⊠ 응용논문

수량적 속성을 포함하는 항목 제약을 고려한 연관규칙 마이닝 앨고리듬
- An Association Discovery Algorithm Containing Quantitative Attributes with Item Constraints -

한 경 록*
Han, Kyong Rok
김 재 련**
Kim, Jae Yearn

Abstract

The problem of discovering association rules has received considerable research attention and several fast algorithms for mining association rules have been developed. In this paper, we propose an efficient algorithm for mining quantitative association rules with item constraints. For categorical attributes, we map the values of the attribute to a set of consecutive integers. For quantitative attributes, we can partition the attribute into values or ranges. While such constraints can be applied as a post-processing step, integrating them into the mining algorithm can reduce the execution time. We consider the problem of integrating constraints that are boolean expressions over the presence or absence of items containing quantitative attributes into the association discovery algorithm using Apriori concept.

1. 서 론

1.1 연구 배경

많은 양의 데이터가 생성, 수집, 저장됨에 따라 데이터베이스가 대용량화되면서 데이터들 사이의 숨겨진 연관성을 정보, 지식으로 변환할 수 있는 새로운 기술 및 도구가 필요하게 되었다. 따라서, 감추어져 있긴 하나 잠재적 사용가치가 큰 패턴이나 추세 및 흥미로운 규칙들을 발견하고자 하는 연구가 활발히 진행되기 시작했고, 표면적으로는 관련되지 않아 보이는 데이터들이 새롭고 유용한 정보를 창출하게 되었다. 대용량 데이터베이스에서의 지식 발견(knowledge discovery in databases)이라고 정의되는 데이터 마이닝(data mining)은 최근 들어 시장전략 수립, 수요예측, 의료진단, 상품진열 등 광범위한 분야에 응용되고 있으며[3,4,7], 감(feeling)을 사실(fact)로 전환시킬 수 있는 능력이 큰 장점으로 인식되고 있다. 이러한 데이터 마이닝의 기법에는 신경망(neural networks), 분류(classification), 연관규칙(association rules), 순차패턴(sequential pattern), 군집화(clustering), 유전 앨고리듬(genetic algorithm)등이 있다.

* 한양대학교 대학원 산업공학과
** 한양대학교 산업공학과 교수

대용량의 데이터베이스에서 어떤 사건들이 함께 발생하거나, 또는 하나의 사건이 다른 사건을 암시하는 것과 같은 사건간의 상호관계를 나타내는 연관규칙을 발견하는 문제는 가장 중요한 데이터 마이닝 문제들 중의 하나이다. 이 문제는 Agrawal 등에 의해 처음 제기되었다[1]. 하나의 트랜잭션을 여러 개의 항목들(items)의 집합으로 보고 이러한 트랜잭션들이 하나의 집합으로 주어지면, 연관규칙은 "X⇒Y"라는 형태로 표현되며 여기서 X와 Y는 항목들의 집합들이다. 그러한 연관규칙의 의미는 X를 포함하는 트랜잭션들이 Y 또한 포함하는 경향이 있다는 뜻이다. 예를 들어, "라면을 구입하는 고객 중에서 90%의 고객이 김치를 같이 구입한다." 또는 "모든 트랜잭션들 중에서 5%는 맥주와 새우깡을 함께 포함하고 있다."라는 정보는 연관규칙이다. 여기서 90%를 신뢰도(confidence)라 하고, 5%를 지지도(support)라고 부른다. 그러므로, 사용자가 지정한 최소지지도(minimum support)와 최소신뢰도(minimum confidence)의 조건을 만족하는 모든 연관규칙을 찾는 것은 중요한 일이 된다[4,6].

연관규칙들을 발견하는 문제는 다음의 두 가지 하위 문제로 세분화할 수 있다[9]. :

1) 사용자가 지정한 최소지지도 이상의 지지도를 갖는 항목집합들(itemsets)을 찾는 단계이다. 항목 집합에 대한 지지도란 그 항목집합을 포함하는 트랜잭션들의 수를 의미한다. 여기서 최소지지도를 만족하는 항목집합을 빈발(large or frequent) 항목집합이라 부르며, 그 이외의 항목집합은 비빈발(small) 항목집합이라고 한다.

2) 빈발 항목집합들을 이용하여 규칙을 생성시키는 단계이다. 예를 들어, ABCD와 AB가 빈발 항목집합들이라면, "신뢰도=지지도(ABCD)/지지도(AB)"의 비율을 계산함으로써 "AB⇒CD"와 같은 연관규칙을 생성시킬 수 있다. 만약 "신뢰도≥최소신뢰도"이면 그 규칙은 강한(strong) 연관규칙이라고 부르며[5,6] 사용자에게는 잠재적 사용가치가 큰 정보 및 지식으로 인식된다. (이 규칙은 ABCD가 빈발이기 때문에 최소지지도를 가질 것이다.) 일단 빈발 항목집합들이 찾아지면 연관규칙을 생성하는 일은 쉬우므로 첫 단계인 빈발 항목집합을 찾는 데 주안점을 두기로 한다.

1.2 연구 목적

이와 같이, 연관규칙을 이용한 데이터 마이닝은 주로 항목 자체에만 흥미를 두고 연구되었기 때문에 "노트 2권과 연필 3자루 이상을 구입하는 고객들의 85%가 지우개를 1개 구입한다."와 같은 수량적 속성이 포함된 연관규칙에 대해서는 소홀했다. 다시 말해서 "노트와 연필을 구입하는 고객들의 85%가 지우개를 구입한다."와 같은 연관규칙은 각 항목에 내포된 수량적 개념이 전혀 고려되지 않고 있다. 또한, 실제적으로 사용자는 "노트"와 "연필"을 구입하는 고객에게만 관심이 있으며 그 항목들에 관련된 연관규칙만 의미 있는 정보로서 받아들인다면 그 항목들과는 상관없는 다른 항목들간의 연관규칙은 많은 시간을 소비하면서 발견되었을지라도 전혀 무의미하게 되어 버린다. 물론 모든 빈발 항목집합을 발견한 후에 관심 있는 특정 항목이 들어있는 항목집합들만 골라내도 되지만, 실행시간을 줄일 수 있도록 하기 위해 처음부터 항목 제약을 주는 것이 보다 효율적이다.

