상품간 연관 규칙의 효율적 탐색 방법에 관한 연구 : 인터넷 쇼핑몰을 중심으로

오은정*, 오상봉**

A Fast Algorithm for Mining Association Rules in Web Log Data

Eun-Jung Oh, Sang-Bong Oh

Abstract Mining association rules in web log files can be divided into two steps: 1) discovering frequent item sets in web data; 2) extracting association rules from the frequent item sets found in the previous step. This paper suggests an algorithm for finding frequent item sets efficiently. The essence of the proposed algorithm is to transform transaction data files into matrix format. Our experimental results show that the suggested algorithm outperforms the Apriori algorithm, which is widely used to discover frequent item sets, in terms of scan frequency and execution time.

1. 서론

WWW(World Wide Web) 서비스의 등장 이후, 계속적인 인터넷의 발달과 사용 환경이 편리해 짐에 따라 사용자는 급속도로 증가하였다. 따라서 웹사이트 운영자들은 사용자의 접속 패턴을 파악하고, 그에 맞는 서비스를 제공하기 위한 방법을 찾기 시작하였다. 가장 먼저 관심을 보인 것이 웹 서버에 남아 있는 로그 파일을 분석하는 것이었다. 하지만 초기의 로그 분석은 총 방문자 수나 시간대별 접속 횟수, 에러 페이지 발견 등의 통계적 수치만을 알아내어, 동적이고 개인화된 웹 서비스의 개발에는 많은 한계를 가져다주었다. 이러한 문제를 해결하기 위해 데이터 마이닝 기법을 웹에 적용하려는 시도가 이루어 졌고, 이를 웹 마이닝(Web Mining)이라 한다[Kosala & Blockeel, 2000].

웹 마이닝의 패턴 발견 과정에서 가장 대표적인 것이 데이터들 간의 연관 규칙을 찾아내는 것이다. 일반적으로 연관 규칙을 생성하는 과정은 크게 빈발 항목(frequent item sets)을 찾아내고, 이들로부터 연관 규칙을 생성하는 과정으로 구성된다. 빈발 항목을 발견하기 위한 대표적인 방법으로 알려진 Apriori 알고리즘은 방대한 양의 트랜잭션 데이터베이스를 각 패스마다 계속적으로

* 대전대학교 대학원 정보통신공학과

** 대전대학교 정보통신공학과 교수

스캔해야 하는 문제점을 지니고 있어, 효율적인 빈발 항목 탐색을 위한 다양한 알고리즘들에 대한 연구가 현재 계속 진행되고 있다[박종수 외 2인, 1998].

본 연구에서는 좀 더 효율적으로 빈발 항목을 탐색하기 위하여 트랜잭션 데이터베이스를 Matrix 형태로 변환하고 빈발 항목 패턴을 발견하므로써 전체 트랜잭션 데이터베이스의 스캔 횟수와 빈발 항목을 발견하기 위한 탐색 시간을 줄이고자 한다.

2 기존의 연관 규칙 알고리즘

2.1 Apriori 알고리즘

Apriori 알고리즘에서는 각 패스마다 후보 집합을 구성한 후에 각 후보 항목 집합의 발생 빈도수를 계산하고, 사용자가 정의한 최소 지지도를 기초로 하여 빈발 항목집합을 결정한다. Apriori-gen이라는 함수를 생성하여 후보 항목집합을 만드는데 효율적인 방법을 제시하였다. Apriori 알고리즘은 후보들을 생성하는 Apriori-gen 함수 호출과 후보들에 대해 지지도를 계산하는 카운팅 단계로 구성되어 있다. 지지도를 계산하기 위하여 전체 트랜잭션 데이터베이스를 스캔하며, 계산을 능률적으로 수행하기 위해 해쉬 트리에 후보 항목집합들을 저장한다[Agrawal. & Srikant, 1994; Mannila et al., 1994].

