韓國經營科學會誌
第27卷 第4號
2002年 12月

개선된 n-항목 연관 규칙 알고리즘 연구

황현숙* · 어윤양**

A Study on the Advanced Association Rules Algorithm of n-Items

Hyun-Suk Hwang* · Yoon-Yang Eh**

Abstract

The transaction tables of the existing association algorithms have two column attributes : It is composed of transaction identifier (Transaction_id) and an item identifier (item). In this kind of structure, as the volume of data becomes larger, the performance for the SQL query statements came applicable decreases. Therefore, we propose advanced association rules algorithm of n-items which can transact multiple items (Transaction_id, Item 1, Item 2, ···, Item n). In this structure, performance hours can be contracted more than the single item structures, because count can be computed by query of the input transaction tables. Our experimental results indicate that performance of the n items structure is up to 2 times better than the single item. As a result of this paper, the proposed algorithm can be applied to internet shopping, searching engine and etc.

Keyword : Association Rules, Database Mining, SQL query statement, Basket Data Analysis, Support and Confidence

1. 서 론

대량의 데이터베이스에서 알려지지 않은 데이터 규칙 및 패턴을 발견하는 데이터 분석 기술인 데이터마이닝에 대한 연구가 활발히 진행되고 있다 [1, 4]. 데이터마이닝에는 연관, 순차, 분류, 클러스

논문접수일 : 2001년 6월 12일 논문게재확정일 : 2002년 9월 2일
* 부경대학교 경영정보학과 시간강사
** 부경대학교 경영학부 교수

터링, 의사결정, 사례기반 추론, 퍼지로직, 신경망, 유전적 알고리즘 등의 방법이 있다[7]. 이러한 방법 중에서 연관 방법[3]은 데이터 집합을 조사하여 데이터 속성간의 관련성을 발견하는 것이다. Ronald and Khabaza(1996)는 장바구니 데이터를 분석한 결과, "슈퍼마켓에서 맥주를 구입할 때 포테이토칩도 같이 구입할 가능성이 80%로 나타나고 있다"는 것을 발견하였다. 이러한 데이터 속성에서 발견된 결과는 아이템 배치, 제품 밀접도, 판매촉진 전략 등에 대한 정보를 제공하게 된다. Marisa et al. (1996)는 연관 방법을 이용하여 건강 및 의학에 대한 정보를 발견하였다.

데이터 간의 연관성을 찾아내는 연관 알고리즘에 대한 연구가 다수 수행되었다[2, 6, 8, 12-15]. Agrawal and Srikant(1994)는 데이터 항목 집합에서 연관성이 있는 집합을 빈발 집합으로 정의하고 이를 발견하는 함수에 대해 논의하였다. Houtsma and Swami(1995)는 데이터베이스 쿼리문을 사용하여 연관 집합을 생성하는 알고리즘을 제안하였다. 초기의 이러한 알고리즘은 성능에서 효율적이지 못하였다. 이후 Agrawal *et al.*(1993)은 이전 단계의 빈발함수 집합을 이용하여 후보집합을 생성하는 Apriori 알고리즘을 제안하였다. 이 알고리즘은 트랜잭션의 수를 계산하기 위해 자신만의 데이터 구조를 필요로 하고 데이터베이스 시스템보다는 파일 시스템에 더 적합하다는 단점을 가지고 있다. 이러한 단점을 해결하기 위해 Sarawagi *et al.* (2000)는 데이터베이스별로 연관 집합을 발견을 위한 쿼리문을 제시하였다. 이때 입력 트랜잭션 테이블은 단일 항목을 가진 입력 데이터 구조(Transaction_id, Item)를 사용하였다. 단일 항목 구조는 한 개의 레코드에 한 개의 항목을 저장하고 있으므로 레코드의 수가 많아지면 알고리즘의 수행 속도가 떨어지게 된다.

그래서 본 논문에서는 입력 데이터 구조를 다중 항목 구조로 확장하여 이에 적합한 연관 규칙 알고리즘을 제시하여 단일 항목 구조와 성능을 비교한다. 본 논문의 구성은 제 2장에서 연관 규칙의 개념과 기존 연관 알고리즘에 대해서 설명하고 제 3장에서는 다중 항목 구조에서의 연관 알고리즘을 제시한다. 그리고 제 4장에서는 단일 항목과 다중 항목 구조에서 쿼리문의 수행 속도를 측정하여 비교하고 마지막으로 제 5장에서는 연구의 결과와 향후 연구과제를 제시한다.

2. 기존 연관 알고리즘

연관 규칙은 데이터베이스의 트랜잭션에서 항목간에 발생하는 규칙을 표현하는 것으로 Agrawal and Srikant(1994)에 의해 처음 소개되었다. 연관 규칙은 어떤 사건이 발생될 때 그 다음 사건의 관련성을 의미하는 것으로 $X \Rightarrow Y$ 규칙의 형태로 표현된다. $X \Rightarrow Y$의 규칙은 데이터베이스의 트랜잭션 중 X라는 항목 집합을 포함하는 트랜잭션은 Y라는 항목집합도 함께 포함하는 경향이 있음을 의미한다.

연관 규칙에서는 지지도와 신뢰도라는 척도로 그 타당성이 판단된다[2, 13]. 지지도는 전체 트랜잭션에 대해 트랜잭션 항목 집합이 차지하는 비율을 의미하고 신뢰도는 조건부 트랜잭션 항목 집합에 대해 규칙에 포함되는 모든 항목 집합이 차지하는 비율을 의미한다.