즉, 본 연구의 취지는 숨겨져 있는 의미 있는 정보를 찾기 위해 행해진 데이터 마이닝이 수량적 정보를 포함하지 않거나 또는 특정 사용자에게는 관심이 없는 항목들 사이의 연관규칙을 만들어냄으로써, 사용자에게 꼭 필요한 정보를 찾기 위해서는 발견된 결과를 대상으로 또다시 마이닝을 수행해야 하는 단점을 극복하려는 데 있다. 본 논문에서는 항목들의 수량적 속성을 포함하는 트랜잭션에 특정 항목에 대한 제약을 고려한 연관규칙 발견 앨고리듬을 제시한다. 각 항목의 수량적 속성을 값이나 구간으로 분할하고, 연관규칙을 모두 찾은 후에 제약 조건을 적용하기보다는 특정 항목의 존재 유무를 Boolean 제약식으로 명시하고 Apriori 앨고리듬을 변형한 연관규칙 발견 앨고리듬에 통합하여 마이닝 시간을 줄임으로써 빠르고 정확한 의사결정 과정을 지원하도록 한다.

1.3 기존 연구

연관규칙 문제가 제기된 초창기에는 규칙이나 패턴을 찾는 빠르고 효율적인 앨고리듬의 개발과, 데이터베이스가 갱신됨에 따라 새로운 연관규칙이 발견되거나 기존의 연관규칙이 소멸될 수 있기 때문에 발견된 규칙들을 유지, 관리하는 데 그 초점이 맞춰졌다[7]. 최근에는 항목들을 범주적 속성(categorical attributes)과 수량적 속성(quantitative attributes)으로 세분화하여 접근하면서 두 속성을 같이 포함하는 연관규칙 발견과 항목 제약을 고려한 연관규칙 발견에 대한 문제가 소개되고 있다. 항목 자체에만 관심을 두면 그 안에 내포된 수량적 개념이 의미를 잃어버리게 되고, 또 사용자가 특정 항목에만 관심이 있다면 데이터 마이닝을 통해 발견된 모든 빈발 항목집합들 중에서 관심 있는 항목이 아닌 다른 항목들 사이의 관계는 가치 있는 정보를 제공하지는 않는다는 데 착안을 한 연구이다. 다만 전자는 기존의 Apriori 앨고리듬을 거의 수용하면서 수량적 속성을 추가하였고[8], 후자는 Apriori 앨고리듬을 약간 수정한 Direct 앨고리듬을 제시하고 있다[9]. 위의 두 가지 앨고리듬에 관한 설명은 2장에 나와 있다.

본 연구는 다음과 같이 구성되어 있다. 2장은 연관규칙에 대한 개념 설명과 기존 연구를 고찰해 보고, 3장에서는 본 연구가 제안하는 항목 제약을 고려한 수량적 연관규칙을 발견하는 앨고리듬을 수치 예제와 함께 논의한다. 4장은 제안하는 앨고리듬을 사용하여 수행된 실험 결과이다. 5장은 본 연구의 결론을 기술한다.

2. 연관규칙과 기존 앨고리듬 연구

2.1 연관규칙

$I=\{i_1,i_2,\cdots\cdots,i_m\}$를 항목(item)이라 불리는 문자들의 집합이라고 하자. D는 트랜잭션들의 집합이고 각 트랜잭션 T는 $T\subseteq I$인 항목들의 집합이라고 하자. 한 트랜잭션에서 구입하는 항목들의 수량은 고려하지 않기로 가정한다. I의 원소인 항목들의 집합(itemset)을 X라 할 때 $X\subseteq T$이면 "트랜잭션 T가 X를 포함한다."라고 말한다. 연관규칙은 "$X\Rightarrow Y$"의 형태로 표시되고, 여기서 $X\subset I$, $Y\subset I$ 및 $X\cap Y=\varnothing$이다. 만약 X를 포함하는 D에 있는 트랜잭션들의 c%가 Y 또한 포함하고 있으면 연관규칙 $X\Rightarrow Y$는 트랜잭션들의 집합 D에서 신뢰도 c를 가지고 있다는 뜻이다. 만약 D에 있는 트랜잭션들의 s%가 $X\cup Y$를 포함하면 연관규칙 $X\Rightarrow Y$는 트랜잭션들의 집합 D에서 지지도 s를 가지고 있음을 의미한다[2].

최소지지도 이상을 갖는 항목집합을 빈발 항목집합이라 한다. k개의 항목들로 이루어진 빈발 항목집합을 빈발 k-항목집합이라 한다. 빈발 k-항목집합들의 집합을 L_k라 하고, 이를 생성하기 위한 후보 k-항목집합들의 집합을 C_k(잠재적 빈발 항목집합)라 한다. D가 주어졌을 때, 연관규칙 문제는 사용자가 지정한 최소지지도(minsup.)와 최소신뢰도(minconf.) 이상의 지지도와 신뢰도를 갖는 모든 빈발 항목집합들과 연관규칙을 생성시키는 문제다.

2.2 Apriori 앨고리듬[2]

앨고리듬의 첫 번째 시행(pass)에서는 빈발 1-항목집합들을 결정하기 위해 단순히 모든 트랜잭션을 읽어서 각 항목별로 빈도수를 계산한다. 이렇게 지지도가 계산된 항목들 중에서 사용자가 정한 최소지지도를 만족하는 항목들만이 L_1을 구성한다. $k(k\geq 2)$ 번째 시행부터는 두 가지 단계를 고려한다. 먼저, (k-1) 번째 시행에서 발견된 빈발 항목집합들인 L_{k-1}을 후보 항목집합들인 C_k를 발생시키기 위해 사용한다. 다음으로, 데이터베이스가 검색되어 C_k에 있는 후보들의 지지도가 계산되고 최소지지도를 만족하는 C_k만이 L_k로 진입한다. 이러한 시행이 반복되어 $L_k(k\geq 1)$가 공집합($\varnothing$)이 되면 앨고리듬을 종료한다. 앨고리듬에서 가장 중요한 부분으로서 C_k를 발생시키는 과정인 apriori-gen 함수는 join과 prune 두 가지 단계로 구성된다.

join 단계에서는, C_k를 생성하기 위해 $L_{k-1}*L_{k-1}$(self-join)을 사용하는데, 여기서 *는 연결(concatenation) 연산이다. prune 단계에서는, join 단계에서 생성된 C_k에 있는 모든 항목집합들 $c(c \in C_k)$에 대해 c의 어떤 (k-1)-부분집합이 L_{k-1}에 존재하지 않으면 그 후보 c를 삭제한다.