2.2 DHP 알고리즘

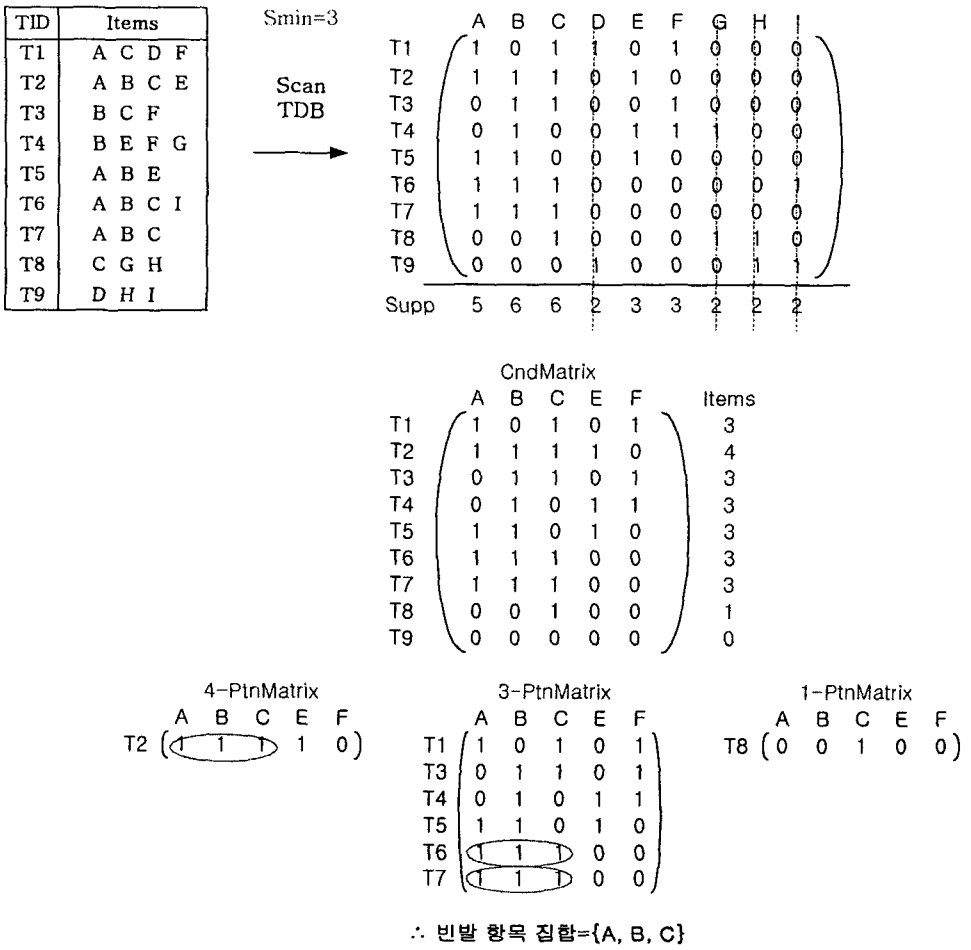
DHP 알고리즘은 Apriori 알고리즘과 비교하여 후보-2 항목집합들을 효율적으로 작게 구하는 방법과 이것을 기초로 전체 트랜잭션의 크기와 개수를 줄여나가는 방법을 제시하였다. 먼저 항목집합의 개수가 2개인 트랜잭션을 데이터베이스에서 추출하여 해쉬 테이블을 만든 다음에 $L_1 * L_1$을 했을 때 해쉬 테이블에 있는 것과 비교하여 최소 지지도를 넘는 것만 가지고 C_2를 만든다. DHP 알고리즘은 초기 단계의 두 번째 패스까지만 전체 데이터베이스를 스캔하고 이후 패스에서는 데이터베이스의 크기를 줄여간다[Park et al., 1995].

