論文
4-10-8

웹 로그 분석을 이용한 추천 에이전트의 개발

Development of Recommendation Agents through Web Log Analysis

김성학(Kim Sung-Hak)* 이창훈(Lee Chang-Hun)**

요 약

웹 로그는 사용자가 웹 사이트의 데이터를 액세스할 때 웹 서버에 의해 기록되는 정보로써 최근 인터넷 이용의 급속한 증가로 인해 웹 로그의 활용가치가 더욱 중요하게 되었으며, 웹 로그의 분석 결과는 웹 사용자들의 행위를 나타내는 패턴을 분석하거나 웹 사이트의 구조를 재배치하는데 이용될 수 있다. 이를 실현하기 위한 많은 연구들은 주로 연관규칙과 순차패턴을 이용하고 있는데, 대다수는 Apriori 알고리즘을 기본으로 하고 있어서 대용량의 데이터베이스에 적용하기에는 컴퓨팅 시간적 측면에서 비효율적이다. 따라서 본 논문에서는 웹 환경에서 흥미있는 패턴을 탐사하는 새로운 알고리즘을 개발하여 보다 빠르게 패턴탐사를 수행하고, 많은 사용자들이 관심있게 순차적으로 접근하고 있는 정보를 시스템 관리자에게 제공할 수 있는 추천에이전트를 개발한다.

ABSTRACT

Web logs are the information recorded by a web server when users access the web sites, and due to a speedy rising of internet usage, the worth of their practical use has become increasingly important. Analyzing such logs can use to determine the patterns representing users' navigational behavior in a Web site and restructure a Web site to create a more effective organizational presence. For these applications, the generally used key methods in many studies are association rules and sequential patterns based by Apriori algorithms, which are widely used to extract correlation among patterns. But Apriori inhere inefficiency in computing cost when applied to large databases. In this paper, we develop a new algorithm for mining interesting patterns which is faster than Apriori algorithm and recommendation agents which could provide a system manager with valuable information that are accessed sequentially by many users.

* 종신회원 : 유한대학 컴퓨터정보과 교수
**정회원 : 건국대학교 컴퓨터정보통신과 교수

논문접수 : 2003. 9. 17.
심사완료 : 2003. 10. 6.

1. 서론

웹(Web)의 급속한 발전으로 인하여, 인터넷을 통한 정보교류와 정보검색이 보편화되면서 정보의 질적양적인 면에서의 급격한 증가와 사용자의 폭발적인 증가를 가져오게 되었다. 이러한 현상은 데이터마이닝(data mining) 기법[1]을 자연스럽게 웹 환경으로 적용시키는 동기가 되었다.

웹 환경에 가능한 정보의 근원(source)이 매우 빠르게 성장함에 따라 원하는 정보의 자원을 자동으로 찾아주는 툴과 그들의 사용 패턴을 분석해야 하는 필요성이 증가하게 되었고, 이러 한 요소들은 정보를 효과적으로 탐사하기 위하 여 서버/클라이언트 시스템 모두에게 필요하다 는 인식이 높아지게 되었다[4,5]. 또한 많은 웹 사이트들에서는 그 사이트를 이용하는 사용자 들이 가장 편리하게 브라우징(browsing)할 수 있는 방법을 모색하여 각 페이지와 그 페이지들 사이에서의 하이퍼미디어(hypermedia) 링크상 에 정보가 구축되도록 설계하려고 한다. 이 때 페이지의 양이 적은 사이트들에서는, 간단한 사용 통계를 적용하는 웹 디자이너의 직관에 따라서 사용자의 브라우징 행위를 예견하는 것이 적합할 수도 있다. 그러나 웹 사이트의 크기와 복잡도가 증가하게 되면, 기존의 웹 로그(log) 분석 툴[9,10,11]에서 제공하는 간단 한 통계적인 방법으로는 웹 사이트가 어떻게 사용되고 있는 가에 대한 의미있는 분석을 제공 하는 것이 어렵게 된다.