Agrawal and Srikant(1994)는 장바구니 데이터를 사용하여 고객이 구매한 상품간에 연관성이 있는 집합을 발견하는 AIS(Association Item Sets) 알고리즘을 제시하였다. AIS 알고리즘에서는 전체 데이터베이스를 검색하여 최소한의 트랜잭션 개수를 가지는 후보 항목 집합을 발견하였다. 또한, 연관 규칙 생성에 대한 기준으로 사용하고 있는 최소 지지도와 신뢰도 설정에 대해서 논의하였다.

Houstma and Swaii(1995)은 AIS 알고리즘을 기반으로 데이터베이스 쿼리문을 사용하여 연관 집합을 발견하는 SETM(SETs Mining) 알고리즘을 제시하였다. 이러한 AIS와 SETM 알고리즘은 후보 집합을 생성할 때 데이터베이스를 여러 번 접근하여 생성하기 때문에 메모리 관리와 성능에서

효율적이지 못하였다.

Agrawal *et al.*(1993)은 이전 단계의 빈발 함수 집합을 이용하여 후보집합을 생성하는 Apriori 알고리즘을 제안하였으며, AIS와 SETM 알고리즘보다 수행시간이 빠르게 나타났다.

수행 속도가 빠르다고 검증된 Apriori 알고리즘의 연관 규칙 생성 과정은 크게 두 단계로 나눌 수 있다.

첫째, 최소 지지도 이상을 갖는 빈발 항목 집합(large item sets)을 발견하는 단계이다. 빈발 항목 집합(L_k)은 이전 단계(k-1)의 빈발 항목 집합에서 k개의 가능한 항목 집합을 생성하여 L_k의 부분 집합이 아닌 경우를 제거하여 후보 항목 집합(C_k)으로 한다. 이때, 생성한 후보항목 집합에서 최소 지지도 이상을 가지는 집합을 빈발 항목 집합으로 한다.

두 번째는 발견한 빈발 항목 집합의 모든 부분집합을 생성하여 최소 신뢰도 이상인 규칙을 발견하는 단계이다.

Apriori 알고리즘은 지지도를 효율적으로 계산하기 위해 해쉬-트리(hash-tree) 데이터 구조를 필요로 하고 제거 과정과 트랜잭션 개수를 계산할 때 쿼리문을 사용하지 않는다.

Sarawagi *et al.*(2000)는 후보 집합 생성과 트랜잭션 개수를 계산하기 위해 Apriori 알고리즘을 기반으로 데이터베이스 시스템과 통합한 알고리즘을 제시하였다. Sarawagi *et al.*(2000)이 제시한 알고리즘의 입력 데이터 구조는 트랜잭션을 구분하는 트랜잭션 번호(Tid)와 데이터 항목을 나타내는 한 개의 항목(Item) 필드로 구성되어 있다. 이러한 단일 항목을 가지는 입력 데이터 구조에서는 한 개의 레코드에 한 개의 항목이 저장되어 있으므로 레코드의 수가 많아지면 알고리즘의 수행 속도가 떨어지게 된다. 또한, 트랜잭션의 개수를 계산하기 위해 k개의 트랜잭션 데이터베이스와 조인을 수행하기 때문에 수행 속도가 느려진다. 또한, 이러한 단일 항목 데이터 구조는 장바구니 데이터를 분석할 때 효율적인 구조로서 활용 범위가 좁다는 제한점이 있다.

3. 다중 항목 연관 알고리즘 설계

3.1 다중 항목 데이터 구조

본 장에서는 기존 연관 알고리즘의 입력 데이터 구조인 단일 항목 구조를 다중 항목 구조로 확장하여 연관 집합을 발견하는 알고리즘을 제시한다.

다중 항목 데이터는 다수의 항목을 가지고 있기 때문에 항목간에 동일성을 가진 속성인지 아닌지를 분류할 필요가 있다. 따라서 각 항목별 속성을 두 가지 유형으로 가정한다. 첫째, 전체 항목이 동일한 속성을 가질 경우이고 둘째, 항목별로 속성이 동일할 경우로 가정한다. <표 2>는 가정한 유형을 테이블로 표시한 것이다. 테이블 (a)는 첫 번째 유형으로 장바구니 데이터를 분석하기 위한 데이터 구조로 트랜잭션별로 구입한 품목을 각 항목에 저장하고 있다. 테이블 (b)는 두 번째 유형으로 상품 속성별로 각 데이터를 저장하고 있다. $Item_1$은 회사에 대한 속성이고 $Item_2$는 부피에 대한 속성이고 $Item_3$은 가격에 대한 속성으로 각 항목별로 동일한 속성을 가지고 있다.

〈표 2〉 다중 항목 데이터 구조

(a) 항목 전체가 동일한 속성일 경우

T_id	$Item_1$	$Item_2$	$Item_3$
1	사 과	콜 라	빵
2	콜 라	빵	오징어
3	사 과	콜 라	

(b) 항목별로 동일한 속성일 경우

T_id	$Item_1$	$Item_2$	$Item_3$
1	삼 성	550리터	90만원
2	대 우	350리터	70만원
3	LG	620리터	120만원

3.2 연관 마이닝 알고리즘 제시

다중 항목 데이터 구조의 입력 테이블은 트랜잭션 번호와 n개의 항목으로 구성되어 있다. 이러한 구조에서 연관 알고리즘을 설계하는데 필요한 기호를 <표 3>에 나타내고 있다.

