SQL을 이용한 연관 규칙 탐사 시스템

전수정*, 김영지*, 우용태*

An Association Rules Mining System based-on SQL

Su-Jung Jun*, Young-Ji Kim*, Yong-Tae Woo*

요 약

본 논문에서는 연관 규칙 탐사 시스템을 설계하고 구현하였다. 본 시스템은 관계형 데이터베이스의 표준 질의어를 이용하여 사용자가 제시한 질의 조건을 만족하는 항목집합에 대해 다양한 형태의 연관 규칙을 탐사하기 위한 시스템이다. 질의처리 모듈에서는 사용자가 제시한 조건을 만족하는 질의를 동적으로 구성하여, 연관 규칙 탐사를 위해 사용되는 대상 트랜잭션 데이타베이스의 범위를 조절할 수 있다. 연관 규칙을 발견하기 위한 후보 항목집합을 생성하기 위해 연관 규칙 탐사 알고리즘을 사용하였다. 연관 규칙 알고리즘에서는 한 트랜잭션 데이타에 대해 생성될 수 있는 후보 항목집합을 배열을 이용하여 처리하는 효율적인 방법을 제안하였다.

Key words : 연관 규칙 탐사, 데이타마이닝, KDD

1. 서론

최근 들어 기업 경영을 위해 생성되는 데이타 양은 급속도로 증가하고 있지만, 기업의 경쟁력을 강화할 수 있는 유용한 정보를 분석하기 위한 기법이 부족하여 의사결정 과정이 점점 어려워지고 있다. 또한 급변하는 시장 환경에서 소비자의 구매 패턴을 분석하여 기업 경영에 활용할 수 있는 고급 지식 정보의 필요성이 증가되고 있다. 이러한 문제점을 개선하기 위하여 대용량의 데이타로부터 기존에 알려지지 않은 유용한 정보를 찾기 위한 데이타마이닝(Data Mining) 기술에 대한 연구가 활발하게 진행되고 있다[1]. 데이타마이닝이란 목표 데이타로부터 탐사 기법을 적용하여 의미 있는 패턴을 발견하기 위한 기법으로 KDD (Knowledge Discovery and Data Mining) 시스템에서 핵심적인 단계중의 하나이다.

지식 탐사 과정에서 대상 지식의 종류에 따라 적용할 수 있는 데이타마이닝 기법도 달라지게 된다. 최근에 연구가 활발하게 진행되고 있는 데이타마이닝 분야는 대량의 트랜잭션 집합에 대하여 동시에 발생하는 항목들간의 관련성을 연구하는 연관 규칙(Association Rules) 탐사[2], 일정한 시간 간격을 두고 발생하는 항목들의 시간적인 함수관계를 연구하는 연속 패턴(Sequential Patterns) 탐사[3], 동시에 발생하는 항목들을 가장 잘 표현할 수 있는 특징을 찾기 위한 분류 규칙(Classification Rules) 탐사[4] 등이 있다.

이 중에서 연관 규칙 탐사의 주된 연구 방향은 데이타 항목의 증가에 따라 기하급수적으로 증가하는 알고리즘 수행 시간을 개선하기 위한 방법과 대량의 연관 규칙 집합에서 의미 있는 연관 규칙을 적절하게 선택하기 위한 방법이다[5]. 대부분의 연관 규칙 탐사 알고리즘에서 전체 수행 속도에 영향을 미치는 주된 요인은 트랜잭션 집합에서 일정한 최소 지지도를 만족하는 빈발 항목집합을 추출하는 데 걸리는 시간이다. 이러한 알고리즘의 수행 속도

* 창원대학교 자연과학대학 전자계산학과

개선을 위해 연관 규칙의 질적 평가를 위한 최소 임계값을 기준으로 광범위한 탐색 공간을 효과적으로 전지하기 위한 방법에 대한 연구가 이루어지고 있다. 이러한 연관 규칙 탐사 알고리즘은 SETM[8], Apriori[6], AprioriTid[6], Sampling[7] 알고리즘 등이 개발되었다.