일반적으로 웹 서버에 저장되는 로그 파일은 사용자가 웹 페이지를 액세스할 때마다 기록되 는데, 웹 서버는 모든 액세스에 대해 웹 로그 엔트리(entry)를 저장한다. 이것은 사용자 액세 스에 대한 중요한 정보들을 기록하고 있으며 IP 주소, 요구된 URL, 시간의 기록(timestamp) 등이 포함된다([그림 1]).

```
203.229.252.206 - - [08/Jan/2003:12:47:21 +0900]
"GET /bbs/menu.html HTTP/1.0" 200 1537
203.229.252.206 - - [08/Jan/2003:12:47:25 +0900]
"GET /images/bbs1.jpg HTTP/1.0" 200 304
```

[그림 1] 웹 로그 파일의 예
[Fig. 1] Example of a Web log file

최근 웹 사용자들이 급증함에 따라 매우 방대한 양의 로그들이 저장되고 있는데, 이러한 대량의 웹 로그들을 분석하여 유용한 정보를 추출하는 것은 쉬운 작업이 아니다. 이를 위해서는 효율적인 알고리즘이 필요한데, 기존의 알고리즘들은 패턴들의 연관성을 찾아내기 위해 일반적으로 Apriori 알고리즘[2]을 이용하고 있 으나, 이 방법은 컴퓨팅 시간에 소요되는 비용 이 크다[7].

따라서 본 논문에서는 웹 로그를 분석하여 트랜잭션 데이터베이스(transaction databases)에서 패턴들 사이에서의 연관성을, 기존의 Apriori를 기반으로 하는 알고리즘들보다 빠르게 탐사할 수 있는 알고리즘을 개발한다. 이는 패턴들간의 연계성을 추출하기 위해서 연관규칙 (association rules)과 순차패턴(sequential patterns)의 기본적인 방법을 이용하는 새로운 구조의 알고리즘으로써, 서로 관련이 깊은 다양 한 패턴들간의 연관도를 보다 빠르고 구체적 으로 나타낼 수 있도록 한다. 또한, 결과로써 추출된 패턴들을 시스템 관리자에게 제공하는 추천 에이전트를 개발하여, 많은 웹 사용자들이 빈발하게 순차적으로 접근하고 있는 정보가 무엇인 지를 시스템 관리자가 효율적으로 파악 할 수 있게 하여 웹 사이트의 구조나 내용 등의 재배치를 위해서 사용될 수 있도록 한다.

본 논문의 구성은 다음과 같다. 제2장에서는 관련 연구로서 연관성을 추출하는 방법에 대해 살펴보고, 제3장에서는 웹 로그 파일을 전처리하여 트랜잭션 데이터베이스로 구성하는 과정을 설명한다. 제4장에서는 시스템 구조 및 패턴탐사 과정에 대해서 논의하고 제5장에서는

실험 및 평가, 제6장에서는 결론을 제시한다.

2. 관련연구

이 장에서는 데이터베이스의 트랜잭션으로부터 관련있는 패턴을 추출하기 위하여 필요한 데이터마이닝 기법 중에서 본 논문과 관련이 깊은 연관규칙과 순차패턴에 대해 알아본다 [2,8].

2.1 연관규칙

데이터베이스에서 잘 알려져 있지 않은 숨겨진 패턴을 탐사하는 연구 중에서 연관규칙에 대해 가장 많은 연구가 이루어 졌다. 연관규칙은 문자 그대로 한 항목 그룹과 다른 항목 그룹 사이에 존재하는 강한 연관성을 찾아내어 그룹화 하는 클러스터링(clustering)의 일종이다. 또한, 동시에 구매될 가능성이 큰 상품들을 찾아냄으로써 장바구니 분석(market basket analysis)에서 다루는 문제들에 적용할 수 있다. 연관규칙 기법에 적용되는 데이터는 판매 시점에서 기록된 거래와 품목에 관한 정보를 담고 있고, 연관규칙 탐사과정은 크게 두 단계로 진행이 된다. 첫번째는 높은 지지도(support)를 갖는 즉, 항목간의 연관성이 높다고 가정되는 항목의 집합(itemset)인 빈발 항목집합(frequent or large itemsets)을 식별하는 작업이고, 두 번째 단계는 이러한 빈발 항목집합을 이용하여 높은 신뢰도(confidence)를 갖는 연관규칙을 도출하는 작업이다. 여기서 지지도와 신뢰도는 매우 중요한 개념으로써 빈발 항목집합과 연관규칙을 찾아내는데 있어서 논리적 타당성을 제공하는 큰 역할을 한다.

[연관규칙의 정의]
$I = \{i1, i2, , im\}$을 항목들의 집합이라 하자. D를 트랜잭션들의 집합이라 부르고, 각 트랜잭션 T는 $T \ I$ 인 항목들의 집합이다. 각 트랜잭션들은 TID(Transaction IDentifier)를 갖고 있다. X를 항목들의 집합이라고 하면, $X \ T$ 이고 이 때, 트랜잭션 T가 X를 포함한다고 한다. 연관규칙은 R : X Y 형식이고, 이 때 X와 Y는 서로 같은 원소를 갖지 않는 항목집합이다. 즉, $X, Y \ I$ 이고 $X \ Y =$ 이다. 단, Y 이어야 한다.