〈표 3〉 기호 정의

기 호	설 명
k	각 단계를 나타내는 첨자
TC_k	단계 k에 대한 임시 후보 항목 집합을 가지는 것으로 k개의 항목 필드와 k개 집합에 대한 카운트 필드로 구성
C_k	단계 k에 대한 후보 항목 집합을 가지는 것으로 k개의 항목 필드와 k개 집합에 대한 카운트 필드로 구성
F_k	단계 k의 후보 항목 집합에서 사용자가 정의한 최소한의 지지도 이상인 빈발 항목 집합을 가지는 것으로 k개의 항목 필드와 k개 집합에 대한 카운트 필드로 구성
$RST_k(j)$	k단계에서 생성 가능한 규칙의 집합을 나타내는 테이블의 j번째 레코드
R_k	최소 지지도와 신뢰도 이상을 가지는 규칙 집합을 가지는 것으로 k개의 항목 필드와 k개 집합에 대한 지지도와 신뢰도 필드로 구성
Min_k (support)	단계별 최소 지지도
Min_k (confidence)	단계별 최소 신뢰도

[그림 1]은 다중 항목 구조에서 연관 알고리즘의 전체 구성이다. 제시한 알고리즘은 크게 3개의 함수로 구성되어 있다. 첫 번째는 후보 항목 집합을 생성하는 Candidate_Item_Generation( ) 함수로 이의 결과는 C_k 테이블에 저장된다. 두 번째는 후보 항목 집합인 C_k에 대해 최소 지지도 이상을 가지는 빈발 항목 집합을 생성하는 Frequent_Item_Generation( ) 함수로 이의 결과는 F_k 테이블에 저장된다. 마지막으로 생성된 빈발 항목 집합(F_k)의 각 단계별로 최소 신뢰도 이상을 가지는 연관 규칙 집합을 생성하는 Rule_Set_Generation( ) 함수로 이의 결과는 R_k테이블에 저장된다.

```
Algorithm Multi_Item_Association
n : 전체항목개수, k : 각 단계의 색인
trans_tbl : 트랜잭션 테이블,
tr_count : 총 트랜잭션 개수
c.count : 각 후보항목집합의 트랜잭션 개수
ikind : 동일 필드의 속성(1 : 동일한 속성)
for (k = 1 ; n ; k++) do begin
 Ck = Candidate_Item_Generation(trans_tbl, k, n,
      ikind) ; // k개 후보 항목 집합 생성
 Fk = Frequent_Item_Generation(Ck, k, n) ;
      // k개의 빈발 항목 집합 생성
end
for (k = 2 ; Fk ≠ Ø ; k++) do begin
 Rk = Rule_Set_Generation(Fk, tr_count) ;
      // 지지도를 상속받은 신뢰도 규칙 집합 생성
end
```

[그림 1] 다중 항목 연관 알고리즘

3.2.1 후보 항목 생성 알고리즘

[그림 2]는 k 단계의 후보 항목 집합을 생성하는 알고리즘이다. 입력 데이터 테이블인 트랜잭션 테이블(Trans_tbl)은 n개의 항목 필드를 가지고 있다. 여기서 k 단계의 후보 항목 집합은 트랜잭션 테이블의 n개 항목 집합에서 가능한 k개의 모든 항목 집합이다. k 단계의 후보 항목 집합을 생성하기 위해서는 우선, n개의 항목 필드에서 가능한 k개의 항목 필드 유형을 찾아야 한다. 예를 들어, 전체 문항수(n)가 4이고 항목(k)이 2일 때, 항목 필드의 유형은 (I_1, I_2), (I_1, I_3), (I_1, I_4), (I_2, I_3), (I_2, I_4), (I_3, I_4)로 모두 6가지로 표시된다. 이러한 각 유형에 대해 쿼리문으로 해당 후보 항목 집합을 생성하게 된다. 이러한 항목 필드의 유형의 개수는 전체 문항인 n개의 집합에서 가능한 k개의 조합의 수이다. 따라서 k 단계의 항목 필드 유형의 개수는 ${}_nC_k$ 개이다.

다중 항목 입력 데이터의 속성은 동일 항목 필드에서 동일한 데이터 속성이 있는 경우와 전체 항목이 동일한 속성일 경우로 분류하였다. 동일 필드에서 동일한 속성이 있는 경우(ikind = 1)는 k개의 항목 집합에 대해 한 번의 그룹화로써 count(*) 함수를 이용하여 레코드의 개수를 계산한다. 전체

항목이 동일한 경우의 카운트는 여러 필드에 분산되어있는 데이터의 카운트를 구해야 하므로 두 번의 그룹화로 계산된다. 두 번째 수행하는 그룹화 연산에서의 레코드 개수는 첫 번째 그룹화에서 생성한 k개의 집합(TC_k)에서 동일한 집합에 대해 sum(count) 함수의 누적으로 계산된다.

```
Algorithm Candidate_Item_Generation(Trans_tbl, k,
    n, ikind)
if (ikind = 1) then C = Ck
else C = TCk
endif
for (i1 = 1 ; n - (k - 1) ; i1++) do begin
 for (i2 = i1 + 1 ; n - (k - 2) ; i2++) do begin
  for (i3 = i2 + 1 ; n - (k - 3) ; i3++) do begin
                    .....
  for (ik = ik-1 + 1 ; n ; ik++) do begin
   insert into C
    select Itemi1, Itemi2, Itemi3, ···, Itemik,
    count(*)
     from Trans_tbl
     group by Itemi1, Itemi2, Itemi3, ···, Itemik
     order by Itemi1, Itemi2, Itemi3, ···, Itemik
   end
                    .....
  end
 end
end
if (ikind ≠ 1) then
  insert into Ck
   select Item1, Item2, Item3, ···, Itemk, sum(count)
        from TCk
        group by Item1, Item2, Item3, ···, Itemk
        order by Item1, Item2, Item3, ···, Itemk
endif
```

[그림 2] 후보 항목 생성 알고리즘

3.2.2 빈발 항목 생성 알고리즘

k 단계의 빈발 항목 집합은 k 단계의 후보 항목 집합 중에서 최소 지지도 이상이고 이전 단계의 빈발 항목 집합인 F_{k-1}의 부분 집합인 것으로 한다. 이를 위해 쿼리문은 후보 항목 집합인 C_k와 F_{k-1}과 k번 조인을 수행하여 k-1 단계의 빈발 항목 집합에 속하는 부분 집합을 추출하는 조건을 포함하여야 한다. 쿼리문의 조건에서는 k개에서 k-1개의 항목의 유형에 대해 F_{k-1}의 항목과 같은 지를 비교한다. 즉, k개 유형의 각 후보 항목 집합을 F_{k-1}의 k-1개 유형과 같은지를 비교한다.