본 논문에서는 관계형 데이타베이스의 표준 질의어를 이용하여 사용자 질의 조건을 만족하는 항목집합에 대해 다양한 형태의 연관 규칙을 탐사하기 위한 연관 규칙 탐사 시스템을 구현하였다. 예를 들어 쇼핑몰에서 소비자들의 구매 패턴을 분석하는 과정에서 단순히 하나의 기준에 의해 연관성을 찾게되면 무의미한 연관 규칙이 대량으로 양산될 수 있다. 또한 계절이나 연령, 학력 등과 같은 특정 요인에 종속적인 연관 규칙을 찾는데 어려움이 있다.

본 연관 규칙 탐사 시스템은 크게 질의처리 모듈과 연관 규칙 탐사 모듈로 구성된다. 먼저, 질의처리 모듈은 사용자가 제시한 조건을 만족하는 질의를 동적으로 구성하여 제한된 범위의 데이타에 대한 연관 규칙을 탐사하기 위한 기능을 담당한다. 질의처리 모듈을 통하여 계절과 같은 시간 조건이나 학력 수준 같은 범위를 지정하여 다양한 형태의 항목집합에 대한 연관 규칙 탐사가 가능하다. 연관 규칙 탐사 모듈은 연관 규칙을 발견하기 위한 후보 항목집합을 생성하기 위해 사용된다. 첫 번째 모듈에서 검색된 트랜잭션 데이타와 최소 지지도를 입력받아 후보 항목집합을 생성하면서 동시에 각 후보 항목집합에 대한 지지도를 계산한 후, 최소 지지도 이상의 빈발 항목집합을 추출한다.

본 논문에서 사용한 연관 규칙 탐사 알고리즘에서는 한 트랜잭션 데이타에 대해 생성될 수 있는 후보 항목집합을 배열로 처리하는 방법을 제안하였다. 관계형 데이타베이스 상에서 SETM은 각 단계의 빈발 K-항목집합을 생성하기 위해 전체 데이타베이스를 최소 2*K 스캔한다. 그러나 본 논문에서 제안한 방법은 후보 1-항목집합과 후보 2-항목집합을 한번의 데이타베이스 스캔을 통해 생성하지만 빈발 K-항목집합을 구하기 위해 데이타베이스를 K-1번 스캔하게 된다. 이러한 방법을 통해 데이타베이스에 대한 스캔 횟수를 줄여서 디스크 I/O를 최소화하였다. 그리고 후보 항목집합에 대한 지지도 계산시 한 트랜잭션에 포함된 항목의 개수가 후보 K-항목집합의 개수보다 적은 것은 비교 대상에서 제외시켜 탐색공간을 축소시켰다.

IBM의 Quest group에서 제공하는 실험 데이타 생성 프로그램을 이용하여 1,000개의 항목과 25,000건의 트랜잭션 집합을 생성하여 실험하였다. 실험 결과 제안한 기법은 기존의 Apriori 알고리즘보다 더 빠르게 수행되었다.

2. 연관 규칙 탐사 알고리즘

2.1. 연관 규칙의 정의

연관 규칙은 하나의 트랜잭션에서 동시에 발생할 수 있는 항목들의 집합으로 다음과 같이 정의한다. 먼저, I = {1, 2, 3, ···, m}을 항목들의 집합, D를 트랜잭션의 집합이라 하면 각 트랜잭션 Tid는 T ⊆ I인 항목들의 집합이다. 임의의 항목집합 X, Y에 대해 X⊆I인 항목집합에 대하여 X⊆T이면 T는 X를 포함한다고 말한다. 여기서 X⊆I, Y⊆I이고 X∩Y=Ø이다. 이 때 연관 규칙은 X⇒Y 로 표현한다. X⇒Y 의 의미는 지지도 support(X)가 트랜잭션 집합에서 X를 포함하는 트랜잭션의 수라고 할 때, 트랜잭션 집합에서 $\frac{support(X \cup Y)}{support(X)}$ 의 신뢰도(Confidence)로 X, Y가 동시에 발생한다는 것을 의미한다. 여기서 support(X)는 전체 트랜잭션에서 X∪Y를 만족하는 트랜잭션의 비율이다. 예를 들어 "빵과 버터를 구매한 고객의 90%가 우유도 같이 구매한다"라는 연관 규칙에서 지지도는 (빵, 버터, 우유를 구입한 트랜잭션)/(전체 트랜잭션)이고, 신뢰도 90%는 (빵, 버터, 우유를 구입한 트랜잭션)/(빵, 버터를 구입한 트랜잭션)의 비율이다.