만일 한 트랜잭션이 X를 지지한다면, 또한 어떤 확률에 의해 Y도 지지할 것이라는 예측이 연관규칙이다. 이런 확률을 이 규칙의 신뢰도(conf(R))라 한다.

$$\text{conf(R)} = p(Y \subseteq T \ X \subseteq T) = \frac{p(Y \subseteq T \wedge X \subseteq T)}{p(X \subseteq T)} = \frac{sp(X \cup Y)}{sp(X)}$$

또한 T가 X의 모든 항목들을 포함한다면(X T) T가 집합 X를 지지(support)한다고 한다. X의 지지도를 $sp(X)$로 정의하며, 이는 X를 지지하는 D에 있는 모든 트랜잭션의 개수를 의미한다. 따라서, D에 있는 규칙 R에 대한 지지도는 $sp(XY)$가 된다. 규칙의 신뢰도는 얼마나 자주 적용할 수 있는 지를 나타내는 반면 지지도는 그 규칙 전부가 얼마나 믿을 만한 지를 보여준다. 규칙이 데이터베이스에서 적절해지려면 충분한 지지도와 신뢰도를 가져야 한다.

[연관규칙 탐사단계]
(1) 빈발항목집합을 찾는다.
미리 결정된 최소지지도 $smin$ 이상의 트랜잭션 지지도를 갖는 모든 빈발 항목집합들을 찾는다.

(2) 데이터베이스로부터 연관규칙 생성을 위하여 빈발 항목집합을 사용한다.
모든 빈발 항목집합 l에 대하여 l의 모든 공집합이 아닌 부분집합들을 찾는다. 각각의

부분집합 a에 대하여, $sp(a)$에 대한 $sp(l)$의 비율이 적어도 최소신뢰도 $cmin$이상이면, 즉

$\frac{sp(l)}{sp(a)} \geq c_{min}$, a $(l - a)$ 형태의 규칙을 생성한다.

그러므로 어떤 주어진 최소신뢰도 cmin와 최소지지도 smin에 대하여 만일 conf(R) cmin 이고 sp(R) smin이면 규칙 R은 D에 대하여 성립한다. 규칙이 성립되기 위하여 필요한 조건으로서 규칙의 조건부(antecedent)와 결과부(decen-dent)가 모두 빈발해야 한다.

Apriori 알고리즘은 빈발패턴 발견을 위해서 먼저 후보패턴을 생성하고 이것이 빈발한 지를 검사하기 위해 매번 데이터베이스를 스캔하는 방식으로, 예를 들면 길이가 50인 빈발패턴을 구하기 위해서 Apriori는 최대 250?105 개의 후보패턴들을 생성해야 하고, 이것이 빈발한 지를 검사하기 위하여 계속해서 데이터베이스를 참조해야만 하는데, 이는 매우 비용이 큰 작업이다. 따라서 후보패턴이 적거나 없다면 알고리즘의 수행시간은 매우 효율적일 것이다.

2.2 순차패턴

순차패턴탐사는 한 트랜잭션 안에서 발생하는 항목들간의 연관규칙에 시간의 변이를 추가한 것이다. 즉, 연관규칙은 트랜잭션 안에서 어떤 항목을 함께 구입하는 가에 관한 문제로 트랜잭션 내의 문제인 반면, 순차패턴을 발견하는 것은 트랜잭션 상호간의 문제인 것이다[3]. 각 트랜잭션은 고객 ID와 트랜잭션 시간과 그 시간에 구매된 항목들로 구성되고, 같은 고객들에 대해서는 같은 시간에 두개 이상의 트랜잭션은 존재하지 않으며 또한 항목의 수량을 고려하지 않는다고 가정한다. 이렇게 구성된 각 고객에 대한 각각의 시퀀스 집합(트랜잭션 데이터베이스)에서 다른 시퀀스에 포함되지 않는 시퀀스 -최대 시퀀스(maximal sequences)를 순차패턴이라 부르며 최소지지도를 만족하는 시퀀스를 빈발 시퀀스(large sequences)라 한다. 시퀀스에 대한 지지도의 정의는 시퀀스를 지지하는 전체 고객들의 수이다. 빈발 시퀀스는 항목집합 목록의 형태로 나타나며, 그 항목집합들은 반드시 최소지지도를 만족해야 한다. 주어진 고객에 대한 트랜잭션 데이터베이스에서 순차패턴 탐사는 사용자가 정의한 최소지지도를 만족하는 모든 빈발 시퀀스들 사이에서의 최대시퀀스를 찾는 것이며, 이것이 연관규칙이 된다.