[그림 3]은 이러한 빈발 항목 집합을 생성하는 알고리즘이다. k가 1일 때, 빈발 항목 집합은 항목 집합의 카운트가 최소 지지도 이상인 조건을 만족하는 쿼리문에서 생성된다. k가 2이상일 때, 빈발 항목 집합은 이전 단계의 빈발 항목 집합인 F_{k-1}의 부분 집합을 추출하는 조건과 항목 집합의 카운트가 최소 지지도 이상인 조건을 만족하는 쿼리문에서 생성된다.

```
Algorithm Frequent_Item_Generation(Ck, k, n)
if (k = 1) then
 insert into Fk
  select Item1, Item2, ···, Itemk, cnt from Ck
       where cnt ≥ Mink (support)
else
 insert into Fk
  select C.Item1, C.Item2, ···, C.Itemk, C.cnt
  from Ck as C, Fk-1 I1, ···, Fk-1 Ik
  where C.Item1 = I1.Item1 and C.Item2 = I2.Item2
                and ··· and
                    C.Itemk-1 = Ik.Itemk-1 and
    C.Item1 = I1.Item1 and C.Item2 = I2.Item2 and
                ··· and
                    C.Itemk = Ik.Itemk-1 and
                    .....
    C.Item1 = I1.Item1 and C.Item2 = I1.Item2 and
                ··· and
        C.Itemk-1 = Ik.Itemk-2 and C.Itemk = Ik.
                                    Itemk-1 and
    C.Item1 = I1.Item1 and C.Item3 = I1.Item2 and
                ··· and
                    C.Itemk = Ik.Itemk 1 and
    C.Item2 = I2.Item1 and C.Item3 = I1.Item2 and
                ··· and
                    C.Itemk = Ik.Itemk 1 and
    C.cnt ≥ Mink (support)
end if
```

[그림 3] 빈발 항목 집합 생성 알고리즘

3.2.3 연관 규칙 생성 알고리즘

연관 규칙 집합은 각 단계의 빈발 항목 집합에서

각 단계의 최소 신뢰도 이상을 가지는 집합으로 한다. 이러한 연관 규칙 집합을 발견하기 위해서는 단계별 빈발 항목 집합 대해 규칙의 조건부와 결과의 항목 순서를 저장하고 있는 2에서 n 단계까지의 규칙 집합 테이블이 필요하다.

k 단계에서 생성 가능한 모든 규칙 집합을 저장하고 있는 테이블을 RST_k라고 정의하면 [그림 4]는 3단계와 4단계에서 생성되는 모든 규칙의 집합을 저장하고 있는 RST_3, RST_4 테이블이다. F_3의 조건부와 결과의 항목 개수의 유형은 (1, 2)과 (2, 2) 형태이고 이때, RST_3은 전체 6개의 레코드를 가진다. F4의 조건부와 결과의 항목 개수의 유형은 (1, 3), (2, 2), (3, 1) 형태이다. 이 유형 중에서 (3, 1)의 유형은 (1, 3)의 역의 형태이므로 RST_4 테이블은 (1, 3)과 (2, 2) 유형에 대한 항목 집합만 구성하면 된다.

select * from rst3 ;

item 1	item 2	item 3
item 1	item 2	item 3
item 2	item 1	item 3
item 3	item 1	item 2
item 1	item 2	item 3
item 1	item 3	item 2
item 2	item 3	item 1

(6 row(s) affected)

select * from rst4 ;

item 1	item 2	item 3	item 4
item 1	item 2	item 3	item 4
item 2	item 1	item 3	item 4
item 3	item 1	item 2	item 4
item 4	item 1	item 2	item 3
item 1	item 2	item 3	item 4
item 1	item 3	item 2	item 4
item 1	item 4	item 2	item 3
item 2	item 3	item 1	item 4
item 2	item 4	item 1	item 3
item 3	item 4	item 1	item 2

(10 row(s) affected)

[그림 4] F_3, F_4의 규칙 집합 테이블

[그림 5]는 규칙 항목의 순서를 가진 RST_k 테이블과 빈발 항목 집합 테이블인 F_k를 이용하여 최소 신뢰도 이상을 가지는 신뢰도 규칙 집합 테이블인 R_k를 생성하는 알고리즘이다. 색인 i는 단계 k의 조건부 항목 집합의 개수를 나타내고 j는 조건부 i개 항목에서 생성되는 항목 유형의 개수를 나타내는 색인이다. jj는 레코드 순서를 누적하기 위한 색인으로 사용된다.

```
Algorithm Rule_Set_Generation (Fk, tr_count)
for (i = 1 ; k / 2 ; i++) do begin
 for (j = 1 ; kCi ; j++) do begin
  jj = jj + j ;
  item1 = RSTi (jj).rs1 ; item2 = RSTi (jj).rs2 ; … ;
  itemk = RSTk (jj).rsk ;
  insert into Rk
    select item1, item2, …, itemi as "itemi ⇒ ",
     itemi+1, …, itemk, cnt / tr_count as support,
     cnt / (select cnt from fi
          where fi.item1 = fk.item1, …,
               fi.itemi=fk.itemi,)
      as confidence from fk
     where confidence ≥ MINk (confidence)
 if (i ≠ k / 2) then
  insert into Rk
    select itemi+1, …, itemk as "itemk ⇒ " item1,
     item2, …, itemi, cnt / tr_count as support,
     cnt / (select cnt from fk-i where fk-i.item1 =
     fk. itemi+1, …, fk-i.itemk-i = fi.itemk) as
     confidence from fk
      where confidence ≥ MINk (confidence)
    endif
  end
end
```