연관 규칙 탐사는 최소 신뢰도(minconf)와 최소 지지도(minsup)를 만족하는 모든 연관 규칙 항목집합을 찾는 과정으로 이러한 항목집합을 빈발 항목집합(Frequent itemsets) L_k이라 하며, 빈발 항목집합을 생성하기 위한 최소 지지도를 적용하기 전의 항목집합을 후보 항목집합(Candidate itemsets) C_k이라 한다[6].

2.2. 기존 연관 규칙 탐사 알고리즘

기존의 연관 규칙 탐사 알고리즘에서는 빈발 항목집합이 될 가능성이 있는 후보 항목집합을 별도로 생성한다. 또한 알고리즘에 따라 후보 항목집합의 생성 크기, 순서 또는 생성 빈도가 서로 차이가 난다. 따라서 후보 항목집합들 중에서 빈발 항목집합을 찾기 위하여 전체 트랜잭션 데이타베이스를 액세스하여 각 후보 항목집합에 대한 지지도를 계산하는 과정이 필요하다.

2.2.1. SETM 알고리즘

SETM 알고리즘은 빈발 항목집합을 찾기 위해 관계형 데이타베이스를 이용하여 SQL의 조인(Join)과 정렬-조인(Sort-Merge)을 사용한 알고리즘이다. SETM에서는 먼저 후보 항목집합을 생성한 후에 지지도를 계산하고, 빈발하지 않는 항목들을 제거하기 위해 데이타베이스를 반복적

으로 생성하여 수정한다[8]. SETM은 각 단계별로 실행 시 모든 빈발하지 않은 항목들은 이미 제거된 것으로 가정한다. L_1은 동일한 Tids 상에서 셀프 조인을 실행하여 각 트랜잭션에 대한 모든 후보 2-항목집합 C_2를 생성한다. 그리고 L_2를 생성하기 위해 C_2를 먼저 정렬하여 지지도를 계산한 후 빈발하지 않은 항목집합을 제거하여 모든 빈발 2-항목집합을 포함하는 L_2를 생성한다. 이 때 L_2는 Tids 대신에 항목집합 데이타베이스로 저장된다. 다음 단계의 모든 후보 K-항목집합과 빈발 K-항목집합은 C_2와 L_2의 생성 과정을 반복하여 만든다. 이러한 SETM 알고리즘의 문제점은 후보 K-항목집합과 빈발 K-항목집합을 생성할 때마다 모든 트랜잭션에 대하여 셀프-조인과 정렬-병합을 반복하여 대량의 중간 결과를 생성하게 된다. 따라서 SETM 알고리즘은 AIS 알고리즘보다 더 적은 후보 항목집합을 생성하지만 별도의 정렬 시간이 필요하며, 각 트랜잭션들에 대해 중복된 후보 항목집합을 생성하는 문제점을 가지고 있다.