2.3 항목 제약을 고려한 Direct 앨고리듬[9]

실제적으로, 사용자들은 연관규칙들 중 일부에만 관심이 있는 경우가 많다. 즉, 특정 항목들만 포함하고 있는 연관규칙들을 필요로 하기도 하는데, 예를 들면 10개의 항목이 있는 데이터베이스에서 어떤 사용자가 2개의 항목에만 흥미를 느낀다면 그 이외의 8개 항목에 관해 빈발하는 항목집합들과 연관규칙들은 비록 많은 컴퓨팅 시간을 소비하면서 발견되었을지라도 그 사용자에게는 어떠한 의미있는 정보도 제공해 주지 못한다. 이러한 관점에서 Direct 앨고리듬은 미리 Boolean 제약식을 명시하여 특정 항목에 대한 제약을 주고 Apriori 앨고리듬의 기본 개념을 이용해서 join과 prune 단계를 반복하면서 사용자에게 흥미 있는 항목들만 빈발 항목집합으로 선택한다. 또한 Direct 앨고리듬은 Boolean 제약식이 어떻게 주어지느냐에 따라 최초의 시행과정이 달라질 수도 있다. 아래에 설명된 단계를 보면, Apriori 앨고리듬의 join과 prune 단계의 개념을 사용하지만 self-join을 하지 않고 최초의 시행에서 빈발 항목집합을 F로 두고, 거기에서 Boolean 제약식을 만족하는 빈발 1-항목집합들의 집합만을 L_1^b로 진입시킨다.

1. $C_{k+1}^b := L_k^b \times F$;
2. C_{k+1}^b에 있는 모든 후보들 중, B를 만족하지 않는 후보들을 삭제한다.
3. C_{k+1}^b에 있는 모든 후보들 중, B는 만족하지만 최소지지도를 만족하지 않는 k-부분집합을 가진 후보들을 삭제한다.
4. (k+1)개의 부정부호(¬)없는 원소로만 이루어진 B에 있는 각 D_i에 대해서, 모든 원소가 빈발이면 C_{k+1}^b에 더한다.

예를 들어, L이 항목들의 집합이라고 할 때, L={1,2,3,4,5}이고 $B=(1 \wedge 2) \vee (4 \wedge \neg 5)$라고 하자. 여기서, 모든 항목들이 빈발이라고 가정하면 L_1^b={{4}} 이다. C_2^b를 생성하기 위해, 먼저 $L_1^b \times F$를 계산하여 {{1 4},{2 4},{3 4},{4 5}}를 얻는다. {4 5}는 B를 만족하지 않으므로 삭제된다. B를 만족하는 모든 1-부분집합들이 빈발이기 때문에 위의 "단계 3"을 적용해도 C_2^b에서 삭제되는 후보들은 없다. 마지막으로 {1 2}를 C_2^b에 더하여 {{1 2},{1 4},{2 4},{3 4}}를 얻는다.

3. 제안하는 앨고리듬

3.1 기호 설명

D : 트랜잭션들의 집합 , BT : Boolean Table(0과 1로 이루어짐) , N : 항목들의 수
T : D의 트랜잭션 , t : BT의 트랜잭션 , I : 항목들의 집합 , TN : D의 트랜잭션들의 수
l_{ij} : 각각의 항목들 - (a,b) , (a,b+)로 표현 , (a,b) : (항목,수량) 또는 (항목,순서화된 속성치)
B(Boolean 제약식) : $D_1 \vee D_2 \vee \cdots \vee D_m$ (m disjuncts)
α_{ij} : l_{ij} 또는 $\neg l_{ij}$ (항목의 존재 유무) , D_i : $\alpha_{i1} \wedge \alpha_{i2} \wedge \cdots \wedge \alpha_{ij}$ (j conjuncts in D_i)
F : 첫 번째 시행에서 빈발하는 항목들의 집합
k-항목집합 : k개의 항목들을 갖는 항목집합
k-부분집합 : k개의 원소가 있는 부분집합
L_1^b : B를 만족하는 빈발 1-항목집합들의 집합
C_k^b : B를 만족하는 후보 k-항목집합들의 집합
L_k^b : B를 만족하는 빈발 k-항목집합들의 집합
최소신뢰도(minimum confidence) : minconf. , 최소지지도(minimum support) : minsup.

3.2 앨고리듬 설명

수량적 정보가 들어있는 연관규칙을 발견하기 위해서 먼저 수량적 속성에 대해서는 각 항목과 수량과의 관계를 순서쌍 즉 (항목,수량)으로 표시하고 범주적 속성에 대해서는 (항목,범주적 속성치)로 표현한다. 수량적 속성은 범위(구간)로도 표현이 가능하고 값으로도 표현이 가능하다. 그래서 수량의 크기가 작으면 각각의 수량을 독립적인 항목으로 인식하고 값으로 처리해도 되지만 크기가 커지면 구간으로 분할(partition)한 다음에 각 구간을 순서화된 (consecutive) 정수치로 변환해야 한다. 여기서는 몇 개 이상은 "+"라는 기호를 사용한다.

예를 들어, "1"이라는 항목을 3개 샀다면 (1,3)으로 표시하고, "3"이라는 항목을 6개에서 10개 사이로 샀으면 (3,6,10)으로 나타낸다. "2"라는 항목을 3개 이상 샀다면 (2,3+)로 표시하기로 한다. 범주적 속성은 구간값으로 나눌 수가 없어서 단지 범주적 속성치를 순서화된 정수값으로 변환(mapping)시키면 된다. 변환하는 과정은 3.3절에 있는 <그림 4>에 나타나 있다. 위와 같은 변환 방법을 통해서 수량적 연관규칙 문제를 Boolean 연관규칙 문제로 변환시킨다. 변환된 Boolean Table에서 값이나 구간을 가지고 나누어진 각각의 항목들에 대해 지지도를 계산하여 최소지지도를 만족하는 항목들만을 가지고서 빈발 1-항목집합(F)으로 놓고 앨고리듬을 시작한다. 본 논문에서 제안하는 앨고리듬의 단계를 <그림 1>에서 설명하고 있다.

단계 1. 트랜잭션들의 집합 D에 있는 속성들을 수량적 속성과 범주적 속성으로 구별한다. 수량적 속성의 경우 수량의 크기가 크면 구간으로 분할한 뒤 각각의 구간을 다시 순서화된 정수치로 변환하고, 수량의 크기가 작으면(즉 구간으로 나눌 필요가 없으면) 구간을 나누지 않고 값을 그대로 사용한다. 범주적 속성치를 순서화된 정수치로 변환한다.