[그림 5] 연관 규칙 생성 알고리즘

규칙 집합인 R_k의 지지도는 빈발 항목 집합 F_k 이미 상속을 받고 있기 때문에 R_k는 최소 신뢰도만을 고려하여 생성하면 된다. 신뢰도 규칙 집합은 RST_k 테이블에서 항목의 순서에 대한 값을 추출한 후 F_k 테이블을 이용하여 신뢰도를 계산하여

최소 신뢰도 이상인 집합을 가지게 된다. 알고리즘에서 지지도(support)는 F_k의 카운트를 전체 트랜잭션 개수로 나눈 값이고 신뢰도(confidence)는 F_k의 카운트를 조건부 항목 개수까지인 F_i의 카운트로 나눈 값이다.

3.3 알고리즘 수행 분석

3.3.1 분석 데이터

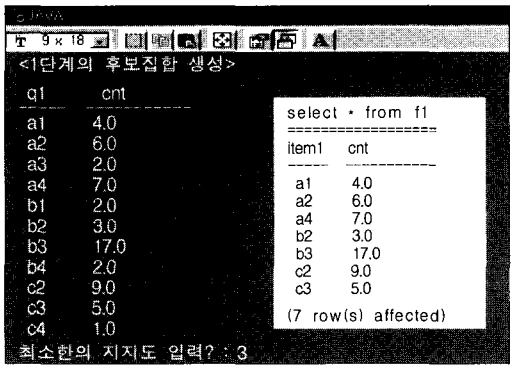
제시한 알고리즘의 수행 결과를 분석하기 위해 [그림 6]에 있는 분석 데이터를 가지고 알고리즘 수행하였다. [그림 6]의 tid 필드는 입력 트랜잭션 테이블의 트랜잭션 번호를 의미하고 q1, q2, q3는 필드 항목을 의미한다. 각 필드는 속성이 같은 유형인 것으로 가정하였다. 그래서 각 레코드의 q1 필드의 데이터 값은 a, q2는 b, q3는 c로 시작하도록 하였다. 데이터가 비어있는 항목은 어떤 값도 지정하지 않았음을 의미한다.

tid	q_1	q_2	q_3
1	a1	b2	c2
2		b3	c2
3	a4	b3	
4	a2	b3	c2
5	a2	b3	c3
6	a1	b2	c2
7		b3	c2
8	a4	b3	
9	a3	b1	c1
10	a2	b1	c1
11	a2	b3	c2
12	a4	b3	
13		b3	c2
14	a2	b2	c3
15	a2		
16	a4	b3	c3
17	a1	b3	c3
18		b3	
19	a2	b3	c2
20		b3	c2
21	a4	b3	
22	a4	b3	c3
23		b3	
24	a1	b4	
25	a3	b4	c4

[그림 6] 분석 데이터 입력 테이블

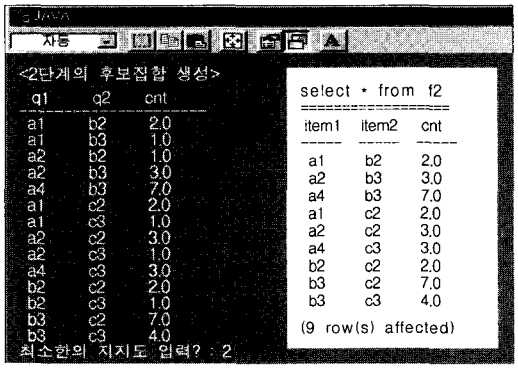
3.3.2 단계별 후보 및 빈발 항목 집합 생성

분석 데이터는 3개의 필드를 가지고 있다. 그래서 후보 및 빈발 집합은 항목이 1, 2, 3개를 가지는 항목 집합으로 구성된다. [그림 7]은 1번째 단계의 후보 항목 집합인 C_1을 생성하는 화면이다. C_1에서 최소한의 카운트를 3으로 입력하였을 경우, 빈발 항목 집합 F_1을 나타내고 있다. 이때, 지지도의 최소 임계값은 휴리스틱한 방법으로 많이 결정하고 있는데, 임계값을 효율적으로 설정하는 연구가 필요하다.



[그림 7] 1단계 후보 및 빈발 항목 집합

[그림 8]은 2번째 단계의 후보 항목 집합 C_2 생성와 F_2를 나타내고 있다. cnt는 동일한 검색 유형에 대한 레코드의 수를 의미한다. 최소한의 카운트를 2로 하여 빈발집합인 F_2를 생성하였다.