2.2.2. Apriori 알고리즘

최초의 연관 규칙 탐사 알고리즘은 1993년에 R. Agrawal 등에 의해 제안된 AIS 알고리즘이다[2]. Apriori 알고리즘은 AIS와 SETM의 문제점을 해결하기 위해 제안된 알고리즘이다[6]. Apriori 알고리즘에서는 빈발 K-항목집합 L_k을 구하기 위해 빈발 (K-1)-항목집합 L_{k-1}로부터 후보 K-항목집합 C_k를 구하고 C_k의 지지도를 계산하여 최소지지도 이상을 만족하는 L_k를 구하는 과정을 반복한다. Apriori에서는 SETM과 달리 후보 K-항목집합 생성시, 후보 (K+1)-항목집합의 존재 여부가 전지 단계에서 검증된다는 것이다. 따라서 후보 K-항목집합에 속하지 않는 항목집합에 대한 후보 (K+1)-항목집합의 생성 단계가 필요없다. 또한 Apriori 알고리즘에서는 apriori-gen함수를 이용한 후보 K-항목집합 생성단계와 지지도를 계산하여 빈발-K항목집합을 생성하는 단계가 서로 분리된다. 그러나 Apriori 알고리즘에서는 C_k에 속한 각 항목집합의 지지도를 계산하기 위해 전체 데이타베이스를 스캔해야 한다. 따라서 가능한 후보 항목집합들의 원소들의 크기를 줄일 수 있는 후보 항목집합을 생성하는 방법이 중요하다. Apriori 알고리즘에서는 각 단계의 빈발 항목집합을 대상으로 1-확장과 부분적 제거 방식을 제시한 후보 항목집합의 탐색 공간을 축소하는 방법을 제시하여 탐사 알고리즘의 수행 속도를 향상시켰다[6].

2.2.3. Sampling Approach

Sampling Approach는 전체 데이타베이스를 한번만 스캔을 실행하여 연관 규칙을 찾아내는 알고리즘이다. 이 알고리즘에서는 전체 트랜잭션 집합에서 랜덤하게 샘플링한 부분 트랜잭션을 이용하여 전체 데이타베이스에 존재 가능한 모든 연관 규칙들을 먼저 추정한다. 그리고 추정된 연관 규칙을 이용하여 전체 데이타베이스를 한번만 스캔하여 정확한 연관 규칙들을 만든다. 샘플링 단계에서 누락된 연관 규칙들은 전체 데이타베이스 스캔 과정에서 찾을 수 있다. 실험 결과에서 대량의 데이타베이스에서도 효율적으로 연관 규칙을 생성하였지만 너무 많은 후보 항목집합을 생성하는 문제점이 있다[7].

3. SQL을 이용한 연관 규칙 탐사 시스템

3.1. 연관 규칙 탐사 시스템의 구성

데이타마이닝에 대한 초기의 연구 대상은 새로운 연산자를 정의하거나 향상된 알고리즘을 발견하는데 관심이 집중되었다. 또한 초기의 데이타마이닝 시스템은 파일시스템을 기본으로 구성되었고, 특별한 형식의 데이타 구조나 효율적인 버퍼 관리를 위한 전략이 제안되었다. 하지만 파일시스템의 경우 대용량의 데이타 처리가 어렵고 사용자가 원하는 다양한 조건에 대한 연관 규칙 탐사가 어렵다[9]. 본 시스템에서는 MINE-RULE에서 처럼 SQL문의 조건절을 이용하여 사용자가 원하는 다양한 범위에 대한 패턴 추출이 가능하다.

대부분의 데이타마이닝 응용들은 ODBC를 통해 데이타베이스와 연동되어 입출력 부하가 크고, 데이타마이닝 응용에서 다룰 수 있는 데이타의 수가 제한적이다[10]. 하지만 본 시스템에서는 데이타베이스의 특성을 그대로 가질 수 있으며, 부하가 적은 오라클의 PL/SQL로 연관 규칙 알고리즘을 패키지로 구현하여, 데이타베이스와 데이타마이닝을 연동한 시스템을 구현하였다. 다음 그림 1은 본 논문에서 제안한 연관 규칙 탐사 시스템의 전체적인 구성도이다.

