터베이스 모델>[6]

TID	Items
100	f, a, c, d, g, i, m, p
200	a, b, c, f, l, m, o
300	b, f, h, j, o
400	b, c, k, s, p
500	a, f, c, e, l, p, m, n

$I=\{i_1, i_2, \cdots, i_m\}$을 구매 가능한 아이템들의 집합이라 하고, 트랜잭션의 집합이 DB일 때, 어떤 트랜잭션 T는 구매물품들의 집합으로 $T \subseteq I$ 관계를 갖고 있다. 각각의 트랜잭션은 구분자 (Identifier) 또는 TID에 의해 서로가 구분된다.

X가 아이템들의 집합이고, $X \subseteq T$와 같이 어떤 트랜잭션 T가 X를 포함하고 있다고 하자. 이때 연관규칙은 $X \Rightarrow Y$라는 형식을 취하게 되

는데 여기서 $X \subseteq I$, $Y \subseteq I$ 그리고 $X \cap Y = \emptyset$라는 조건을 만족해야 한다.

연관규칙 $X \Rightarrow Y$는 트랜잭션 집합 D 속에서 X가 포함되어 있을 때 Y역시 포함되어 있는 트랜잭션의 백분율인 신뢰도 (Confidence) C를 유지한다. 또한 연관규칙 $X \Rightarrow Y$는 트랜잭션 집합 D 속에서 X와 Y가 함께 포함된 $(X \cup Y)$ 트랜잭션 백분율인 지지도 (Support) S를 갖게 된다. 즉, 한 트랜잭션 내에서 X의 모든 아이템들이 포함되어 있을 확률 (Probability)을 Pr(X)로 표기한다면, $support(X \Rightarrow Y) = Pr(X \cup Y)$이고, $confidence(X \Rightarrow Y) = Pr(Y|X)$로 나타낼 수 있다[7].

연관 규칙 탐사는 크게 두 단계로 나뉜다[6]. 첫 번째 단계에서는 미리 정해둔 최소 지지도 (Minimum Support) 이상을 만족하는 최대 아이템집합 (The large itemsets)을 구하고, 두 번째 단계에서는 최대 아이템집합을 이용하여 미리 정해둔 최소 신뢰도 c 이상을 만족하는 연관규칙을 발견하는 것이 그것이다.

연관규칙 마이닝의 성능은 주로 첫 번째 단계에서 결정된다. 의미 있는 연관규칙 발견을 위해서는 거대한 데이터베이스 분석이 필수적인데 방대한 양의 데이터베이스 자료 중에서 가장 빈번하게 나타나는 최대 아이템 목록을 얼마나 효율적으로 찾느냐가 이미 찾아진 아이템 목록을 통해 연관규칙을 생성하는 것보다 어렵기 때문이다[4].

2) Apriori 알고리즘

Apriori 알고리즘은 연관 규칙을 찾아주는 가장 대표적인 알고리즘이다[4,6,8]. 연관규칙을 탐사하는 알고리즘들의 목적은 사용자가 미리 정해둔 최소지지도와 최소 신뢰도보다 크거나 같은 모든 연관규칙을 찾는 것이다[9]. 여기서 최소 지지도 이상을 갖는 항목집합을 빈발항목집합 (Frequent Itemset)이라 하고 k개의 항목들로 이루어진 빈발항목 집합을 k-빈발항목집합이라고 한다.

Apriori 알고리즘은 1-빈발항목집합을 생성하기 위해 데이터베이스를 스캔하며, 개개 항목들의 지지도를 계산하여 최소지지도 이상을 만족하는 1-빈발항목집합을 생성한다. 항목의 개수가 2 이상인 k-빈발항목집합 생성시부터는 전단계 빈발항목집합 즉, $(k-1)$-빈발항목집합으로부터 후보항목집합을 만들고, 데이터베이스 스캔을 통해 지지도를 계산하여 k-빈발항목집합을 구하게 된다.