즉, 순차패턴은 동시에 구매될 가능성이 큰 상품 군을 찾아내는 연관규칙에, 시간의 개념이 포함되어 순차적인 구매 가능성이 큰 상품 군을 찾아내는 방법이다. 순차패턴에서는 연관규칙 A B는 "상품 A가구매되면 일정 시간이 경과한 다음 상품 B가 구매된다."라고 해석된다. 즉, 순차패턴은 구매 순서가 고려되어 상품간의 연관성이 측정되고, 이에 따라 유용한 연관규칙을 찾는 기법이다.

대표적인 알고리즘으로는 AprioriAll과 GSP(Generalized Sequential Pattern)[8]가 있는데, 이 또한Apriori를 기본으로 하고 있기 때문에, 후보패턴을 만들고 그 후보패턴이 트랜잭션 데이터베이스에 얼마나 많이 존재하는가, 즉 최소지지도를 초과하는 패턴을 찾는데 소요되는 시간이 오래 걸리는 단점이 여전히 존재한다[6].

3. 트랜잭션데이터베이스구축

웹 서버에 저장된 로그 파일은 인터넷 사용자가 한 웹 사이트를 액세스했던 모든 페이지 명과 시간 등이 기록되어 있다. [그림 1]에서의 첫번째 레코드는 IP 203.229.252.206인 사용자가 2003년 1월 8일 12시 47분 21초에 kokkuk.ac.kr/BBS/menu.html 문서를 액세스했으며 1537바이트가 전송되었음을 의미하고 있다. 그러나 이러한 웹 로그들에는 필요치 않은 부분과 오류 등도 포함되어 있기 때문에 탐사

하려는 패턴에 적합하게 전처리(pre-processing)하여 트랜잭션 데이터베이스를 구축해야 한다. 일반적인 방법은 사용자 IP와 액세스된 페이지를 정수로 1:1매핑(mapping)하고, IP주소를 주 키로 하고 액세스 시간을 보조 키로 하여 트랜잭션들을 오름차순으로 정렬하게 된다. 여기서 하나의 트랜잭션은 어떤 사용자가 액세스한 하나의 페이지로써 간주한다. 이러한 트랜잭션 데이터베이스를 이용하여 패턴탐사를 수행한 후에는 다시 정수를 해당 페이지로 매핑하여 디코딩하게 된다. 아래의 <표 1>, <표 2>, <표 3>은 사용자 및 액세스된 페이지의 매핑 테이블과 전처리된 트랜잭션 데이터베이스에 대한 예를 보이고 있다.

표 1> 사용자 매핑 테이블
<Table 1> Mapping table for users

매핑번호	사용자 IP 주소
1	203.225.124.45
2	203.232.116.90
3	203.247.200.68
4	203.252.134.73

<표 2> 액세스된 페이지의 매핑 테이블
<Table 2> Mapping table for accessed pages

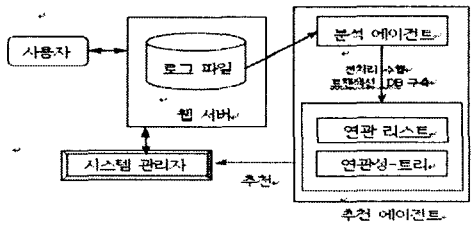
매핑번호	액세스된 웹 페이지
1	/
2	/HTTP/1.0
3	/bbs/HTTP/1.0
4	/bbs/menu.html

<표 3> 전처리된 트랜잭션 데이터베이스의 예
<Table 3> Example of a transaction database after preprocessing

사용자 ID	트랜잭션 시간	페이지
1	03/06/25	4
1	03/06/30	102
2	03/06/10	328
3	03/07/25	17

4. 시스템구조

본 논문에서 제안하고 있는 시스템의 구조는 [그림 2]와 같다. 사용자가 한 웹 사이트에 액세스를 하게 되면, 웹 서버에 로그가 남게 된다. 먼저, '분석 에이전트'가 이 정보를 전처리하여 트랜잭션 데이터베이스를 구축하고, '추천 에이전트'는 이를 이용하여 연관 리스트를 추출하고 웹 페이지들간의 구조화된 연관성-트리를 생성하여, 웹 사용자들이 빈번하게 접근하는 페이지들의 연관성을 시스템 관리자에게 추천하게 된다.