2.3 Partition 알고리즘

Partition은 I/O 와 CPU의 오버헤드를 줄임으로써 방대한 데이터를 처리하는 빠르고 효율적인 알고리즘으로 제시되었다. 이전 알고리즘들의 주요 단점은 데이터베이스에 다중 패스를 요구한다는 것이다. Partition 알고리즘은 방대한 데이터베이스로부터 의미있는 연관 규칙을 만들어 내기 위해 2회 스캔만으로 disk I/O와 CPU 오버헤드를 획기적으로 줄일 수 있다. 알고리즘은 크게 두 단계로 나눌 수 있다. 첫 단계에서는 데이터베이스를 중복되지 않는 크기로 partition하고 한번에 한 개의 partition만을 고려하여 그 안에서 빈발 항목집합들을 생성한다. 각각의 partition에서 생성된 빈발 항목집합들은 잠재적인 빈발 항목집합들로 통합된다. 두 번째 단계에서 각 항목에 대한 실제 지지도가 생성되고 빈발 항목집합들이 식별된다[Savasere et al., 1995].

2.4 DIC 알고리즘

DIC는 빈발 항목집합들을 찾는데 있어서, 각 패스에서 카운트되는 항목 집합들의 수를 상대적으로 적게 유지하면서, 데이터베이스 스캔 회수를 줄이는 알고리즘이다. 기존의 Apriori 알고리즘은 레벨 단위로 진행되면서 빈발 항목집합들을 찾는 반면에 DIC 알고리즘에서는 간격(interval., M)을 두고, 항목집



(그림 1) 제안하는 Matrix 알고리즘 과정

합의 크기를 증가시키면서 동시에 빈발 항목집합들을 찾는다. 이때, 각각의 항목집합마다 카운터기를 유지하고 트랜잭션을 읽었을 때, 해당 트랜잭션에서 해당 항목집합들이 나타나면 그 카운터기를 증가시키므로써 항목 집합들의 상태를 변화시켜서 최종적인 빈발 항목집합들을 찾는다. 이러한 방법으로 알고리즘이 진행된다면, 빈발 항목집합들을 찾는데 있어서, 만일 Apriori 알고리즘에서 3번의 스캔이 요구된다면, DIC 알고리즘에서는 1.5번의 스캔으로 빈발 항목집합들을 찾을 수 있다[Brin et al., 1997].

3 제안하는 Matrix 알고리즘

본 연구에서 제안하는 빈발 항목집합 탐색 알고리즘은 전체 트랜잭션 데이터베이스를 Matrix 형태로 변환한 후 항목 패턴이 같은 트랜잭션을 찾아내어 빈발 항목을 탐색한다.

(그림 1)은 제안하는 Matrix 알고리즘의 과정을 설명한 것이다. 첫 번째 단계에서는 전체 트랜잭션 데이터베이스를 스캔하여 Matrix형태로 변환한다. 행에는 트랜잭션 데이터베이스에 있는 항목 List가, 열에는 고유한 ID를 갖는 트랜잭션(TID)이 부여된다.

Matrix에는 전체 트랜잭션 데이터베이스에 나타난 항목을 1, 그렇지 않은 항목은 0으로 표시한다. 다음은 각 항목에 대한 지지도를 계산하여 미리 사용자가 정의한 지지도와 같거나 이상인 트랜잭션을 찾아 후보 Matrix를 생성한다. 세 번째 단계에서는 각 트랜잭션의 항목 수를 계산한 후 같은 항목 수를 갖는 트랜잭션들로 후보 Matrix를 Partition한다. 이를 항목별 Matrix라 하며, 항목 수가 가장 많은 4-partition matrix부터 항목 패턴이 같은 트랜잭션을 찾는다. k-partition matrix부터 (k-1)-partition matrix 순서로, 자신의 항목 수와 같거나 많은 항목별 matrix에서 같은 항목 패턴을 찾아나간다. 같은 패턴의 트랜잭션을 발견했을 때에는 정의한 지지도에 합당한지 여부를 판단한 후, 합당하다면 빈발 항목을 발견하는 과정은 마치게 된다. 예를 들어, T_2, T_6, T_7에서 같은 항목 패턴을 찾았고, 정의한 지지도(S_{min}=3)에 합당하므로 빈발 항목집합 {A, B, C}가 생성되었다.

