해시테이블을 이용한 속성값 간의 연관관계 추출

양종원[O] 이상희 이동주 양정연 이상구
서울대학교
{zeromus[O], louder81, rocker, jyyang, sglee }@europa.snu.ac.kr

Extraction Association Rule between Attribute Values Using Hash Table

Jongwon Yang[O] Sanghee Lee, Dongjoo Lee, Jungyun yang, Sanggoo Lee
Seoul National University

요 약

전자상거래의 발전은 필연적으로 상품 데이터베이스화를 수반하게 되었다. 이 상품 데이터베이스에 존재하는 각 상품들의 속성값들의 연관관계 추출은 검색ㅡ 유의어 추출 혹은 클러스터링등에 활용될 수 있다. 본 논문에서는 상품 속성값들의 연관관계 추출을 위하여 해쉬 테이블에 기반한 트리 형태 자료구조을 제안한다. 그리고 이 자료구조를 이용하여 상품 데이터베이스의 각 속성값 간의 연관관계를 threshold를 이용하여 선형 시간에 추출하는 알고리즘을 제시한다. 마지막으로, support를 이용하여 트리의 탐색 공간을 줄이는 방식으로 최적화를 시키는 기법을 제시한다.

1. 서 론

전자상거래의 발전은 필연적으로 상품 정보의 데이터베이스화를 수반하게 되었다. 상품 정보의 데이터베이스화를 통해 풍부해진 상품 데이터는 단순한 상품 정보의 추출, 조회뿐만 아니라 통계 정보, 저장되어 있는 데이터 간의 연관 관계를 파악을 통하여 부가적인 정보를 추출하는 것을 가능하게 한다.

상품 정보를 저장하고 있는 데이터베이스는 공통된 속성에 대해 각 상품마다 고유한 속성값을 가진 형식으로 되어있다. 상품 정보 데이터베이스의 예는 [표 1]과 같다.

상품명	사이즈	비율	종류
A	21inch	4:3	비평면
B	18inch	4:3	비평면
C	21inch	16:9	평면
D	30inch	16:9	평면
E	21inch	16:9	평면
F	18inch	4:3	평면
G	30inch	16:9	평면
H	30inch	16:9	평면
I	21inch	4:3	비평면

[표 1] 텔레비전 상품 정보 데이터베이스의 예

[표 1]에 제시된 데이터베이스는 텔레비전 상품 정보 데이터베이스의 예를 든 것으로 상품 A~I는 공통적으로 사이즈, 비율, 종류라는 속성을 가지고 있으며 각 상품마다 서로 다른 속성값을 가지고 있다.

본 연구는 정보통신부의 대학 IT 연구센터 ITRC(Information Technology Research Center)의 지원을 받아 수행되었음

본 논문에서는 어떤 상품에서 속성값 a가 나올 경우 속성값 b가 나오는 경우의 빈도를 a에서 b로의 연관도라고 정의한다. 예를 들어, [표 1]에서 속성 ' 종류' 의 속성값 ' 평면' 이 나온 상품은 모두 6개가 존재하며 이 6개의 상품 중에서 속성 ' 비율' 의 속성값 ' 16:9' 가 나오는 것은 5개가 존재한다. 이 경우 속성값 ' 평면' 에서 속성값 ' 16:9' 의 연관도는 5/6 = 0.83이다.

본 논문에서는 속성값 a에서 속성값 b로의 연관도가 사용자가 미리 지정한 threshold이상일 경우 연관관계가 있다고 정의한다. 이러한 연관관계 정보는 검색이나 유의어 추출에 활용할 수 있다.

본 논문에서는 속성값간의 연관관계를 효율적으로 추출하기 위한 알고리즘과 그 알고리즘에 대한 분석, 그리고 count를 이용한 pruning 방법을 통한 최적화 방안을 제시한다.

2. 관련 연구

속성값의 연관관계를 파악하는 작업의 알고리즘은 일반적인 데이터 마이닝 알고리즘을 참조하였다. 본 논문에서는 FP-tree의 알고리즘[1]과 association rule[2]을 기반으로 하여 pruning을 위하여 succinctness의 개념[3]을 참조하였다.

3. 연관관계 추출 알고리즘

3.1 나이브한 연관도 추출 알고리즘

데이터베이스에서 연관 관계가 있는 속성값을 찾기 위해서는 해당 속성값을 포함하고 있는 속성들에 대한 분석이 필요하다.

속성 A에서 속성 B로의 연관도를 측정할 경우 속성 A를 subject라고 하고 속성 B를 object라고 한다. 예를 들어 [표 1]에서 속성 '종류'의 각 속성값들에 대한 속성 '비율'의 속성값들의 연관도를 측정한다고 할 경우에 속성 '종류'가 subject가 되며 속성 '비율'이 object가 된다.

Subject에 있는 모든 속성값들과 object에 있는 모든 속성값들의 연관도를 측정하기 위해서 나이브하게 subject에 존재하는 모든 속성값들에 대해 object에 있는 모든 속성값들의 연관도를 측정하는 방식을 생각할 수 있다. 모든 상품의 개수를 n개라 할 때 나이브한 속성값들의 연관도 추출 알고리즘은 [알고리즘1]과 같다.