[그림 2] 시스템의 구조
[Fig. 2] The organization of system

4.1 빈발패턴의 추출 및 구조화

연관성을 발견하기 위한 방법은, Apriori와는

Input : A transaction database *DB*and a minimum support *min_sup*

Output : Frequent structured tree(FST)

[Step 1] Scan the transaction database *DB* once, and find the frequent 1-itemset.

[Step 2] Delete the non-frequent items in transaction database *DB* and reorganize transaction database *DB* having frequent items only. It is called reorganized transaction database *DB'*.

[Step 3] Create the frequent structured tree. The root of a tree labels it as "null". For each transaction (TR) in *DB'* do the following.

Let the items list in a TR be [*p*|*P*], where *p* is the first element and *P* is the remaining list. Call *construct_tree*([*p*|*P*], FST).

```
construct_tree([p|P], FST)
{
 if (FST has a child N such that N
     = p)
 then
    increment N's count by 1
 else
    create a new node N, and
    let its count be 1,
    its parent link be linked to FST.
    if (P is non-empty)
       construct_tree(P, N)
  }
```

[Step 4] For the created tree in [Step 3], call *reform_tree*([*n*|*N*], *FST*).

Let the node list be [*n*|*N*], where *n* is a subpath having one branch and *N* is the remaining subpath of the FST.

```
reform_tree([n|N], FST)
{
  while (N is non-empty) {
   if (n'sup < min-sup)
   if (n exists in other subpath)
      then
      add count of its node to that
      of existing one, and
      unlink its link to FST
      else
        unlink its link to FST
  }
  delete terminal nodes having not
  any other
  branch underneath root
}
```

[그림 3] 연관성-트리 구축 및 빈발패턴 탐사 알고리즘

[Fig. 3] Algorithm for tree construction and mining frequent patterns

전혀 다른 알고리즘으로써 후보 2-항목집합 이상은 생성하지 않으면서 최대 빈발 항목집합을 생성하는 방법을 시도하였으며, 빈발 1-항목집합을 생성한 후 전체 트랜잭션 데이터베이스의 크기를 효과적으로 줄여 데이터베이스의 탐색이 빠르게 이루어지도록 하였다. 수행된 알고리즘과 추출된 패턴의 구조화는 [그림 3]의 각 단계를 통해 이루어 진다.

Apriori에 근거하는 대다수 알고리즘에서 시간 비용이 큰 지속적인 데이터베이스 스캔의 목적은 생성된 후보패턴들이 빈발한 지를 검사하기 위한 것이기 때문에, 이를 해결하기 위하여 본 논문에서는 후보패턴 생성없이 빈발 항목집합으로만 되어 있는 간결한 자료구조인 트리를 구축하고, 각 노드(항목)에 관한 출현 횟수를 첨부해서 저장하여 데이터베이스 스캔없이 빈발한 지를 결정하는 지지도 계산을 대신할 수 있도록 하는 알고리즘을 개발하여 효율을 극대화하였다.

<표 4>를 예로 하여 알고리즘의 각 단계에 해당되는 과정을 보이면 다음과 같다.

<표 4> 트랜잭션 데이터베이스 *DB*의 예

<Table 4> A transaction database as running example

[illegible]	[illegible]	[illegible]
1	03/06/25	3
1	03/06/30	4, 6, 7
2	03/06/10	1, 2
2	03/06/15	3
2	03/06/20	9
3	03/06/25	3, 5, 7
4	03/06/25	3
4	03/06/30	4, 7
4	03/07/25	9
5	03/06/12	9

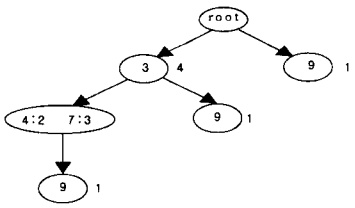
[Step 1] <표 1>의 *DB*에서 최소지지도 *min-sup* = 25% 라고 가정하면, <(1:1), (2:1), (3:4), (4:2), (5:1), (6:1), (7:3), (9:3)>>인 항목들의 리스트를 구할 수 있고, 빈발 1-항목집합 = {3}, {4}, {7}, {9}를 발견할 수 있다.

[Step 2] 빈발1-항목집합에 속하지 않는 모든 항목들을 제거한 새로운 트랜잭션 데이터베이스(*DB'*)를 생성(<표 5>)한다.