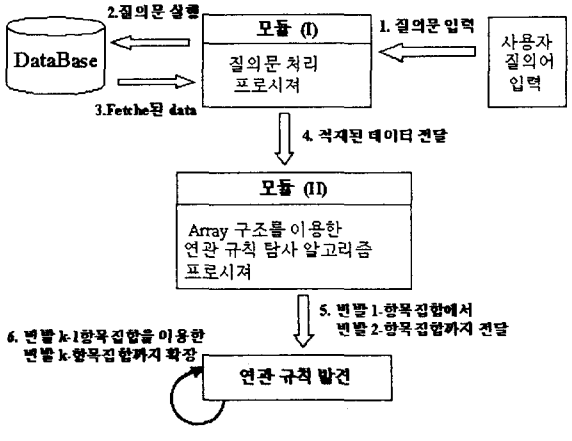


그림 1. 연관 규칙 탐사 시스템의 전체적인 구성도

제안한 시스템은 그림 1과 같이 두 개의 모듈로 구성된다. 첫 번째 모듈에서는 SQL질의를 입력 변수로 받아 조건에 맞는 대상 데이타 집합을 추출하는 기능을 담당한다. 사용자가 제시한 조건을 만족하는 질의를 동적으로 구성하여 전체 데이타베이스 중에서 관심 항목으로 제한된 범위의 데이타 항목에 대해서만 연관 규칙 탐사가 가능하다. 예를 들면 계층적 구조에서 상위 항목과 하위 항목간의 연관 규칙, 특정 속성에 대한 그룹화, 기간이나 가격별 조건을 다양하게 지정할 수 있다. 사용자가 선택한 조건에 따라 동적 SQL로 처리하였다. 동적 SQL은 본 논문에서 제시한 연관 규칙 알고리즘을 사용하기 위해 특정범위의 트랜잭션 데이타베이스를 단지 한 번만 스캔하여 해당 항목 데이타베이스의 복사본을 메모리에 적재시킨다.

두 번째 모듈은 연관 규칙 탐사를 위한 모듈로 첫번째 모듈에서 추출된 데이타와 최소 지지도를 입력 변수로 받아 메모리에 저장된 데이타베이스의 복사본에 의해 최종적인 빈발 항목집합을 추출한다. 이러한 빈발 항목집합을 이용하여 연관 규칙을 생성하여 의미 있는 패턴을 발견할 수 있다.

3.2 연관 규칙 탐사 알고리즘

본 논문에서는 Apriori와 같이 여러번 데이타베이스를 액세스해야 하는 문제점을 해결하기 위해 2차원 배열 구조를 이용하여 한번의 트랜잭션 데이타베이스 액세스로 후보 2-항목집합을 생성하고, 최소 지지도를 만족하는 빈발 2-항목집합을 추출하는 방법을 제안하였다. 이러한 방법에 의해 빈발 1-항목집합에서 빈발 K-항목집합까지 단계별로 찾을 수 있다. 즉, SQL 질의 조건을 이용하여 탐색 대상 트랜잭션 집합을 선택한 후, 전체 트랜잭션 데이타베이스를 한번만 스캔하면서 연관 규칙을 효율적으로 찾아낼 수 있다.

3.2.1 배열 구조를 이용한 후보 K-항목집합 생성

본 논문에서 제안한 연관 규칙 탐사 과정은 다음과 같은 두 단계로 구성된다.

(1) 먼저, 최소 지지도 이상의 트랜잭션들의 지지도를 갖는 빈발 항목집합을 찾는다.
(2) 빈발 항목집합들을 이용하여 연관 규칙을 찾는다.

연관 규칙 탐사 과정에서 사용하는 후보 항목집합의 구조는 배열 구조로 구성하였다. 후보 K-항목집합은 배열 구조로 구성되며, 배열 구조의 각 항목을 이용하여 후보 K-항목집합의 모든 항목 값을 저장한다. 또한 기존의 알고리즘에서 후보 (K-1)-항목집합과 후보 K-항목집합을 분리하여 생성하는 것과 달리, 배열 구조의 대각선 부분을 이용하여 후보 (K-1)-항목집합을 선정한다. 모든 후보항목은 배열 구조의 항목에 의해 표현된다.