단계 2. 트랜잭션들의 집합 D를 Boolean Table로 변환하여 각 항목들이 몇 번씩 발생하는지 지지도를 계산한다.

단계 3. 최소지지도를 만족하는 항목들만 뽑아서 F라 하고 F의 원소들 중에서 Boolean 제약식을 만족하는 빈발 1-항목집합들을 선택하여 L_1^b를 생성한다. 만약 L_1^b를 생성할 수 없는 경우는, Boolean 제약식에 있는 각 D_i에 대해서 부정부호(¬)가 없는 α_{ij}가 $k(2 \le k \le N)$개이고 그 α_{ij}가 모두 빈발이면 α_{ij}의 개수인 k에 해당하는 C_k^b의 원소로 취한다. k값이 가장 작은 C_k^b에서부터 단계 6을 시작하고, 가장 작은 k값을 제외한 그 이외의 값을 갖는 D_i는 해당 $C_k^b(=L_{k-1}^b \times F)$의 계산이 수행될 때 바로 후보로서 사용된다.

단계 4. join : $C_{k+1}^b := L_k^b \times F \ (k \ge 1)$: L_1^b와 앞서 구한 F를 join하여 C_2^b를 구한다.

단계 5. prune(1) : C_{k+1}^b에 있는 모든 후보들 중, Boolean 제약식을 만족하지 않는 후보들을 제거한다.

단계 6. prune(2) : C_{k+1}^b에서 Boolean 제약식을 만족하지만 최소지지도를 갖지 못하는 k-부분집합을 가진 후보들을 제거한다.

단계 7. Boolean 제약식에서 정확히 (k+1)개의 부정부호(¬)가 없는 원소들을 가진 각 D_i에 대해서, 만약 그 모든 원소들이 빈발이면 그 항목집합을 C_{k+1}^b에 추가한다.

단계 8. 위의 3번 ~ 7번 단계를 통해 얻어진 C_{k+1}^b에 있는 (k+1)-항목집합들의 지지도를 계산한다.

단계 9. 최소지지도를 만족하는 것만 L_{k+1}^b로 취하며, 이 과정을 계속 반복해서 L_{k+1}^b가 공집합(∅)이면 실행을 종료한다.

<그림 1> 제안하는 앨고리듬의 단계

그리고 본 논문에서는 다른 항목들 사이의 연관규칙만을 찾는 것으로 가정하고, 동일한 항목이 서로 다른 D_i에 나타날 때의 경우는 고려하지 않기로 한다. 예를 들어, B가 $\{(1,2) \vee (1,3)\}$라고 주어지면 1번 항목을 2개 사거나 1번 항목을 3개 구입한 고객들 사이의 연관규칙에 흥미가 있다는 뜻인데, 1번 항목을 3개 구입했다는 말은 이미 2개를 샀다는 뜻이 되어서 논리적으로 모순된 제약식이 되기 때문이다.

다음은 본 논문에서 제안하는 앨고리듬을 기존에 제안된 Direct 앨고리듬과 비교하여 차이점을 서술한다. Direct 앨고리듬은 L_1^b를 만들 수 있는 경우만 고려했는데, Boolean 제약식이 어떻게 주어지느냐에 따라 L_1^b를 찾을 수 없는 경우도 생기므로 최초의 시행과정이 달라질 수 있다. 즉, 주어진 Boolean 제약식을 만족하는 L_1^b를 만들 수 있느냐 없느냐에 따라서 차이를 두고 앨고리듬이 실행되어야 한다. 기존 앨고리듬은 L_1^b를 찾을 수 있는 경우만 제시했으나 본 논문에서 제안하는 앨고리듬에서는 L_1^b를 찾을 수 없는 경우의 해결 방법까지 제시한다.

예를 들어, Boolean 제약식이 {(A∧B)∨(C∧E)}와 같이 주어졌다고 하자. 이 경우는 A와 B를 같이 구입한 고객 또는 C와 E를 함께 구매한 고객에 관한 연관규칙을 찾는다는 뜻으로 해석되고, 따라서 2개의 항목이 같이 빈발해야 되므로 주어진 Boolean 제약식을 만족하는 빈발 1-항목집합(L_1^b)을 만들 수 없게 된다. 이 예제에서는 먼저 첫 번째 시행을 통해 최소지지도를 만족하는 항목들의 집합인 F를 생성한 다음에, L_1^b를 만들 수 없으므로 (A B)와 (C E)를 C_2^b의 원소로 사용하여 L_1^b가 아닌 C_2^b에서부터 앨고리듬의 join과 prune 과정을 반복한다. 다만 여기서 주의할 점은 A, B, C, E 각각이 모두 최초의 시행에서 얻어진 빈발 1-항목집합(F)의 원소여야만 한다. 만약 (A B)의 1-부분집합인 A와 B 중에서 하나라도 빈발이 아니면 (A B)은 L_2^b의 원소가 될 수 없고, 또 (C E)의 1-부분집합인 C와 E 중에서 하나라도 빈발이 아니면 (C E)은 L_2^b의 원소가 될 수 없다. 다시 말해서, Boolean 제약식이 k(k≥2)개 이상의 항목들이 같이 빈발이도록 주어지면 $L_{k-i}^b(1 \le i \le k-1)$을 생성할 수 없으므로, 바로 C_k^b부터 시작을 하되 반드시 1-부분집합들이 모두 빈발이어야 한다.

또 다른 예제로서 {(1,2)∧(2,1)∧(5,2)}는 "1", "2", "5" 항목이 같이 빈발해야 하고, {{(1,2)∧(2,2)}∨{(3,1)∧(5,3+)}}는 "1", "2" 항목이 같이 빈발하거나 "3", "5" 항목이 같이 빈발해야 하며, {{(1,1)∧(3,2)}∨{(2,2)∧(4,2)∧(5,2)}}와 같은 Boolean 제약식이 주어지면 "1", "3" 항목 또는 "2", "4", "5" 항목들은 꼭 같이 빈발해야 한다는 뜻이다. 그렇다면, 이 과정에서 k값이 가장 작은 C_k^b보다 이전 단계의 모든 후보 항목집합과 빈발 항목집합은 아무런 의미가 없어진다.(즉, 사용자에게 어떠한 정보 제공도 없다.) 다시 말해서 {(1,2)∧(2,1)∧(5,2)}와 같은 경우라면 3개의 항목이 함께 빈발해야 하기 때문에 C_3^b에서부터 앨고리듬을 실행하게 되고 L_1^b와 L_2^b는 발견할 수가 없게 된다. 주의할 점은 <그림 1>의 단계 7에 의해서 (1,2), (2,1), (5,2) 모두가 각각 빈발이어야 한다. 세 개의 항목 중에서 하나라도 빈발이 아니면 C_3^b의 원소가 될 수 없다.