Apriori 알고리즘이 연관규칙을 찾는데 가장 대표적인 알고리즘이기는 하나, 알고리즘 수행에 있어 몇 가지 문제점을 지적할 수 있겠는데, 첫 번째는 $\{i_1, i_2, \cdots, i_{100}\}$과 같이 길이가 100인 빈발항목을 찾기 위해 생성해야할 후보항목의 수는 적어도

$$\sum_{i=1}^{100} \binom{100}{i} = 2^{100} - 1 \approx 10^{30}$$

정도가 된다는 것이며, 두 번째는 여러 번의 데이터베이스 스캔이 필요하다는 점이다[6]. 이를 개선하기 위해 Apriori 알고리즘을 수정한 AprioriTID, Apriori Hybrid알고리즘, DHP 알고리즘 등이 제시되고 있다[8].

3) FP-growth 알고리즘

FP-growth 알고리즘은 빈발항목집합을 구하기 위해 후보를 생성하지 않으며, 한 번의 데이터베이스 스캔으로 FP-tree라는 자료구조를 구축하고 그것을 통해 빈발항목집합을 구하도록 되어 있다[10].

FP-growth의 빈발항목집합 탐사과정을 알아보기 위해 <표1>과 같은 트랜잭션 데이터베이스 TDB가 있다고 하고 최소지지도를 3이라 하자[11]. 데이터베이스가 스캔되고 그 결과 최소지지도 이상을 만족하는 빈발항목 리스트가 구해진다. L=<(f.4), (c.4), (a.3), (b.3), (m.3), (p.3)>이 그것이고 항목을 의미하는 문자와 그것의 지지도가 '.'으로 연결되어 표현되었다.

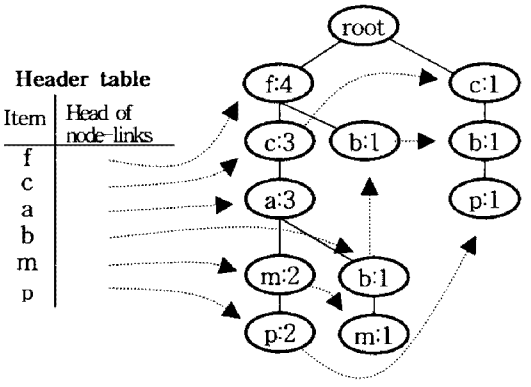
한 트랜잭션 내에서 빈발항목들이 그들의 지지도를 기준으로 내림차순으로 정렬되며, 동일 지지도일 경우 알파벳 오름차순으로 정렬된다. <표2>는 트랜잭션 별로 빈발항목을 정렬하여 다시 작성한 것이다[11].

<표 2 트랜잭션 데이터베이스>

TID	Items	(Ordered) Frequent Ites
100	f, a, c, d, g, i, m, p	f, c, a, m, p
200	a, b, c, f, l, m, o	f, c, a, b, m
300	b, f, h, j, o	f, b
400	b, c, k, s, p	c, b, p
500	a, f, c, e, l, p, m, n	f, c, a, m, p

이후 FP-tree가 만들어지는데, 먼저 null 값을 갖는 FP-tree의 루트 (Root)가 만들어지고, 빈발항목리스트의 항목들만이 선택되어 노드(Node)가 되며, L의 순서에 따라 노드가 정렬되고 간선이 연결된다. TDB 내에 있는 각각의 트랜잭션들이 고려될 때 FP-tree는 새로운 노드와 간선에 의해 확장되어 간다.

트리의 순회를 손쉽게 하기 위해 헤더 테이블 (Header Table)이 만들어지는데, 이 헤더 테이블은 <그림1>에서와 같이 빈발항목과 그것이 트리 내에서 처음 등장하는 노드를 가리키기 위한 링크 정보로 구성되어 있다. <그림 1>은 완성된 FP-tree와 그것의 헤더테이블 모습을 보여주고 있다.