특히, 제안된 방법에서는 Apriori와 같이 후보 K-항목집합을 생성하기 위해 빈발 (K-1)-항목집합에 대한 셀프-조인이 필요없다. 또한 최대 항목 수 M에 의해 M*M 크기를 가지는 후보 K-항목집합을 생성할 수 있다. 지지도 계산 과정에서 한 트랜잭션에 포함된 항목의 개수가 후보 K-항목집합의 개수보다 적은 것은 비교 대상에서 제외되어 탐색공간이 축소된다. 또한 하나의 트랜잭션을 처리하는 동안 후보 항목집합 대한 지지도를 동시에 계산할 수 있고, 트랜잭션에 포함되지 않은 항목은 지지도 계산에서 제외되어 실행 시간을 개선할 수 있다.

3.2.2 단계별 후보 항목집합의 생성 과정

먼저, 전체 트랜잭션 집합 D는 트랜잭션 번호 Tid와 항목들의 집합으로 구성된다. 항목들의 집합은 $I=\{item_1, item_2,...,item_m\}$, 하나의 트랜잭션에서 동시에 발생할 수 있는 항목들의 집합은 $P=\{p_1,p_2,..,p_m\}$로 정의한다. 그리고 R_n은 D에서 나타날 수 있는 n개의 item을 가지는 부분 n-항목집합이다. 다음 그림 2는 임의의 트랜잭션 집합 D에 대한 후보 항목집합이다.

TID	ITEM	Relation
100	A,C,D	{A} {A E} {E} {A C D} {A D} {C D} {D}
200	B,C,E	{B} {B C} {C} {B C E} {B E} {C E} {E}
300	A,B,C,E	{A} {A B} {B} {A B C}{A C} {B C} {C} {A B C E}{A B E}{A C E}{B C E}{A E} {B E} {C E}{E}
400	B,E	{B} {B E} {E}

그림 2. 트랜잭션 집합 D에 대한 후보 항목집합

다음 그림 3는 반복 그룹을 포함하는 초기 트랜잭션 집합 D를 관계형 데이타베이스에 저장하기 위해 정규화한 형태이다. 그림에서처럼 트랜잭션 데이타베이스의 구조는 <Tid , item>의 쌍으로 구성된다.

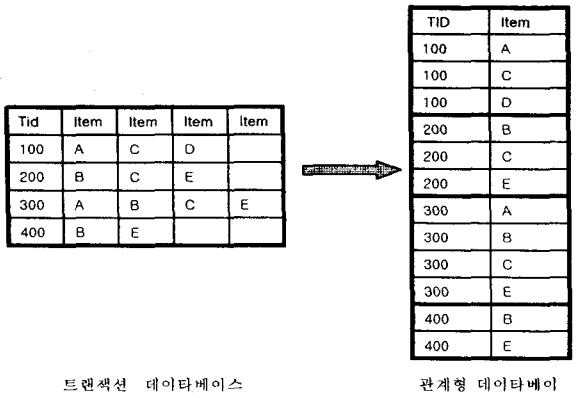


그림 3. 트랜잭션 데이타베이스

트랜잭션 데이타베이스에 대하여 단계별로 후보 항목집합을 생성 과정은 다음과 같다. 먼저, $|P_i| \geq 2$를 만족하는 트랜잭션에 대해 단계별로 스캔하면서 $R_2 \subseteq P_i$ 인 R_2를 찾아내어 후보 1-항목집합과 후보 2-항목집합을 발견하여 지지도를 계산한다. 따라서 전체 트랜잭션 데이타베이스에 대한 한 번 검색으로 후보 2-항목집합을 구할 수 있다. 후보 2-항목집합에 대해 지지도를 적용하면 빈발 2-항목을 구할 수 있다. 빈발 2-항목에 대해 후보 2-항목집합을 구하는 과정을 반복하면 후보 3-항목집합과 지지도를 구할 수 있다. 이와 같은 과정을 반복하면 전체 데이타베이스를 단계별로 한번만 스캔하여 후보 K-항목집합을 구할 수 있다.