3.3 수치 예제(1)

<그림 2>는 수치 예제(1)을 위한 데이터베이스이고 <그림 3>은 Boolean Table로 변환시키고 각 항목의 지지도를 계산한 상태를 보여준다. 이 예제는 범주적 속성과 수량적 속성이 같이 포함되어 있는데, 여기서 성별은 범주적 속성으로서 남자(M)와 여자(F)로 구분된다. 나이는 수량적 속성으로서 구간의 크기를 10으로 하여 1~10살, 11~20살, 21살 이상으로 분할하였고, 21살 이상은 "21+"와 같이 나타낸다. A, B, C는 품목의 명칭으로서 구입한 수량을 고려하기로 한다.

예제에서는 나이를 "1"로, 성별을 "2"로, A, B, C 각각의 품목을 "3", "4", "5"로 대응시켰다. 또한, 1~10살은 "1", 11~20살은 "2", 21살 이상은 "3"과 같이 순서화된 양의 정수로 변환(mapping)시키고 여자(F)는 "1", 남자(M)는 "2"로 대응시킨다. 각 품목은 1개, 2개 이상으로 분할하여 "B" 품목을 2개 이상 구입하였다면 (4,2+)로 표시한다. 최소지지도는 20%, 최소신뢰도는 80%, Boolean 제약식은 {(1,3)∧(4,2+)}∨{(2,2)}로 주었다. 본 논문의 앨고리듬을 적용하기 위하여 정수로 대응(mapping)시키는 과정이 <그림 4>에 나와 있고, 주어진 조건을 만족하는 빈발 항목집합들은 L_1^b={(2,2)}, L_2^b={{(1,3),(2,2)},{(2,2),(3,1)},{(2,2),(4,2+)},{(2,2),(5,1)},{(1,3)

,(4,2+)}}, L_3^b={{(1,3),(2,2),(3,1)}}로서 얻어졌다. 구해진 빈발 항목집합들을 사용하여 앞의 1.1 절에서 설명한 연관규칙 생성 공식에 의해 최소신뢰도 80%를 만족하는 연관규칙을 발견한다.

번호	나이	성별	항목		
			A	B	C
1	17	F	3	2	3
2	23	M	1	0	1
3	25	M	0	1	0
⋮	⋮	⋮	⋮	⋮	⋮
23	17	F	1	0	2
24	18	M	2	0	1
25	19	M	0	0	1

<그림 2> 예제 데이터베이스 - 수치 예제(1)

번호	(1,1)	(1,2)	(1,3)	……	(5,1)	(5,2+)
1	0	1	0		0	1
2	0	0	1		1	0
3	0	0	1		0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
23	0	1	0		0	1
24	0	1	0		1	0
25	0	1	0		1	0
Support	2	8	15	……	8	9

<그림 3> BT(Boolean Table) - 수치 예제(1)

속성	정수치
나이	1
성별	2
A	3
B	4
C	5

Partition	Mapping
(나이,1,10)	(1,1)
(나이,11,20)	(1,2)
(나이,21+)	(1,3)
(성별,F)	(2,1)
(성별,M)	(2,2)

항목	Mapping
A	(3,1) , (3,2+)
B	(4,1) , (4,2+)
C	(5,1) , (5,2+)

나이(1)는 3개로 분할, 성별(2)은 2개로 분할, "A"항목(3), "B"항목(4), "C"항목(5)은 각각 "1개" , "2개 이상" 구입으로 분할
모든 항목의 최소지지도는 20% , 최소신뢰도는 80%로 지정
Boolean 제약식(B)은 {(1,3)∧(4,2+)}∨{(2,2)}
L_1^b = {(2,2)}
L_2^b = {{(1,3),(2,2)},{(2,2),(3,1)},{(2,2),(4,2+)},{(2,2),(5,1)},{(1,3),(4,2+)}}
L_3^b = {(1,3),(2,2),(3,1)}

<그림 4> 변환과정과 결과값 - 수치 예제(1)

3.4 수치 예제(2)

<그림 5>는 수치 예제(2)를 위한 데이터베이스이고 <그림 6>은 Boolean Table로 변환시키고 각 항목의 지지도를 계산한 상태를 보여준다. 다음은 25개의 트랜잭션을 대상으로 고객이 구입한 항목들의 수량을 고려하여 사용자가 원하는 특정 항목을 Boolean 제약식으로 명시하고 앨고리듬을 적용한 예제이다. 사용자는 자신이 흥미를 갖고 있는 항목에 대한 정보들을 찾아낼 수 있음을 보여준다.

L={1,2,3,4,5}={노트,볼펜,지우개,칼,연필}이라 하고 수량은 1개, 2개, 3개 이상으로 분할(partition)했으며, 최소지지도를 20%(총 트랜잭션 수가 25개이므로 5개(25×0.2) 이상이면 빈발임.), 최소신뢰도를 85%로 지정했다. B={(1,2)∧(5,3+)}∨{(2,2)}인데, 즉 노트 2권과 연필 3자루 이상을 같이 구입하거나 볼펜 2자루를 구입한 고객에게만 관심이 있다는 의미이다. <그림 6>에서 최소지지도를 만족하는 항목들을 F={(1,1),(1,2),(1,3+),(2,2),(3,1),(4,1),(5,2),(5,3+)}라 하면, L_1^b={(2,2)}이다. C_2^b=F×L_1^b를 이용한 연산결과, C_2^b={{(1,1),(2,2)},{(1,2),(2,2)},{(1,3+),(2,2)},{(2,2),(3,1)},{(2,2),(4,1)},{(2,2),(5,2)},{(2,2),(5,3+)}}이다. C_2^b의 원소 모두가 B를 만족하고 각 원소에 대한 <그림 1>의 단계 6에 있는 pruning이 발생하지 않으며, 단지 단계 7을 실행하면서 {(1,2),(5,3+)}가 C_2^b에 추가된다.