<그림 1 완성된 FP-tree와 헤더 테이블>

위와 같이 FP-tree가 구성되면, FP-tree를 입력인자로 받아 FP-growth 알고리즘이 빈발패턴을 마이닝하게 된다. FP-growth 알고리즘은 완전한 빈발항목집합을 만들기 위해, 빈발항목리스트 (f, c, a, m, p)에 대해, 노드 링크 정보를 기반으로 p가 포함되는 트랜잭션들을 선택해서 p를 포함하고 있는 모든 경로(Path)를 구성하는데, 이를 p의 Conditional Pattern Base라고 한다. 이후 각각의 경로를 참조하여 p에 대한 FP-tree, 즉 p의 Conditional FP-tree가 만들어지고 이것을 참조하여 최소지지도를 만족하는 빈발항목의 조합으로 패턴을 생성한다. 위와 같은 과정을 나머지 빈발항목에 대해 반복한다. 앞의 예에서 최종 빈발항목집합은 {{c,p}, {f,c,a,m}}이다.

2.3 적응형 웹사이트

적응형 웹사이트를 구축하는 기술로 크게 개인화 (Personalization)와 고객화 (Customization)를 들 수 있다[12]. 개인화된 시스템에서는 웹사이트의 내용 또는 심지어 웹사이트의 구조에 관련된 수정이 동적으로 수행된다[13]. 최근에는 국내 유명 포털사이트뿐만 아니라 경매·소핑 사이트 등 다양한 분야에서 개인화된 웹서비스를 제공하고 있다.

개인화가 특정 사용자를 위한 맞춤화 과정이라면, 고객화된 사이트는 전체 방문자들의 특성을 고려하여 그것의 구조나 표현을 적응시켜가는 과정이라고 말할 수 있다[13]. 고객화는 과거에 사이트를 방문했던 사람들의 정보를 학습하여, 이후 방문하게 될 사용자들이 웹 정보를 좀 더 쉽게 사용할 수 있게끔 웹사이트에 변형을 가한다.

적응형 웹사이트 구축을 위한 최근의 연구 논문을 살펴보면, [2]에서는 적응형 웹사이트 구축을 위해 클러스터링 마이닝을 위한 알고리즘을 제안하였고, [14]에서는 클러스터링 기술을 이용해 사용자들을 그룹핑하고, 그 그룹에 속한 사람들이 자주 방문하는 사이트에 대한 연관규칙을 찾아 개인화된 웹 페이지 추천 시스템 개발 방안을 제시하였으며, [12]에서는 마코프체인의 성질을 이용하여 추천 링크를 구성하였다. [8]에서는 Apriori 알고리즘을 응용한 TPA와 LFD 알고리즘을 제안하며 하이퍼링크 구조상 의미 있으나 거리가 먼 페이지를 하이퍼링크로 연결하여 색인페이지를 생성하는 적응형 웹사이트 구현 방안을 제시하였다.

3. 적응형 학교 웹페이지 알고리즘

적응형 학교 웹페이지 시스템은 크게 전처리과정 (Preprocessing)과 패턴탐색 (Pattern Mining)그리고 하이퍼링크 자동 생성 과정으로 나뉜다.

3.1 전처리과정 (Preprocessing)

웹서버의 로그 파일은 방문객이 접근했던 문서들에 대한 정보를 담고 있다. 웹서버가 취하는 로그 파일 형식에 따라 그 외형이 조금 다를 수 있으나 공통적으로 그 안에는 방문자의 IP, 접근 날짜와 시간, 요청방법, 접근 문서와 URL, 전송 바이트 수 등 웹사용 마이닝을 위한 충분한 자료가 들어있다.

<표3>은 아파치 웹서버 엑세스 로그 파일의 일부를 나타낸 것이다.

<표 3 아파치 웹서버 엑세스 로그 파일>

```
127.0.0.1 - - [22/Jun/2004:18:48:24 +0900] "GET
 /project/article/article/main_new.php HTTP/1.1" 304 -
127.0.0.1 - - [22/Jun/2004:18:55:20 +0900] "GET
 /project/article/article.htm HTTP/1.1" 304 -
127.0.0.1 - - [22/Jun/2004:18:56:14 +0900] "GET
 /home/class.php HTTP/1.1" 304 -
127.0.0.1 - - [22/Jun/2004:18:56:14 +0900] "GET
 /project/article/article_icon/home.gif HTTP/1.1" 304 -
```