<표 5> 빈발 항목집합으로 구성된 *DB'* 의 예

<Table 5> A transaction database *DB'* as constructed only frequent itemsets

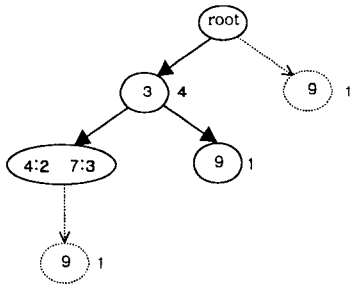
[Step 3] <표 5>의 *DB'* 를 이용하여 트리를 구축하면 [그림4]와 같다.



[그림4] 빈발 1-항목집합으로 구축된 트리

[Fig. 4] A tree created by frequent 1-itemset

[Step 4] 이 단계는 빈발패턴을 탐사하는 과정으로 트랜잭션 데이터베이스를 스캔하지 않고 구축된 트리를 이용하여 효율적으로 수행한다. [그림 4]에서 *min-sup* = 25% 이하인 노드를 조사하면 경로 '③→④⑦→⑨'에서 ⑨가 발견된다. 따라서 다른 경로에 ③을 부노드(parent node), ⑨를 자노드(child node)로 하는 경로 '③→⑨'가 존재하는 지를 찾게 되고, 발견되기 때문에 이 경로의 ⑨에 기존의 ⑨의 가중치 1을 더해준다. 동시에 경로 '③→④⑦→⑨'에서 ⑨의 링크를 제거한다. 만일 다른 경로에서 ③→⑨를 발견하지 못하면, 또 다른 경로 ④⑦→⑨를 찾아서 수행하게 된다. 더 이상의 서브트리가 없기 때문에, 루트 바로 아래에 있는 단말(terminal) 노드를 제거함으로써 알고리즘은 종료하게 된다. [그림 5]에서 *min-sup* 를 초과하는 실선으로 된 최대 빈발 항목집합을 구하게 된다. 여기서 점선으로 된 노드들은 제거된 것이다.



[그림 5] 패턴탐사가 수행된 후의 트리

[Fig. 5] A reformed tree after mining frequent patterns

Apriori를 기반으로 하는 알고리즘에서는 후보 빈발 항목집합을 구하고 빈발 항목집합을 산출할 때까지, 매번의 패스마다 트랜잭션 데이터베이스를 참조해야만 하는 단점이 있어 실제 대용량의 데이터베이스에 그대로 적용하기에는 문제를 갖고 있다.

4.2 패턴탐사결과에따른추천

패턴탐사 후 얻어진 트리는 전처리 과정에서 정수로 매핑된 것이기 때문에 이들을 실제의 페이지로 변환하는 디코딩 과정을 거쳐야

한다. 그러나 이 과정은 매우 간단한 것으로서 전처리 과정에서 생성된 <표 2>의 액세스된 페이지의 매핑 테이블을 이용하여 쉽게 변환할 수 있다. 본 논문에서 제안하고 있는 추천 에이전트의 장점은 패턴탐사의 결과를 테이블 형태로 표현하는 것이 아닌 [그림 5]와 같이 트리 형태 그대로 제공하고 있다는 것이다. 예를 들면, 기존의 테이블 형태의 표현 'A⇒B'는 "A를 액세스한 후에 B를 액세스할 확률이 높다."로 해석할 수 있는 정도지만, 트리형태에서는 A와 B의 관계에 가중치가 함께 제공되며 또한, 다른 경로와의 관계가 도식적으로 표현되어 제공되기 때문에 시스템 관리자가 탐사된 패턴들간의 관련 정도를 정확하고 쉽게 이해하여 제공된 정보를 효율적으로 이용할 수 있을 것이다.

4.3 트리의수정

사용자의 액세스 패턴이 점점 쌓여 갈수록 액세스 패턴의 성향을 트리에 반영해주어야 보다 정확하며 신뢰성있는 결과를 얻을 수 있게 된다. 따라서 본 논문에서는 항상 최신의 패턴을 반영하기 위하여 사용자의 액세스 패턴이 특정 임계치(threshold)를 초과할 때마다 트리가 수정되도록 하였다.

예를 들어, 새로운 패턴에서B G의 지지도가 4로 변경되었다면 B G에는 *sup*old *sup*new를 추가하고, 같은 레벨의 나머지 노드에는 *sup*old *sup*new / (동일 레벨 노드의 개수-1) 만큼 각각 감소시켜 준다. 이와 같은 방식으로 사용자의 액세스 패턴에 맞추어 트리를 수정해 나간다면 사용자의 액세스 패턴을 보다 정확하고 효과적으로 반영할 수 있을 것이다.