4. 실험 및 성능 평가

4.1. 실험환경

본 논문에서 제안한 시스템은 Sun Sparc 10의 Solaris 2.6에서 오라클 DBMS 8.0.5를 기반으로 구현하였다. 연관 규칙 탐사 패키지 시스템은 오라클에서 제공하는 PL/SQL로 구현하였다. 제안한 시스템의 성능 평가를 위해 기존 Apriori 알고리즘과 SETM 알고리즘을 최소지지도에 따라 실행시간과 데이타베이스 스캔 횟수를 비교 분석하여 그에 따른 성능을 평가하였다.

4.1.1. 실험 데이타의 생성

실험 데이타는 IBM의 Quest group이 구현한 실험 데이타 생성 프로그램을 이용하여 생성하였다. 표1은 실험 데이타에 대한 세부적인 내용이다. 최대 잠재적인 빈발 항목집합의 크기는 |L|은 10,000이며, 최대 크기를 가지는 후보 2-항목집합을 구하기 위해 배열의 크기를 $5*10^5$ 크기로 설정하였다.

표 1. 실험 데이타 예

구 분	총 수
총 트랜잭션 수	25,000개
총 항목 수	1,000개
트랜잭션 당 평균 항목 수	5, 15
최대 빈발 항목집합의 수	4

4.2. 실험 결과 및 고찰

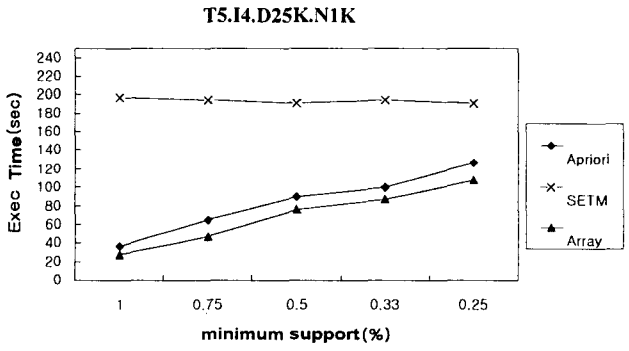
연관 규칙 알고리즘의 성능 비교는 알고리즘마다 서로 다른 실험 환경에서 수행되었기 때문에 하나의 기준으로 비교하기는 어렵다. 그러나 대부분의 연구에서 Apriori 알고리즘과의 비교를 통해 제시된 알고리즘의 성능을 비교하고 있다.

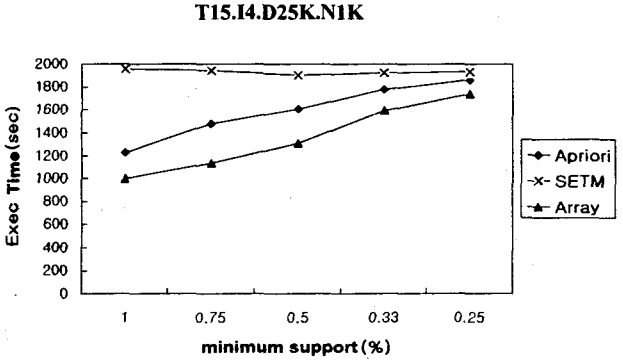
동일한 실험 데이타에 대하여 최소 지지도에 따른 실행시간을 Apriori 알고리즘, SETM 알고리즘과 제안한 알고리즘의 성능을 비교 분석하였다. 제안한 알고리즘은 후보 1-항목집합과 후보 2-항목집합을 한꺼번에 추출하기 때문에 다른 알고리즘보다 실행시간이 단축되었다.

다음 표 2는 실험 데이타와 관련된 파라미터이다.