4. 성능 평가 및 결과 분석

4.1 평가 방법

본 연구에서는 빈발 항목을 탐색하는 가장 대표적인 방법인 Apriori 알고리즘과 비교하여 현재 웹에서 운영 중인 소규모 종합 쇼핑몰의 로그 파일을 이용하여 제안하는 알고리즘의 성능을 평가하였다.

연관 규칙 알고리즘의 성능 비교는 크게 두 가지 측면에서 분석하였다. 첫 번째로는, 전체 트랜잭션 데이터베이스 스캔 횟수를 비교하고, 두 번째로는 정의한 지지도를 변화시키면서 빈발 항목을 발견하는 탐색 시간을 가지고 두 알고리즘을 비교하였다. 이때 사용된 트랜잭션 데이터베이스는 2003년 4월부터 9월까지 사용자들의 구매 항목으로 이루어졌다.

4.2 평가 결과 및 분석

<표 1>은 트랜잭션 데이터베이스 스캔 횟수를 비교한 결과이다. 연관 규칙의 기존 Apriori 알고리즘(A)은 빈발 항목을 탐색하기 위해서 각 단계마다 전체 트랜잭션 데이터베이스를 스캔하므로, k번의 단계를 거쳐야 한다면 스캔 횟수도 k번에 해당한다. 그러나 제안하는 Matrix 알고리즘(B)은 전체 트랜잭션 데이터베이스를 Matrix 형태로 변환할 때 즉, 첫 단계에서만 스캔하므로 한 번에 해당한다.

<표 1> 트랜잭션 데이터베이스 스캔 횟수

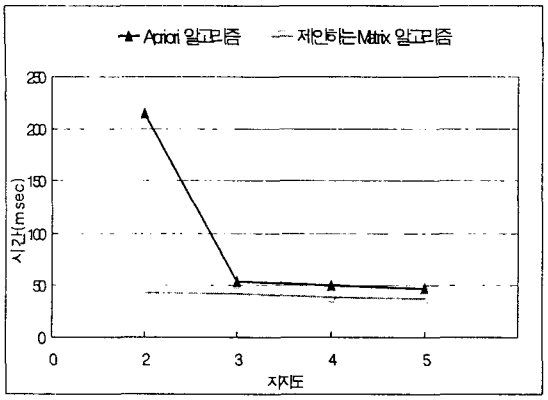
알고리즘	A	B
트랜잭션 데이터베이스 스캔 횟수	k	1

정의한 지지도를 변화시키며 탐색 시간을 비교하는 과정에서는 여러 번의 실험을 거친 평균값이다. 이때, 정의한 지지도가 낮다는 것은 그만큼 빈발 항목집합이 많다는 것을 의미한다.

<표 2> 빈발 항목 탐색 시간

(단위:msec)

알고리즘 \ 지지도	2	3	4	5
A	215	53.4	49.4	46.7
B	43	42.5	38	37.5



(그림 2) 빈발 항목 탐색 시간

<표 2>와 <그림 2>에서 볼 수 있듯이 제안하는 Matrix 알고리즘이 기존의 Apriori 알고리즘보다 짧은 시간에 빈발 항목집합을 발견하였다. 특히 Apriori 알고리즘은 지지도가 낮을수록 선택되는 후보 항목수가 많으므로 지지도별로 탐색 시간에 많은 차이를 보였다. 이러한 결과는 다양한 상품을 지니고 있는 종합 쇼핑몰일수록 트랜잭션 데이터베이스에는 많은 수의 항목이 포함되므로 기존의 Apriori 알고리즘보다 제안하는 Matrix 알고리즘을 이용한다면 더 나은 효과를 가져올 것이라 기대된다.