그러나 이런 정보들이 웹사용 마이닝을 위해 전부 필요한 것은 아니기 때문에, 마이닝 작업에 적절한 형태로 전처리하는 과정이 필요하게 된다. 적응형 학교 웹페이지 구축을 위해 최소한 방문객의 IP와 접속한 날짜와 시간, 접근 파일명만 있으면 되므로 전처리된 로그 파일은 <표4>와 같은 정보를 갖게 된다. 하나의 파일이라 할지라도 CSS나 js 및 그림 파일들은 마이닝 대상이 아니므로 전처리 과정에서 제외시켜 준다.

<표 4 전처리된 엑세스 로그 파일>

```
127.0.0.1 [22/Jun/2004:18:48:24] main_new.php
127.0.0.1 [22/Jun/2004:18:55:20] article.htm
127.0.0.1 [22/Jun/2004:18:56:14] class.php
```

3.2 패턴탐색 (Pattern Mining)

1) 트랜잭션 구성

연관규칙 탐사를 위해 사용되는 구매 데이터베이스는 고객별 Id와 그 고객이 구매한 장바구니 속 물품들을 하나의 트랜잭션으로 구성한다. 즉, 각각의 고객을 구별하기도 쉽고 트랜잭션을 구성하기도 용이하다.

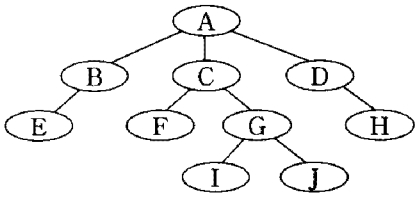
그러나 우리가 마이닝 할 웹 영역에서는 방문자들이 사용했던 웹문서들을 가지고 트랜잭션을 구성해야 하는데, 로그 파일은 시간의 순서에 따라 차곡차곡 쌓여진 데이터들의 묶음일 뿐이므로 다음과 같은 가정을 바탕으로 방문객이 구별되고, 트랜잭션이 구성된다.

· **방문객의 구별** : IP 주소와 방문객은 일대일 대응된다. 따라서 서로 다른 IP는 서로 다른 방문객으로 취급될 것이다.

· **세션 (Session)의 구분** : 미리 정해둔 '사용자 정의 최대 시간 간격 (User Specified Maximum time gap)' 내에 접근한 로그 파일 안의 연속 히트 (Series of Hit)를 한 세션으로 본다.

따라서 패턴 마이닝을 위해 사용될 새로운 데이터베이스는 방문객 구별을 위해 IP 주소를 사용할 것이며, 트랜잭션은 세션 내에 방문객이 거쳐간 웹문서들의 순서열로 구성하되, 백트래킹에 의한 문서접근은 고려하지 않으며 새로고침 등으로 문서를 여러 번 요청한 경우도 한 번의 접근으로 간주한다.

<그림2>와 같은 문서 구조를 갖는 학교 홈페이지가 있다고 가정하자. 각 노드들은 홈페이지 상의 문서이며 간선은 하이퍼링크 되었음을 의미한다.



<그림 2 학교 홈페이지 문서 구조>

<표5>는 학교 홈페이지에 접근한 방문객들의 웹문서 운행 경로를 트랜잭션으로 구성한 데이터베이스이다.

<표 5 트랜잭션 데이터베이스>

TID	Transation sequence
100	A B E H
200	A C F J I
300	H A B E
400	A C G J F
500	J G C F
600	A C G B E D H

위 트랜잭션 데이터베이스에서 100이라는 ID를 부여받은 방문객은 A 문서를 본 후 B 문서를 보았으며 같은 방식으로 E와 H 문서를 순서대로 보았음을 알 수 있다.

2) FP-tree 구성

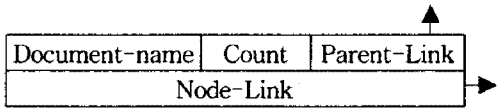
최대 빈발 웹문서 탐색 과정은 크게 두 단계로 나뉘는데, 첫 번째 단계에서는 FP-tree를 만들고, 두 번째 단계에서는 이미 만들어진 FP-tree를 바탕으로 빈발 웹문서의 패턴을 마이닝 한다.