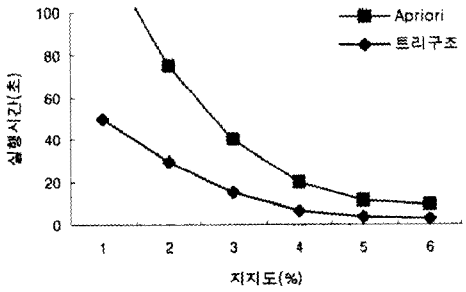
5. 실험 및 평가

본 논문에서의 실험은 UCI (University of California at Irvine)의 machine learning data 생성 프로그램을 이용하여 얻은 실험 데이터를 사용하여 연관규칙 탐사에서는 Apriori 알고리즘과, 순차패턴 탐사에서는 AprioriAll 알고리즘과 본 논문에서 제안하고 있는 트리구조 알고리즘과의 최소지지도에 따른 실행시간(runtime)의 성능을 비교하고 분석하였다.

제안된 알고리즘의 성능평가를 위해서 주기억장치 512MB를 장착한 2.4G Pentium-Ⅳ PC에서 실험하였으며, 운영체제는 Microsoft Windows/NT ver. 4.0, 언어는 Visual C++ ver. 6.0을 사용하였다. 서로 다른 기계구조에서는 같은 알고리즘들에 대한 실행시간이 크게 다를 수 있기 때문에, 기존의 Apriori-based 알고리즘들과 제안한 알고리즘의 성능을 평가하기 위하여 동일한 하드웨어에서 알고리즘들을 구현하고 같은 환경에서 비교하였다. 여기서 사용된 실행시간은각 알고리즘에서 측정된 CPU시간이다.

5.1연관 규칙 탐사에서의 성능 분석

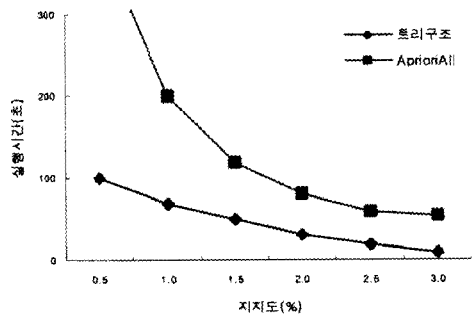
[그림 6]은 연관규칙 탐사를 위한 대표적 알고리즘인 Apriori와 본 논문에서 개발한 알고리즘과의 최소지지도에 따른 실행시간의 차이를 보이고 있다. 여기서 제안하고 있는 알고리즘이 Apriori보다 매우 효율적으로 나타나고 있는데, Apriori는 수행방법의 특성 때문에 최소지지도가 작을수록 빈발 항목집합의 길이뿐 아니라 수 또한 매우 증가하기 때문이다. 이는 Apriori가 처리해야만 하는 후보항목집합의 수가 많음을 의미하며, 트랜잭션 데이터베이스를 탐색하여 후보항목집합이 빈발한 지를 계산하는 패턴매칭 과정에 시간이 오래 걸리기 때문이다. 그러나 제안하는 알고리즘에서는 후보집합을 생성하지 않으며, 트랜잭션 데이터베이스는 단지 두 번만 스캔하여 트리를 구축하고 탐색을 수행하기 때문에 계산 비용이 상대적으로 매우 적게 된다.



[그림 6] 지지도 임계치에 따른 실행시간
[Fig. 6]Runtime with support threshold

5.2 순차 패턴 탐사에서의 성능 분석

[그림 7]은 순차패턴 탐사에서의 실행시간을 보이고 있는데, AprioriAll에서는 패턴의 길이가 긴 트랜잭션들이 존재하게 되면 이에 따른 후보 항목집합의 수가 기하급수적으로 생성될 수 있



[그림 7] 지지도 임계치에 따른 실행시간
[Fig. 6]Runtime with support threshold

기 때문에, 후보 항목집합 생성에 따른 시간 비용이 매우 커지게 된다.

제안하고있는 알고리즘은 최소지지도가 감소하는 경우에 단지 선형적으로 증가하는 우수한 결과를 보이고 있다.