표 2. 실험 데이타와 관련된 파라미터

파라미터	설 명
| D|	데이타베이스 D의 트랜잭션의 총 개수
| T|	트랜잭션당 항목의 평균 개수
| L|	최대 잠재적인 빈발 항목집합들의 개수
| I |	최대 잠재적인 빈발 항목집합들의 평균 크기
N	항목의 총 개 수





5. 결 론

본 논문에서는 관계형 데이타베이스를 기반으로한 연관 규칙 탐사 시스템을 제안하였다. 본 시스템은 동적 질의문을 처리하는 질의어 처리 모듈과, 질의어를 통해 추출된 대상 데이타베이스에 대한 연관 규칙을 찾는 연관 규칙 탐사 모듈로 구성된다. 질의어 처리 모듈에서는 사용자가 제시한 조건에 따라 제한된 범위의 데이타베이스를 검사할 수 있으므로, 사용자의 관심분야에 대해 연관 규칙 추출을 위한 데이타베이스의 범위를 축소할 수 있다. 또한 연관 규칙 탐사 모듈에서는 배열 구조를 활용하여 후보 항목집합을 표현한다. 이와 같은 방법에 의해 한번의 트랜잭션 데이타베이스 스캔으로 단계별 후보 항목집합의 추출과 최소 지지도를 만족하는 빈발 항목집합의 추출이 가능하다. 본 논문에서 제안한 연관 규칙 탐사 알고리즘은 관계형 데이타베이스에 기반하여 IBM의 Quest group을 통해 생성한 실험 데이타에 대해 Apriori 알고리즘과 SETM 알고리즘을 비교 분석하였을 때, 성능이 개선되었음을 확인할 수 있었다. 향후 연구 방향은 연관 규칙 탐사 시스템을 패키지화하여, GUI환경을 통해 사용자가 연관 규칙의 조건을 다양하고 편리하게 사용할 수 있도록 사용자 인터페이스를 구축하여야 할 것이다.

참 고 문 헌

[1] Ming-Syan Chen,Jiawei Han, Philip S. Yu, "Data Mining: An Overview from Database Perspective," IEEE Transaction on Knowledge and Data, Vol.8, No.6 , Engineering, pp.2~4, 1996.

[2] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. "Mining Association Rules between Sets of Items in Large Databases," In Proceedings of ACM SIGMOD Conference on Management of Data (SIGMOD'93), pp.207~216, May 1993.

[3] Rakesh Agrawal and Ramakrishnan Srikant, "Mining Sequential Patterns," In Proc. of the 11th Int'l Conference on Data Engineering, pp.3~14, Taipei,Taiwan, March 1995.

[4] Buntine, W.L., "A Theory of Learning Classification Rules," Ph.D.Dissertation, University of Technology, School of Computing Science, Sydney, 1990.

[5] Jochen Hipp, Ulrich Guntzer, Gholamreza, "Algorithms for Association Rule Mining - A General Survey and Comparison," Proceeding of ACM SIGKDD, pp.58~64, June 2000.

[6] R. Agrawal and R. Srikant," Fast Algorithms for Mining Association Rules," In Proc. of the 20th VLDB Conf., pp.487~499, Santiago, Chile, Sept. 1994.

[7] H. Toivonen, "Sampling Large Databases for Association Rules," Proceedings of the 21th International Conference on Very Large Databases", pp.134~144, 1995.

[8] Maurice A. W. Houtsma, Arun N. Swami," Set-Oriented Mining for Association Rules in Relational Databases," ICDE, pp.25~33 , 1995.

[9] Rosa Meo, Giuseppe Psaila, Stefano Ceri, "A New SQL-like Operator for Mining Association Rules," Proceedings of the 22nd VLDB Conference Mumbai(Bombay), India, pp.122~133, 1996.

[10] Rakesh Agrawal , Kyuseok Shim, "Developing Tightly-Coupled Data Mining Applications on a Relational Database System," In Proc. of the 2nd Int'l Conference on Knowledge Discover in Databases and Data Mining ,Portland , Oregon, August, pp.287~291, 1996.