FP-tree는 헤더 테이블, 루트 노드, 그리고 아이템-프리픽스-서브트리 (Item-Prefix-Subtree)로 구성된다.

헤더 테이블은 서두에서 언급했듯이 FP-tree의 순회를 돕기 위해 만들며, 테이블 내에는 두 개의 필드가 존재한다. 하나는 웹문서의 이름을 저장하기 위한 필드 (Document-name field)인데, 데이터베이스의 첫 번째 스캔 결과 빈발 문서로 선별된 문서 이름이 저장된다. 나머지 하나는 링크 정보를 저장하기 위한 필드 (Head of Node-Link)로서, FP-tree 생성 과정에서 어떤 문서 노드가 처음 생성되었다면 헤더 테이블의 이 필드에는 방금 생성된 노드를 가리키기 위한 포인터 정보가 저장될 것이다.

루트 노드를 제외하고, FP-tree에 속한 노드들은 공통적으로 <그림3>과 같이 4개의 필드를 갖게 된다. Document-name 필드는 웹문서 이름을 저장하기 위한 필드이며, Count 필드는 해당 웹문서가 포함된 트랜잭션의 수를 저장하게 될 것이다.

Parent-Link는 부모 노드를, 그리고 Node-Link필드는 FP-tree 내에서 같은 이름을 갖는 노드들을 가리키는데 만약 해당 사항이 없다면 'null' 값을 갖도록 한다.



<그림 3 FP-tree 노드 필드 구성>

FP-tree 알고리즘은 [15]에 제시되어 있으며 <표6>은 웹문서를 대상으로 하는 수정된 FP-tree 구성 알고리즘이다.

알고리즘은 Step1에서, 데이터베이스 스캔

<표 6 웹 문서를 대상으로 하는 FP-tree 구성 알고리즘>

Input : A transaction database DB and a minimum support threshold ε.
Output : FP-tree(The frequent-web document pattern tree of DB.)
Method : The FP-tree is constructed as follows.
Setp1. Scan the transaction database *DB* once.
Collect *F*, the set of frequent document, and the support of each frequent document.
Sort *F* in support-descending order as *FList*, the list of frequent document.
Setp2. Create the root of an FP-tree, *T*
Setp3. For each transaction *Trans* in DB do{
Select the frequent document in *Trans* and sort them according to the order of *FList*.
Let the sorted frequent-document list in *Trans* be [*p* | P].
(p is the first element and P is the remaining list.)
Call ***make_tree***([*p*|*P*], *T*). }
Setp4. Function ***make_tree***([*p*|*P*], *T*) {
for each p do {
If T has a child N such that N.Document-name=p.Document-name then increment N's Count by 1.
else { Create a new node N.
N.Document-name = *p*.Document-name.
N.Count = 1.
N.Parent-link = *T*.
N.Node-link=null.
The previous nodes with the same Document-name links to N.}
}

을 통해 최소 지지도를 만족하는 빈발 문서들의 리스트를 작성하고 각 빈발 문서들의 지지도를 계산한다. 빈발 문서들은 그들의 지지도에 따라 내림차순으로 정렬된다.

이후 Step2에서, 'null' 값을 갖는 루트 노드가 생성됨을 볼 수 있다. 마지막 Setp3에서는 DB 내에 있는 모든 트랜잭션에 대해 다음과 같은 반복 작업을 하고 있다. 각각의 *Trans*에 들어 있는 웹문서 중에서 빈발 문서를 선별하여 지지도 순으로 정렬하고, 정렬된 빈발 문서에 대해 make_tree() 함수를 호출함으로써 FP-tree를 구성한다.

<표5>에서 최소 지지도를 3이라고 했을 때 빈발 문서와 그것의 지지도를 구하면 {(A.5), (C.4), (B.3), (E.3), (F.3), (G.3), (H.3), (J.3)}이 된다. 이후 알고리즘은 각각의 트랜잭션을 스캔하면서 트랜잭션 내에 (A, C, B, E, F, G, H, J)에 해당되는 문서가 있는지 확인하고 순서대로 정렬한다.