[그림 8] 2단계 후보 및 빈발 항목 집합

[그림 9]는 3번째 단계의 항목 집합인 C_3과 F_3를 나타낸 것이다. F_3을 생성할 때 최소한의 카운트를 2로 하였다.

select * from f3 ;

item1	item2	item3	cnt
a1	b2	c2	2.0
a2	b3	c2	3.0
a4	b3	c3	3.0

(3 row(s) affected)

[그림 9] 3단계 후보 및 빈발 항목 집합

3.3.3 규칙집합 생성

빈발 집합을 가지고 각 단계의 규칙 집합을 생성하였다. [그림 10]은 F_2 집합에서 생성할 수 있는 모든 가능한 규칙 집합이다. F_2의 최소 지지도 0.08과 최소 신뢰도 0.7로 가정할 경우, 이를 만족하는 연관 규칙 집합은 {a4, b3}, {b3, c2}, {b3, c3}이다. 이때, 신뢰도의 최소 임계값은 휴리스틱한 방법으로 많이 결정하고 있지만 효율적인 설정에 대한 연구가 필요하다.

item1=>	item2	cnt	support	confidence
a1	b2	2.0	0.08	0.5
a1	c2	2.0	0.08	0.5
a2	b3	3.0	0.12	0.5
a2	c2	3.0	0.12	0.5
a4	b3	7.0	0.28	1.0
a4	c3	3.0	0.12	0.43
b2	c2	2.0	0.08	0.67
b3	c2	7.0	0.28	0.41
b3	c3	4.0	0.16	0.24
b2	a1	2.0	0.08	0.67
b3	a2	3.0	0.12	0.18
b3	a4	7.0	0.28	0.40
c2	a1	2.0	0.08	0.22
c2	a2	3.0	0.12	0.33
c2	b2	2.0	0.08	0.22
c2	b3	7.0	0.28	0.78
c3	a4	3.0	0.12	0.60
c3	b3	4.0	0.16	0.80

(18 row(s) affected)

[그림 10] F_2 집합에서 생성된 모든 규칙 집합(R_2)

[그림 11]은 F_3 집합에서 생성할 수 있는 모든 가능한 규칙 집합이다. F_3의 최소 지지도 0.08와 최소 신뢰도를 0.8이상으로 할 경우 이를 만족하는 연관 규칙 집합은 {a1, b2, c2}, {a2, b3, c2}, {a4, b3, c3}이다. 지지도가 낮은 것은 고객이 선택할 수 있는 검색 유형은 모두 124가지 유형인데 비해 분석 데이터는 25개의 레코드를 가지고 있기 때문이다.

item1 =>	item2	item3	cnt	support	confidence
a1	b2	c2	2.0	0.08	0.5
a2	b3	c2	3.0	0.12	0.5
a4	b3	c3	3.0	0.12	0.43
b2	a1	c2	2.0	0.08	0.67
b3	a2	c2	3.0	0.12	0.18
b3	a4	c3	3.0	0.12	0.18
c2	a1	b2	2.0	0.08	0.22
c2	a2	b3	3.0	0.12	0.33
c3	a4	b3	3.0	0.12	0.60

(9 row(s) affected)

item2	item3 =>	item1	cnt	support	confidence
a1	b2	c2	2.0	0.08	1.0
a1	c2	b2	2.0	0.08	1.0
a2	b3	c2	3.0	0.12	1.0
a2	c2	b3	3.0	0.12	1.0
a4	b3	c3	3.0	0.12	0.43
a4	c3	b3	3.0	0.12	1.0
b2	c2	a1	2.0	0.08	1.0
b3	c2	a2	3.0	0.12	0.43
b3	c3	a4	3.0	0.12	0.75

(9 row(s) affected)

[그림 11] F_3 집합에서 생성된 모든 규칙 집합(R_2)

Ⅳ. 성능 분석

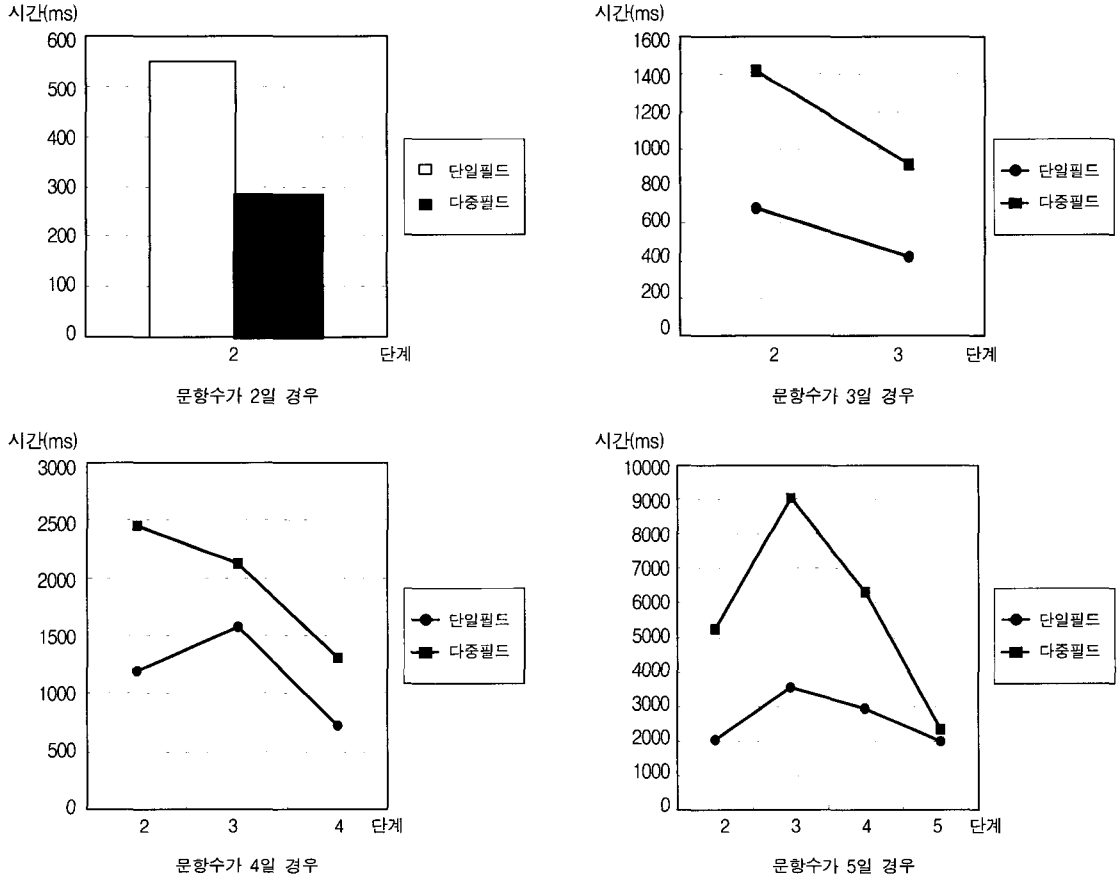
제시한 다중 항목 필드 구조에서의 알고리즘이 단일 항목 필드 구조보다 쿼리문의 수행속도가 어느 정도 효율성이 있는지를 실험하였다. 쿼리문의 수행시간은 테이블의 조인의 수에 영향을 받기 때문에 우선, 단일 항목과 다중 항목 구조에서의 항목 집합 생성때 수행한 테이블 조인을 비교하였다. <표 4>는 두 구조에서 트랜잭션 테이블 구조와 후보 및 빈발 항목 집합 생성에 필요한 테이블과 조인 형태를 나타내고 있다. 여기에서 단일 항목 구조는 Sarawagi[12]가 제시한 알고리즘을 사용하였다.