6. 결론

본 논문에서는 트랜잭션 데이터베이스에서 패턴들 사이에서의 연관성을 빠르게 탐사하고, 탐사된 결과를 테이블 구조에서 보여지는 형태가 아닌 트리로 표현하는 추천 에이전트를 개발하여, 보다 구체적이며 효율적인 패턴들의 다양한 관계를 시스템 관리자에게 제공함으로써, 웹 사이트의 내용이나 구조를 재배치하거나 인터넷 쇼핑몰에서는 소비자의 구매패턴을 분석하여 상품을 추천하는데 이용할 수 있도록 하였다. 제안된 알고리즘의 성능을 입증하기 위하여 기존의 연관규칙, 순차패턴 탐사 알고리즘인 Apriori와 AprioriAll과의 성능을 비교하고 분석하였다. 이실험의 결과 트랜잭션 데이터베이스에서 패턴들의 연관성을 표현하는 빈발 항목집합을 탐사하는 실행시간에서 제안된 알고리즘이 우수함을 보였다.

탐사결과를 트리 형태로 제공하기 때문에 보다 더 구체적인 패턴들간의 연관성을 알 수 있게 되며, 또한 웹 서버의 로그 파일에 추가되는 최근 일정량의 액세스 패턴들이 발생될 때마다, 트리를 수정하여 보다 정확하고 신뢰성 높은 연관성을 구할 수 있게 하였다.

향후 과제로는 본 연구의 확장으로서 트리의 수정 방법을 개선하기 위하여, 특정 지지도 임계치에 따라 트리에 반영될 수 있도록 통계적 기법을 연구하는 것이다. 또한 전처리 과정에서 불필요한 데이터를 효율적으로 제거하여 보다 유용한 트랜잭션으로 가공할 수 있는 방법과 개인화 서비스를 제공하는 방법 및 순회패턴 탐사에 관한 연구 과제들이 있다.

참고 문헌

[1] Agrawal R., Imielinski T., and Swami A., "Database Mining: A Performance

Perspective", IEEE Tran. on Knowledge and Data Engineering, Vol. 5, No. 6, pp. 914-925, 1993.
[2] Agrawal R, and Srikant R., "Fast Algorithms for Mining Association Rules in Large Databases", In Proc. Of the 20th Int. Conf. on Very Large Databases, 1994.
[3] Bettini C., Wang X.S, and Jajodia,"Mining Temporal Relationships with Multiple Granularities in Time Sequences", Data Engineering Bulletin, 21:32-38, 1998.
[4] Bchner A.G., Baumgarten M., Mulvenna M.D., Anand S.S, and Hughes J.G., "Navigation Pattern Discovery from Internet Data", WebKDD '99, 1999.
[5] Cooley R., Mobasher R., and Srivastava J,"Web Mining: Information and Pattern Discovery on World Wide Web, In Proc. 9th IEEE Int. Conf. On Tools with Artificial Intelligence, 1997.
[6] Han J.,Pei J., Mortazavi-Asi B. and Pinto H., "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth", the 17th Intl' Conf. on Data Engineering(ICDE '01), Heidelberg, Germany, April 2001.
[7] Park J.S., Chen M.S., and Yu P.S., "An Effective Hash-Based Algorithm for Mining Association Rules", In Proc. Of ACM SIGMOD, pp. 175-186, 1995.
[8] Srikant R., and Agrawal R., "Mining Sequential Patterns: Generalizations and Performance Improvements", In Proc. of 5th Int. Conf. on Extending Database Technology, pp. 3-17, 1996.
[9] Funnel web professional. http://www.activeconcepts.com, 1999.
[10] Hit list commerce. http://www.marketwave.com, 1999.
[11] Webtrends log analyzer. http://www.webtrends.com, 1999.

김성학

1985년 건국대학교 수학과 졸업 (이학사)
1987년 건국대학교 대학원 전자계산학과(공학석사)
1995년 건국대학교 대학원 컴퓨터공학과 박사수료
1987년 삼성종합기술원 정보시스템연구소 연구원
1989~현재 유한대학 컴퓨터정 보과 교수
관심분야 : 인공지능, 전자상거래, 데이터마이닝

이창훈

1975년 연세대학교 수학과 졸업 (이학사)
1977년 한국과학기술원 전산학과 (공학석사)
1993년 한국과학기술원 전산학과 (공학박사)
1980년~현재 건국대학교 정보통신대학 컴퓨터정보통신공학과 교수
2002년6월~현재 정통부전산관리소 전파방송관리통합정보 시스템 구축사업 자문위원
2002년 9월~현재 정통부 정보격차해소 자문위원회 위원
2002년 5월~현재 한국정보통신 산업현회 정보보호마크인증 위원회 위원
관심분야 : 인공지능, 지식기반시스템, 데이터마이닝, 정보통신 및 보안, 컴퓨터 네트워크