아래 <표6>은 FP-tree 구축 과정의 이해를 돕기 위해 빈발 문서를 중심으로 <표5>를 다시 정리한 것이다.

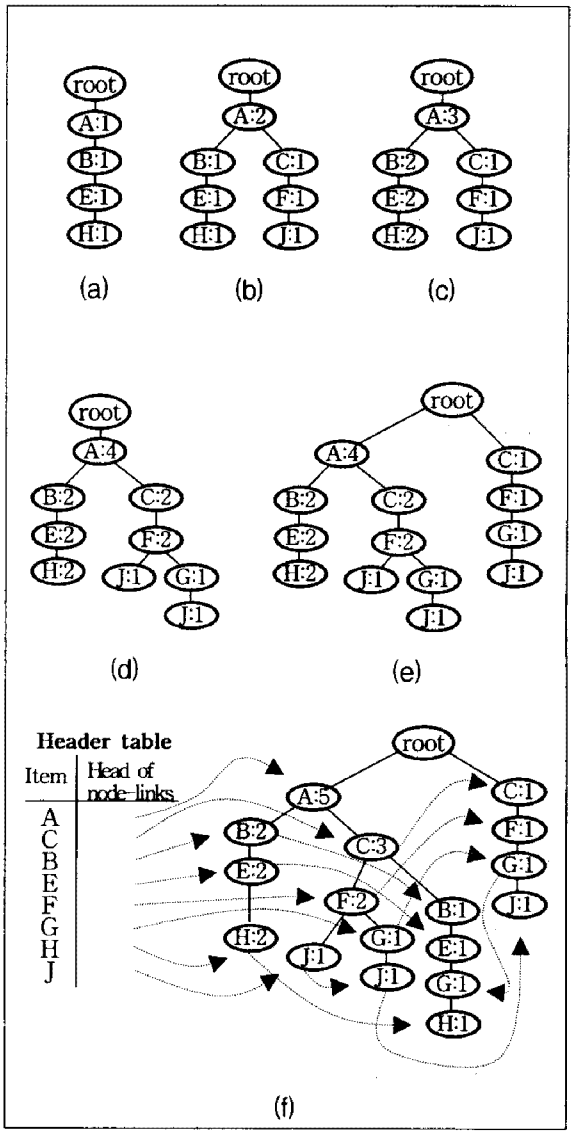
<표 6 트랜잭션 시퀀스와 정렬된 빈발문서>

TID	Transation sequence	Ordered Frequent Document
100	A B E H	A B E H
200	A C F J I	A C F J
300	H A B E	A B E H
400	A C G J F	A C F G J
500	J G C F	C F G J
600	A C G B E D H	A C B E G H

<그림4>는 <표6>을 바탕으로 FP-tree를 구축하는 과정을 도해한 것이다. 편의상 헤더 테이블의 링크 구축 과정은 마지막 단계에만 나타내었다.

3) FP-tree를 이용한 패턴 탐색

적응형 웹 페이지 시스템에 적합한 패턴 탐색을 위하여 FP-growth 알고리즘이 수정되어 적용된다. 다음은 수정된 FP-growth 알고리즘에 대한 간략한 서술이다.



<그림 4 FP-tree 구축 과정>

단계1. 알고리즘은 전 단계에서 만들어진 FP-tree와 최소지지도를 입력받는다.

단계2. Header Table의 Document-name field에 있는 문서 α 중에서 가장 마지막에 정렬된 문서 *Doc*가 선택된다.

단계3. FP-tree 링크 구조를 이용하여 *Doc*가 포함된 모든 경로 (Path)를 구한다. 이 경로 집합을 *Doc's Prefix path* 또는 *Doc's Conditional Pattern Base* 라고 부르며, *Doc's Conditional Pattern Base* 내에 있는 문서들을 β로 설정한다.

단계4. *Doc's Conditional Pattern Base*를 기반으로 하여 FP-tree를 만드는데, 이를 *Doc's conditional FP-tree* 라고 한다.