5. 결 론

본 논문에서는 접속 패턴을 발견하는데 대표적인 연구인 연관 규칙에서 빈발 항목을 효율적으로 찾아내기 위하여 전체 트랜잭션 데이터베이스를 Matrix 형태로 변환하고 빈발 항목 패턴을 발견하는 알고리즘을 제안하였다. 또한 실제 웹에 운영 중인 쇼핑몰의 로그 파일을 이용하여 기존의 Apriori 알고리즘과 비교하여 제안하는 알고리즘의 성능을 평가를 하였다. 예측한 바와 같이 전체 트랜잭션 데이터베이스 스캔 횟수 면에서 기존의 Apriori 알고리즘은 각 단계마다 트랜잭션 데이터베이스를 스캔해야 하므로 횟수를 예측 할 수 없는 반면에 제안하는 Matrix 알고리즘은 전체 트랜잭션 데이터베이스를 Matrix 형태로 변환하는 첫 단계에서만 필요하므로 더 나은 결과를 볼 수 있었다. 탐색 시간 면에서도 큰 시간차를 보일 수는 없었으나 기존의 Apriori 알고리즘보다 지지도가 낮은 경우엔 약 50%, 지지도가 높은 경우엔 약 20%의 향상된 결과를 확인 할 수 있었다.

본 연구에서 제안하는 Matrix 알고리즘을 통하여 얻은 결과를 이용하여 쇼핑몰 운영자들은 웹사이트를 개선하거나, 관련 상품의 추천을 통하여 수익을 높이는데 응용 할 수 있으며 또한 트랜잭션을 다양하게 확장하여 잠재적인 고객들을 위해 응용 할 수 있을 것이다.

참 고 문 헌

[1] 박종수, 유원형, 홍기형, "연관 규칙 탐사와 그 응용", 정보과학회지, 제 16권, 제 9호, pp. 37-44, 1998.

[2] Agrawal R. and R. Srikant, "Fast algorithms for mining association rules", In Proceedings of the 20th VLDB Conference, Santiago, Chile, Sept, 1994.

[3] Brin S., R. Motwani, J.D. Ullman, and S. Tsur, "Dynamic Itemset Counting and Implication Rules for Market Basket Data", In Proceedings of ACM SIGMOD Conference on Management of Data, Tucson, Arizona, pp. 255-264, May, 1997.

[4] Kosala R. and H. Blockeel, "Web mining research : A survey",

SIGKDD Explorations - Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining, Vol2. No.1, pp. 1-15, July, 2000.

[5] Mannila H., H. Toivonen, and A.I. Verkamo, "Efficient Algorithms for Discovering Association Rules", In AAAI Workshop on Knowledge Discovery in Databases(KDD'94), U.M. Fayyad and R. Uthurusamy (eds.), Seattle, Washington, pp. 181-192, July, 1994.

[6] Park J.S., M.S. Chen, and P.S. Yu, "An effective hash-based algorithm for mining association rules", In Proceedings of ACM SIGMOD Conference on Management of Data, San Jose, California, pp. 175-186, May, 1995.

[7] Savasere A., E. Omiencinsky, and S. Navathe, "An efficient algorithm for mining association rules in large databases", In Proceedings of the 21st VLDB Conference, Zurich, Swizerland, pp. 432-444, 1995.

오은정(Eun-Jung Oh)

2002년 2월: 대전대학교 정보통신공학과 (공학사)

2002년 3월～현재: 대전대학교대학원 정보통신공학과 석사과정

<관심분야> 데이터 마이닝, 전자상거래

오상봉(Sang-Bong Oh)

1983년 2월: 서울대학교 경제학과 (경제학사)

1990년 2월: 학국과학기술원 경영과학과 (공학석사, 박사)

1993. 3월～현재: 대전대학교 정보통신공학과 교수

<관심분야> 정보시스템 분석/설계 및 응용, Expert Systems and AI Applications, MIS/DDS, 전자 상거래