〈표 4〉 항목 집합 생성 과정

구분 / 항목	단일 항목	다중 항목
트랜잭션 테이블구조	T = (T_id, Item)	T = (T_id, Item$_1$, …, Item$_n$)
후보 항목 집합	$F_{k-1} \bowtie_1 F_{k-1} \bowtie_2 \cdots \bowtie_k F_{k-1}$	$\sigma(T)$ (카운트 계산)
빈발 항목 집합	$C_k \bowtie T \bowtie_1 T \bowtie_2 \cdots \bowtie_k T$ (카운트 계산)	$\sigma(C_k) \bowtie F_{k-1} \bowtie_1 F_{k-1} \cdots \bowtie_k F_{k-1}$

단일 항목 구조에서 후보 항목 집합은 이전 단계 F_{k-1}의 조인으로 생성되고 빈발 항목 집합은 C_k와 k개의 트랜잭션 테이블과의 조인으로 생성된다. 다중 항목 구조에서 후보 항목 집합은 트랜잭션 테이블의 조회로 생성되고 빈발 항목 집합은 C_k와 k개의 F_{k-1} 테이블의 조인으로 생성된다. 이러한 구조에서 보면 단일 항목 구조에서는 레코드 개수가 많은 트랜잭션 테이블과의 k번 조인으로 실행 시간이 많이 걸리게 된다. 하지만 다중 항목 구조에서는 트랜잭션 테이블의 조회에서 후보 항목을 생성함으로 수행 시간이 단축된다.

다음으로 단일 항목과 다중 항목 구조의 쿼리문의 성능을 비교 평가하기 위해 k 단계에 따라 후보 항목 집합과 빈발 항목 집합을 생성하는 쿼리문의 수행 시간을 측정하였다. 입력 트랜잭션 테이블은 두 구조 모두 동일한 데이터를 사용하였고 문항의 수에 따라 10,000건의 레코드를 가지게 하였다. 문항수가 2개일 경우는 단일 항목 데이터 구조에서

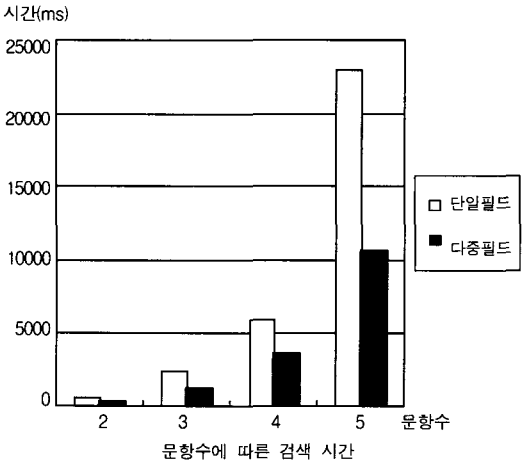


[그림 12] 문항수에 따른 단계별 수행 속도 비교

는 항목 필드의 수가 1개이므로 전체 20,000건의 레코드를 가지고 다중 항목 데이터 구조에서는 항목 필드가 2개이므로 전체 10,000건의 레코드를 가지게 된다. 입력 트랜잭션 데이터를 구성한 후 각 구조에 대해 후보 및 빈발 항목 집합을 생성하는 쿼리문을 수행시킨 후 수행 시간을 측정하였다.

[그림 12]는 단일 항목과 다중 항목 필드의 항목의 수에 따라 단계별로 수행속도의 변화를 나타내는 차트이다. 차트를 보면 다중 항목 구조가 단일 항목 필드 구조보다 수행속도가 빠르게 나타남을 알 수 있다. 초기 단계에서는 수행속도의 차이가 조금씩 나다가 마지막 단계로 갈수록 후보 항목 집합 레코드의 수가 줄기 때문에 속도의 차이는 줄어들었다. 문항수가 5일 경우 k값이 3인 중앙값을 가질 때 실행 속도의 차이가 가장 크게 나타남을 볼 수 있다.