지지도 계산 결과, L_2^b={{(1,2),(5,3+)},{(1,3+),(2,2)},{(2,2),(5,2)}}이고, 같은 방법으로 C_3^b=F ×L_2^b={{(1,2),(2,2),(5,3+)},{(1,2),(3,1),(5,3+)},{(1,2),(4,1),(5,3+)},{(1,3+),(2,2),(3,1)},{(1,3+),(2,2),(4,1)},{(1,3+),(2,2),(5,2)},{(1,3+),(2,2),(5,3+)},{(1,1),(2,2),(5,2)},{(1,2),(2,2),(5,2)},{(2,2),(3,1),(5,2)},{(2,2),(4,1),(5,2)}}이다. 일단 모두 B를 만족하지만, {(1,2),(2,2),(5,3+)}는 2-부분집합 {(1,2),(2,2)}이 B 만족이나 지지도가 0 이고, {(1,3+),(2,2),(3,1)}는 {(2,2),(3,1)}이 B 만족이나 지지도가 0, {(1,3+),(2,2),(4,1)}는 {(2,2),(4,1)}이 B 만족이나 지지도가 1, {(1,3+),(2,2),(5,3+)}는 {(2,2),(5,3+)}이 B 만족이나 지지도가 0, {(1,1),(2,2),(5,2)}는 {(1,1),(2,2)}이 B 만족이나 지지도가 1, {(1,2),(2,2),(5,2)}는 {(1,2),(2,2)}이 B 만족이나 지지도가 0, {(2,2),(3,1),(5,2)}는 {(2,2),(3,1)}이 B 만족이나 지지도가 0, {(2,2),(4,1),(5,2)}는 {(2,2),(4,1)}이 B 만족이나 지지도가 1 이므로 삭제된다. <그림 1>의 단계 7을 적용할 때에는 {(△)∧(◇)∧(▽)}와 같이 항목 세 개가 함께 발생하는 경우가 Boolean 제약식에 없으므로 C_3^b에 추가될 항목이 없다. 지지도를 계산하여 L_3^b={{(1,2),(3,1),(5,3+)},{(1,3+),(2,2),(5,2)}}를 생성한다.

번호	Items				
1	(1,3)	(2,2)	(3,3)	(5,2)	
2	(1,1)	(3,1)	(4,1)	(5,2)	
3	(2,1)	(3,2)			
4	(1,4)	(2,2)	(5,2)		
⋮	⋮	⋮	⋮	⋮	⋮
22	(3,1)	(4,1)	(5,3)		
23	(1,1)	(2,2)			
24	(2,1)	(3,1)	(4,1)		
25	(1,2)	(3,1)	(5,4)		

<그림 5> 예제 데이터베이스
- 수치 예제(2)

번호	(1,1)	(1,2)	(1,3+)	·····	(5,1)	(5,2)	(5,3+)
1	0	0	1		0	1	0
2	1	0	0		0	1	0
3	0	0	0		0	0	0
4	0	0	1		0	1	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
22	0	0	0		0	0	1
23	1	0	0		0	0	0
24	0	0	0		0	0	0
25	0	1	0		0	0	1
Support	5	7	6	·····	2	10	7

<그림 6> BT(Boolean Table)
- 수치 예제(2)

같은 방법으로 C_4^b={{(1,2),(2,2),(3,1),(5,3+)},{(1,2),(3,1),(4,1),(5,3+)},{(1,3+),(2,2),(3,1),(5,2)},{(1,3+),(2,2),(4,1),(5,2)}}가 되고, 이들은 B를 모두 만족한다. 그러나, {(1,2),(2,2),(3,1),(5,3+)}는 {(1,2),(2,2),(3,1)}이 B 만족이나 지지도가 0, {(1,2),(3,1),(4,1),(5,3+)}는 {(1,2),(4,1),(5,3+)}이 B 만족이나 지지도가 1, {(1,3+),(2,2),(3,1),(5,2)}는 {(1,3+),(2,2),(3,1)}이 B 만족이나 지지도가 0, {(1,3+),(2,2),(4,1),(5,2)}는 {(1,3+),(2,2),(4,1)}이 B 만족이나 지지도가 1 이기 때문에 삭제된다. C_4^b가 모두 삭제되어서 L_4^b를 생성할 수 없으므로 실행을 종료한다. 따라서, B를 만족하는 빈발 항목집합들은 L_1^b={(2,2)}, L_2^b={{(1,2),(5,3+)},{(1,3+),(2,2)},{(2,2),(5,2)}}, L_3^b={{(1,2),(3,1),(5,3+)},{(1,3+),(2,2),(5,2)}}와 같이 얻어진다.

이 예제에서 알 수 있듯이, 빈발 1-항목집합들은 사용자에게 가치 있는 정보인 2개 이상의 다른 항목간의 연관규칙을 찾기 위한 최초의 수단일 뿐, 그 자체로는 연관규칙의 의미가 없다. 그리고, 빈발하는 항목들의 집합들과 그 집합의 항목들 사이의 연관관계가 대등하게 중요성을 갖지는 않는다. 다시 말해서, 빈발이라고 해도 그 빈발 항목집합의 항목들 사이의 연관이 최소신뢰도 미만이 되어 아무런 정보를 얻지 못하는 수도 있다. 즉, (1,2)⇒(5,3+)과 같은 연관규칙은 신뢰도가 6/7, 즉 85.7%로서 최소신뢰도를 만족하므로 의미 있는 정보를 제공하지만 (2,2)⇒(5,2)는 신뢰도가 5/6이므로 83.3%로서 최소신뢰도를 만족하지 못한다. 물론 최소지지도와 최소신뢰도는 사용자가 지정해 주기 때문에 다양한 값을 주면서 다른 결과들을 얻어내는 것은 가능하다.

4. 실험 결과 및 분석

제안하는 앨고리듬의 성능평가를 위해 여섯 가지의 각기 다른 트랜잭션들의 집합을 가지고 실험을 수행하였다. Visual C++ 5.0을 사용하여 프로그램하였고, 실험은 CPU 120MHz, 메모리 32MB를 가진 컴퓨터에서 행해졌다. 트랜잭션의 수는 100개, 500개, 1000개로 하고 항목의 수는 3개, 5개로 하며 최소지지도를 조정해 가면서 실험을 수행했다. 즉 T100_N3, T100_N5, T500_N3, T500_N5, T1000_N3, T1000_N5의 여섯 가지 경우에 대해 최소지지도의 값을 변화해 가면서 실험을 수행하였다.