단계5. *Doc's conditional FP-tree* 순회를 통해 최소지지도를 만족하는 빈발 웹문서 *FDocs* 들을 추출한다.

단계6. *FDocs*∪*Doc* 연산을 통해 *Doc*를 포함한 최대 빈발 웹문서를 생성한다.

단계7. Header Table의 나머지 웹문서에 대해서도 같은 작업을 반복한다.

<표7>은 수정된 FP-growth 알고리즘을 통해 생성된 Conditional Pattern Base와 Conditional FP-tree를 표로 나타낸 것이며, <표8>은 수정된 FP-growth 알고리즘이다.

<표 7 Conditional Pattern Base와 Conditional Pattern FP-tree>

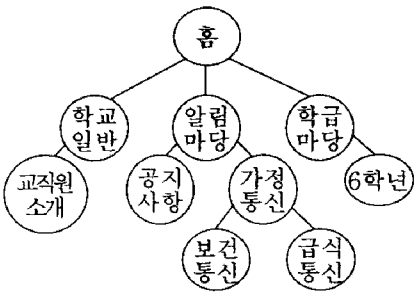
Doc-ument	Conditional Pattern Base	Conditional FP-tree
J	{(A,C,F:1),(A,C,F,G:1),(CFG:1)}	{(C,F:3)}|J
H	{(A,B,E:2),(A,C,B,E,G:1)}	{(A,B,E:3)}|H
G	{(A,C,F,G:1),(CF:1),(A,C,B,E:1)}	{(C:3)}|G
F	{(A,C:2),(C:1)}	{(C:3)}|F
E	{(AB:2),(A,C,B:1)}	{(A,B:3)}|E
B	{(A:2),(AC:1)}	{(A:3)}|B
C	{(A:3)}	{(A:3)}|C
A	∅	∅

탐색 결과를 살펴보면, J 페이지를 방문한 사람은 C와 F 페이지를 같이 방문함을 알 수 있으며, 같은 방법으로 H 페이지를 방문하는 사람들은 동시에 A, B, E 페이지를 같이 방문하는 경향이 있음을 알 수 있다.

3.3 웹 마이닝 알고리즘의 적용

적응형 웹사이트에서는 사용자들의 웹 문서 운행의 편리를 위해 기존 웹사이트를 자동적으로 갱신해 나간다. 하이퍼링크 생성과 웹사이트 갱신을 위한 다양한 방법이 제안되고 있다. 예를 들면, 색인페이지를 생성해서 해당 웹페이지에 삽입하는 방법, 새롭게 하이퍼링크된 글자들을 진하게 또는 크게 하는 방법 등이다. 본 연구에서는 각 웹페이지 특정 영역을 하이퍼링크 삽입부분으로 미리 마련해 놓고 연관규칙 결과 연관성 높은 페이지가 발견될 경우 해당 페이지를 그 특정 영역에 하이퍼링크하여 삽입하는 방법을 사용한다.

만약 어떤 학교 홈페이지의 구조가 아래 <그림5>와 같다고 하자.



<그림 5 학교 홈페이지 구조>

패턴 탐색 결과 6학년 학급홈페이지에 접근한 사람들은 다음으로 학교 일반 페이지와 교직원 소개를 본다고 했을 때 6학년 홈페이

<표 8 수정된 FP-growth 알고리즘>

```
Input : FP-tree, ε(minimum support threshold)
Output : Frequent patterns
Procedure FP-growth(Tree, α){
        if Tree contains a single path P
           then {
                   generate pattern β∪α with support=minimum support of nodes in β
                   (β are nodes in the path P)
                   return pattern β∪α
                 }
        else for each Doc in α do{
                   generate all path set PATH via the node-link structure
                   construct Doc's conditional pattern base using the PATH
                   construct Doc's conditional FP-tree Tree_Doc
                     (C is the set of nodes with minimum support in the Tree_Doc)
                   generate pattern c∪Doc (c are nodes in the C)
                   return pattern c∪Doc
        }
}
```

지에 '학교일반'과 '교직원소개' 텍스트를 하이퍼링크하여 삽입함으로써 적응형 학교 웹페이지를 구축하게 된다.