[그림 13]은 문항 수에 따른 전체 수행 속도를 나타내는 차트이다. 문항 수에 따른 수행 속도는 제시한 다중 항목 구조가 약 2배정도 빠르게 나타났다. 이러한 결과는 단일 필드 구조에서는 레코드 개수가 많은 트랜잭션 테이블과의 k번 조인으로 카운트를 구하기 때문에 실행 시간이 많이 걸리게 되고 다중 항목 필드 구조에서는 트랜잭션 테이블의 조회에서 카운트가 계산되기 때문에 수행 시간이 단축되는 것으로 분석될 수 있다.



[그림 13] 문항수에 따른 전체 수행 속도

V. 결 론

연관 집합 발견에서 기본으로 사용하고 있는 Apriori 알고리즘은 이전 단계의 빈발함수를 이용하여 후보집합을 생성하였다. Sarawagi는 Apriori 알고리즘을 기반으로 SQL 기반의 연관 알고리즘을 제안하였다. 이러한 기존 알고리즘의 입력 트랜잭션 테이블은 단일 항목을 가진 입력 데이터 구조를 사용하였다. 이러한 구조는 레코드 수가 많아지면 알고리즘의 수행 속도가 떨어지는 단점이 있고 장바구니 분석에서 주로 활용할 수 있다는 제한점이 있다.

본 논문에서는 이를 확장하여 다중 항목 데이터 구조에서의 연관 알고리즘을 제시하였다. 다중 항목 구조에서 후보 항목 집합은 입력 트랜잭션 테이블의 조회로 생성되고 이때 레코드의 개수가 계산되기 때문에 단일 항목 구조보다 수행시간을 단축시킬 수 있었다. 그리고 단일 항목과 다중 항목 구조에서 수행되는 쿼리문의 수행시간을 측정하였다. 그 결과, 다중 항목 구조에서 쿼리문의 수행 속도가 빠르게 나타났다. 그러므로 다중 항목 구조에서의 데이터 연관 집합을 생성하는 것이 효율적임을 발견하였다.

연관 알고리즘은 마케팅, 생산, 금융, 건강과 의학 등 여러 분야에서 활용될 수 있다. 본 논문에서 제시한 다중 항목 알고리즘으로는 상품, 고객, 환자 등이 가지는 여러 가지의 속성을 이용하여 분석할 수 있는 분야에 적합하다. 예를 들어, 인터넷 쇼핑몰에서 고객이 구매한 상품 데이터를 사용하여 제시한 알고리즘으로 어떤 고객이 어떤 특징을 가지는 상품을 선호하는지에 대한 정보를 개인별 광고로서 제공한다면 판매 촉진에 도움이 될 것이다. 또한, 금융 분야에서 고객과 신용카드 사용에 대한 연관성을 발견하여 사기 유형에 대한 정보를 분석할 수 있다.

향후 다음과 같은 연구가 추가되어야 한다. 첫째, 연관 집합 생성의 척도인 신뢰도 및 지지도의 최소 임계값 결정을 어떻게 할것인가에 대한 연구

가 필요하다. 둘째, 제시한 연관 알고리즘을 여러 분야에 적용하여 마이닝 작업의 효율성을 검토할 필요가 있다.

참 고 문 헌

[1] Adriaans, P. and D. Zantiage, *Data Mining*, Addison-Wesley, 1996.

[2] Agrawal, R. and R. Srikant, "Fast Algorithms for Mining Association Rules," *Proceeding of the 20th VLDB Conference*, 1994.

[3] Agrawal, R., T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," *Proceeding of ACM SIGMOD Conference on Management of Data*, 1993, pp.207-216.

[4] Berson, A. and S. Smith, *Data Warehousing, Data Mining and OLAP*, McGraw-Hill, 1997.

[5] Brin, S., R. Motwani, J.D. Ullman and S. Tsur, "Dynamic Itemset Counting and Implication rules for market basket data," *Proceeding of ACM SIGMOD Conference on Management of Data*, 1997, pp.255-264.

[6] Craig S., B. Sergey and M. Rajeev, "Beyond Market Baskets : Generalizing Association Rules to Dependence Rules," *Data Mining and Knowledge Discovery*, Vol.2, No.1(1998), pp.39-68.

[7] Deker, K.M. and F. Sergio, "Technical Overview : A Report on Data Mining," CSCS-ETH, *Swiss Scientific Computing Center*, 1995.

[8] Houtsma, M. and A. Swami, "Set-oriented mining of association rules," *Proceedings of 11th International Conference on Data Engineering*, 1995.

[9] Marisa, I., S. Viveros, J.P. Nearhos and M.J. Rothman, "Applying Data Mining Techniques to a Health Insurance Information System," *Proceeding of the 22nd VLDB Conference*, 1996.

[10] Meo, R., G. Psaila and S. Ceri, "An Extension to SQL for Mining Association Rules," *Data Mining and Knowledge Discovery*, Vol.2, No.2(1998), pp.195-224.

[11] Ronald, J.B. and T. Khabaza, "Mining Business Databases," *Communications of the ACM*, Vol.39, No.11(1996).

[12] Sarawagi, S., S. Thomas and R. Agrawal, "Integrating Association Rule Mining with Relational Database Systems : Alternatives and Implications," *Data Mining and Knowledge Discovery*, Vol.4, No.2(2000), pp.89-125.

[13] Srikant, R., T. Imielinski and A. Swami, "Mining Associations between Sets of Items in Massive Databases," *Proceedings of ACM SIGMOD*, 1993.

[14] Srikant, R., R. Agrawal, "Mining Quantitative Association Rules in Large Relational Tables," *Proceedings of ACM SIGMOD*, 1996.

[15] Thomas, S. and S. Sarawagi, "Mining Generalized Association Rules and Sequential Patterns Using SQL Queries," *Processing of the 4th Int'l Conference on Knowledge Discovery in Databases and Data Mining*, 1998.