실험번호	T	N	Partition	최소지지도		
1	100	3	1;2, 2;3, 3;3	0.05	0.1	0.15
2	100	5	1;2, 2;2, 3;2, 4;2, 5;2	0.05	0.1	0.15
3	500	3	1;2, 2;2, 3;2	0.02	0.04	0.08
4	500	5	1;3, 2;3, 3;2, 4;2, 5;2	0.01	0.02	0.03
5	1000	3	1;3, 2;3, 3;3	0.01	0.02	0.03
6	1000	5	1;4, 2;4, 3;4, 4;4, 5;4	0.01	0.02	0.03

<그림 7> 실험 예제

<그림 7>에서 실험번호 4번의 경우는 트랜잭션이 500개이고 항목수가 5개이며 항목 "1"과 "2"는 3개로 분할했고(즉, 1개, 2개, 3개 이상) 항목 "3", "4", "5"는 2개의 구간으로 분할했다.(즉, 1개, 2+) 값이나 구간으로 나누어지는 분할(partition)의 개수는 사용자가 직접 입력하도록 했고 각 항목에 대해서 똑같은 분할을 할 필요는 없다. 그리고 최소지지도는 1%, 2%, 3%의 세 가지 경우에 대해서 실험했다. <그림 8>은 각각의 실험에서 명시된 Boolean 제약식(B)을 나타내며 <그림 9>는 여섯 번의 실험에 의해 발견된 빈발 항목집합들의 결과를 보여주고 있다.

실험번호	Boolean 제약식(B)
1	{ (1,1) ∧ (2,2) } ∨ { (3,3+) }
2	{ (1,2+) ∧ (5,2+) } ∨ { (3,1) }
3	{ (1,1) } ∨ { (3,2+) }
4	{ (1,1) ∧ (2,3+) } ∨ { (4,2+) ∧ (5,2+) }
5	{ (1,2) ∧ (3,3+) } ∨ { (2,3+) }
6	{ (1,2) ∧ (3,1) ∧ (5,4+) }

<그림 8> Boolean 제약식

실험번호	최소지지도	빈발 항목집합
1	0.05	{{(3,3+)},{(1,1),(3,3+)},{(1,2+),(3,3+)},{(2,2),(3,3+)},{(2,3+),(3,3+)},{(1,1),(2,2)},{(1,1),(2,2),(3,3+)},{(1,2+),(2,3+),(3,3+)}}
	0.1	{{(3,3+)},{(1,1),(3,3+)},{(1,2+),(3,3+)},{(2,2),(3,3+)},{(2,3+),(3,3+)},{(1,1),(2,2)}}
	0.15	{{(3,3+)},{(1,2+),(3,3+)},{(2,3+),(3,3+)}}
2	0.05	{{(3,1)},{(1,2+),(3,1)},{(2,2+),(3,1)},{(3,1),(4,1)},{(3,1),(4,2+)},{(3,1),(5,2+)},{(1,2+),(5,2+)},{(1,2+),(3,1),(5,2+)},{(2,2+),(3,1),(5,2+)},{(1,2+),(2,2+),(5,2+)},{(1,2+),(3,2+),(5,2+)},{(1,2+),(4,1),(5,2+)},{(1,2+),(2,2+),(3,2+),(5,2+)},{(1,2+),(2,2+),(4,1),(5,2+)}}
	0.1	{{(3,1)},{(3,1),(5,2+)},{(1,2+),(5,2+)},{(1,2+),(2,2+),(5,2+)},{(1,2+),(3,2+),(5,2+)}}
	0.15	{{(3,1)},{(1,2+),(5,2+)}}

3	0.02	{{(1,1)},{(3,2+)},{(1,1),(3,2+)},{(1,2+),(3,2+)},{(1,1),(2,1)},{(2,1),(3,2+)},{(1,1),(2,2+)},{(2,2+),(3,2+)},{(1,1),(3,1)},{(1,2+),(2,2+),(3,2+)}}
	0.04	{{(1,1)},{(3,2+)},{(1,2+),(3,2+)},{(1,1),(2,2+)},{(2,2+),(3,2+)},{(1,2+),(2,2+),(3,2+)}}
	0.08	{{(1,1)},{(3,2+)},{(1,2+),(3,2+)},{(2,2+),(3,2+)}}
4	0.01	{{(1,1),(2,3+)},{(4,2+),(5,2+)},{(1,3+),(4,2+),(5,2+)},{(2,2),(4,2+),(5,2+)},{(2,3+),(4,2+),(5,2+)},{(3,1),(4,2+),(5,2+)},{(3,2+),(4,2+),(5,2+)},{(1,1),(2,3+),(5,2+)},{(2,2),(3,1),(4,2+),(5,2+)},{(2,2),(3,2+),(4,2+),(5,2+)},{(2,3+),(3,2+),(4,2+),(5,2+)}}
	0.02	{{(1,1),(2,3+)},{(4,2+),(5,2+)},{(2,2),(4,2+),(5,2+)},{(3,1),(4,2+),(5,2+)},{(3,2+),(4,2+),(5,2+)}}
	0.03	{{(1,1),(2,3+)},{(4,2+),(5,2+)},{(3,2+),(4,2+),(5,2+)}}
5	0.01	{{(2,3+)},{(1,1),(2,3+)},{(1,2),(2,3+)},{(1,3+),(2,3+)},{(2,3+),(3,1)},{(2,3+),(3,3+)},{(1,2),(3,3+)},{(1,1),(2,3+),(3,3+)},{(1,2),(2,3+),(3,1)},{(1,2),(2,3+),(3,3+)},{(1,3+),(2,3+),(3,3+)},{(1,2),(2,2),(3,3+)}}
	0.02	{{(2,3+)},{(1,1),(2,3+)},{(1,2),(2,3+)},{(1,3+),(2,3+)},{(2,3+),(3,1)},{(2,3+),(3,3+)},{(1,2),(3,3+)},{(1,3+),(2,3+),(3,3+)}}
	0.03	{{(2,3+)},{(1,1),(2,3+)},{(1,2),(2,3+)},{(1,3+),(2,3+)},{(2,3+),(3,1)},{(2,3+),(3,3+)},{(1,2),(3,3+)}}
6	0.01	{{(1,2),(3,1),(5,4+)},{(1,2),(2,2),(3,1),(5,4+)},{(1,2),(3,1),(4,1),(5,4+)},{(1,2),(3,1),(4,2),(5,4+)},{(1,2),(2,2),(3,1),(4,2),(5,4+)}}
	0.02	{{(1,2),(3,1),(5,4+)}}
	0.03	해당 사항 없음

<그림 9> 실험 결과

위의 <그림 9>의 실험 번호 "6"을 살펴보면 최소지지도를 3%로 지정하는 경우에는 최소지지도를 만족하는 빈발 항목집합을 하나도 발견할 수 없음을 알 수 있다. 그리고 주어진 Boolean 제약식이 "1" 항목, "3" 항목, "5" 항목이 같이 빈발이도록 주어져서 빈발 1-항목집합과 빈발 2-항목집합은 찾지 않고 바로 L_3^b, L_4^b, L_5^b를 발견함을 보여준다.