4. 결론 및 향후 연구과제

본 논문에서는 연관 규칙을 사용하여 학교 적응형 웹페이지를 구축하는 방안을 제시하였다. 학교 홈페이지 방문객들의 접근 패턴을 알기 위해 웹서버 로그 파일을 사용하였으며, 패턴을 탐색하기 위해 FP-tree를 구성하고 수정된 FP-growth 알고리즘을 통해 빈발 웹문서 패턴을 얻었다. 얻어진 빈발 문서들은 해당 페이지에 방문객들이 접근했을 때 추천문서로 제공되며, 하이퍼링크로 연결되어 제공되기 때문에 해당 추천 페이지로 쉽게 이동이 가능하게 된다.

학교 적응형 웹사이트 구축은 학교 홈페이지 담당교사들의 업무 부담을 줄여줄 것으로 기대하며, 학생 및 일반 방문객들에게는 편리한 학교 홈페이지 탐색 경험을 제공할 것이다.

향후 연구 과제로는 첫째, 제안한 알고리즘의 효율성과 복잡도 분석이 필요하며, 둘째 제안한 적응형 학교 웹사이트 알고리즘을 운영중인 학교 웹페이지에 실제 적용시키는 것이다.

5. 참고문헌

[1] M.Perkowitz, O.Etzioni, "Adaptive Web Site: an AI challenge", IJCAI:Proceedings of the conference, V.15, No. 1, pp.16-23, 1997.

[2] M.Perkowitz, O.Etzioni, "Adaptive Web Site: Automatically Synthesizing Web Page", IJCAI:Proceedings of the conference, V.15, No. 1, pp.727-732, 1998.

[3] M.S.Chen, J.Han and P.S.Yu, "Data Mining : An Overview from a Database Perspective", IEEE Transactions on Knowledge and Data Engineering, Vol.8, No. 6, pp.866-883, 1996.

[4] 송명근, "연관규칙탐사를 위한 효율적인 자료구조에 관한 연구", 홍익대학교 대학원, 석사학위논문, 2001.

[5] R.Kosala, H.Blockeel, "Web Mining Research: A Survey", ACM SIGKDD, V.2, Issue 1, pp.3-9, 2000.

[6] M.Kantardzic, "Data Mining", WILEY-NTERSCIENCE, 2003.

[7] R.Scrikant, R.Agrawal, "Mining Generalized Association Rules", Future generations computer systems : FGCS, v.13 no.2/3, pp.161-180, 1997

[8] 고경자, "웹마이닝을 이용한 적응형 웹사이트 구축에 관한 연구", 경기대학교 대학원, 석사학위논문, 2001.

[9] 황종원,강맹규, "후보 2-항목집합의 개수를 최소화한 연관규칙 탐사 알고리즘", 工業經營學會誌, 第21卷, 第48輯, pp.53-63, 1998.

[10] J.Han, J.Pei, "Mining Frequent Patterns by Pattern-Growth : Methodology and Implications", ACM SIGKDD, V.2, Issue. 2, pp.14-20, 2000.

[11] J.Han, J.Pei,Y.Yin, "Mining Frequent Patterns without candidate generation", SIGMOD record, V.29, No. 2, pp.1-12, 2000.

[12] 이상민, "웹마이닝을 통한 적응적 웹사이트 구축에 관한 연구", 한양대학교 대학원 석사학위논문, 2000.

[13] M.Eirinaki, M.Vazirgiannis, "Web Mining for Web Personalization", ACM Transactions on Internet Technology, Vol.3, No.1, pp.1-27, 2003.

[14] 이은경, "클러스터링과 연관규칙을 이용한 개인화된 웹페이지 추천시스템", 인하대학교 대학원 석사학위논문, 2002.

[15] J.Han, J.Pei, Y.Yin, R.Mao, "Mining Frequent Patterns Without Candidate Generation : A Frequent-Pattern Tree Approach", Data Mining and Knowledge Discovery, Vol.8, pp.53-87, 2004.