다음은 Boolean 제약식에 부정부호가(¬)가 있는 경우를 보여준다. 지금까지의 예제에서는 모두 Boolean 제약식에 부정부호가 없었지만 부정부호가 있는 경우에도 본 논문에서 제안하는 앨고리듬을 적용할 수 있음을 설명한다. 예제로는 <그림 9>에 있는 실험 번호 2번의 트랜잭션들의 집합인 T100_N5를 그대로 사용한다. 다만 여기서 다루는 예제에서는 각 항목의 구간을 5개로 동일하게 분할하고 부정부호가 있는 Boolean 제약식을 B={(1,2)∧¬(3,1)}∨{(5,2)}로 명시하며 최소지지도를 8%로 정했다. 이 실험을 수행한 결과, L_1^b={{(1,2)},{(5,2)}, L_2^b={(1,2),(4,1)},{(4,1),(5,2)}}의 빈발 항목집합들을 얻을 수 있었다. 즉, 100개의 트랜잭션들 중에서 최소지지도를 만족하면서 8번 이상이 빈발하는 항목들은 빈발 1-항목집합이 2개이고 빈발 2-항목집합이 2개임을 알았고 사용자가 제약식(B)에 나타낸 특정 항목의 존재 유무를 고려하여 마이닝을 했다. 1번 항목을 2개 구입한 고객, 5번 항목을 2개 구입한 고객과 1번 항목 2개와 4번 항목 1개를 함께 구입하는 고객, 4번 항목 1개와 5번 항목 2개를 같이 구입하는 고객이 빈발로 나타났고 수량적 정보를 포함하여 사용자가 원하는 특정 항목들 사이의 연관관계만 마이닝할 수 있음을 보여준다. 사용자는 미리 지정한 최소신뢰도를 이용해서 항목들간의 연관성을 확률로 계산하여 발견한 연관규칙을 의사 결정에 응용할 수 있다.

5. 결론

데이터 마이닝은 "데이터베이스에서 데이터들 사이의 관계의 본질을 발견하는 과정"이다. 즉, 마치 거대한 철광석에서 철을 제련하는 것과 마찬가지로 거대한 정보의 섬에서 사용자에게 필요한 정보를 추출하는 일련의 과정을 뜻한다. 이러한 데이터 마이닝의 여러 가지 기법들 중에서 연관규칙은 증가하는 데이터들 사이에서 잠재적으로 유용한 정보를 제공하는 데 중요한 역할을 하고 있다. 본 연구에서는 대용량의 데이터베이스를 대상으로 사용자가 원하는 특정항목들 사이의 연관관계를 수량적 속성을 포함하여 마이닝하는 앨고리듬을 제안했다. 제안하는 앨고리듬을 이용하면 사용자는 분할하고자 하는 개수와 Boolean 제약식과 최소지지도를 입력하여 원하는 빈발 항목집합들을 빠른 시간에 찾아낼 수 있었다.

본 논문의 앨고리듬은 Boolean 제약식을 미리 명시함으로써 불필요하게 발견되는 빈발 항목집합들의 수를 줄여서 수행상의 속도를 효율적으로 높이면서 원하는 정보를 마이닝하며, 얻어진 정보들은 더 높은 수익성의 잠재력을 가진 질적으로 우수한 고객 발견 및 구매행태 분석, 시장 점유율 향상 및 수익성 개선, 제품 판촉 비용 감소와 재고관리, 마케팅 전략의 수립에 응용되어 비즈니스 성과를 높이고 고객 만족을 증진시킨다.

참고문헌

[1] R. Agrawal, T. Imielinski, A. Swami. "Mining Association Rules between Sets of Items in Large Databases", Proc. of the ACM SIGMOD Int'l Conference on Management of Data, Washington D.C., pp. 207-216, May 1993.

[2] R. Agrawal and R. Srikant. "Fast Algorithms for Mining Association Rules", In Proc. of the 20th Int'l Conference on Very Large Databases, Santiago, Chile, September 1994.

[3] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo. "Finding interesting rules from large sets of discovered association rules", In Proc. 3rd Int'l Conf. on Information and Knowledge Management, Gaithersberg, Maryland, pp. 401-408, Nov. 1994.

[4] R. Srikant, R. Agrawal. "Mining Generalized Association Rules", Proc. of the 21st Int'l Conference on Very Large Databases, Zurich, Switzerland, Sep. 1995.

[5] J. Han and Y. Fu. "Discovery of Multiple-Level Association Rules from Large Databases", Proc. of 1995 Int'l Conf. on Very Large Data Bases (VLDB'95), Zurich, Switzerland, pp. 420-431, September, 1995.

[6] M.S. Chen, J. Han and P.S. Yu. "Data Mining: An Overview from a Database Perspective", IEEE Transactions on Knowledge and Data Engineering, 8(6): pp. 866-883, 1996.

[7] D. Cheung, J. Han, V. Ng and C.Y. Wong. "Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique", Proc. of 1996 Int'l Conf. on Data Engineering (ICDE'96), New Orleans, Louisiana, USA, Feb. 1996.

[8] R. Srikant, R. Agrawal. "Mining Quantitative Association Rules in Large Relational Tables", Proc. of the ACM SIGMOD Conference on Management of Data, Montreal, Canada, June 1996. 3.

[9] R. Srikant, Q. Vu, R. Agrawal. "Mining Association Rules with Item Constraints", Proc. of the 3rd Int'l Conference on Knowledge Discovery in Databases and Data Mining, Newport Beach, California, August 1997.