나이브한 연관도 추출 알고리즘

n – 모든 상품의 개수

usr_threshold – 사용자 지정 threshold

1 Subject에 존재하는 각 속성값 S_a에 대해

2 Object에 존재하는 각 속성값 O_b에 대해

3 count ← 0

4 assoc_count ← 0

5 for i← 1 to n

6 if i번째 상품의 subject 속성값 == S_a

7 count++

8 if i번째 상품의 object 속성값 == O_b

9 assoc_count++

10 if count/assoc_count >= usr_threshold

11 Sa와 Ob간에는 연관관계 존재

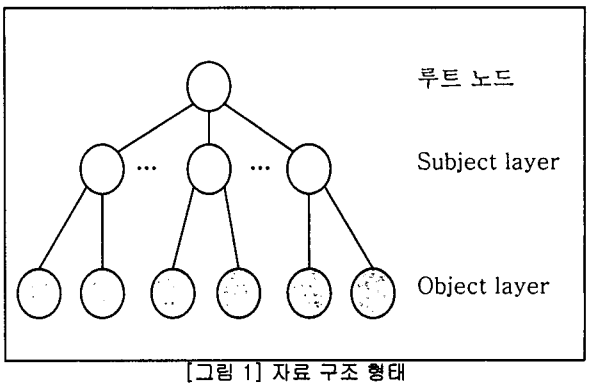
[알고리즘 1] 나이브한 연관도 추출 알고리즘

이 방법은 O(subject의 속성값의 수 * object의 속성값의 수 * n)의 cost를 가지는 알고리즘으로 극단적으로 모든 상품이 서로 다른 subject의 속성값과 object의 속성값을 가질 경우에는 $O(n^3)$의 cost를 보이기 때문에 상품의 수가 많아질수록 수행 시간이 기하급수적으로 증가한다.

3.2 자료구조

본 논문에서는 O(n)의 cost를 가지는 알고리즘을 제시하며 이 알고리즘을 위해 해시테이블에 기반한 트리 형태의 자료 구조를 제시한다. 이 구조는 [그림 1]에 도식화되어 있다. 이 트리의 depth는 2로 depth 1에 존재하는 노드들은 subject의 속성값들을 나타내며 depth 2에 존재하는 노드들은 object의 속성값들을 나타낸다. 본 논문에서는 depth 1에 존재하는 노드들과 depth 2에 존재하는 노드들을 각각 subject layer와 object layer로 부른다.

루트 노드와 depth 1에 존재하는 각 노드들은 해시 테이블로 이루어져 child 노드들은 해시 키에 의하여 parent 노드에 삽입되어 있다. 따라서 child 노드의 각 원 소들을 O(1)에 접근할 수 있으며 새로운 child 노드의 생성 또한 해시테이블에 새로운 값을 삽입하는 작업으로 역시 O(1)이다. 각 노드들은 이외에도 각 속성값의 count를 저장하고 있다.



[그림 1] 자료 구조 형태

3.3 트리 생성 알고리즘

주어진 상품 데이터베이스로부터 연관관계를 추출하기 위해서는 우선 데이터베이스를 읽어나가며 트리를 생성하는 작업이 필요하다. 트리 생성 알고리즘은 [알고리즘 2]와 같다.

트리 생성 알고리즘

n – 모든 상품의 개수

1 for i←1 to n

2 cursor ← 루트 노드

3 cursor.count++

4 if subject layer에 i번째 상품의 subject 속성값을 가진 노드 S가 존재

5 cursor ← S

6 cursor.count++

7 else subject layer에 새로운 노드 S_{new} 생성

8 cursor ← S_{new}

9 cursor.count++;

10 if cursor의 children에서 object layer에 속하여 i번째 상품의 object 속성값을 가진 노드 O가 존재

11 cursor ← O

12 cursor.count++

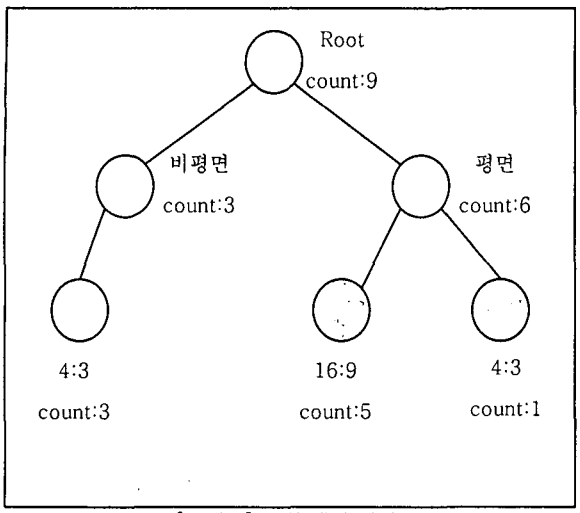
13 else object layer에 S의 child로 새로운 노드 O_{new} 생성

14 cursor ← O_{new}

15 cursor.count++

[알고리즘 2] 트리 생성 알고리즘

트리 생성 알고리즘을 이용하여 [표 1]에 존재하는 데이터베이스를 이용하여 subject가 속성 '종류'이고 object가 속성 '비율'인 트리를 생성한 결과는 [그림 2]와 같다. [그림 2]에서는 각 노드가 가지고 있는 속성값과 count가 표시되어 있다.



[그림 2] 트리 생성 결과

트리 생성 알고리즘은 트리의 노드들이 해시테이블 형태이기 때문에 새로운 child의 생성, child 값의 조회가 모두 O(1)에 수행된다. 따라서 상품 수가 n개인 경우 트리 생성 알고리즘의 cost는 O(n)이 된다.

3.4 트리 탐색을 통한 연관관계 추출

트리 생성 알고리즘으로 생성된 트리를 이용하여 연관관계를 추출하기 위해서는 트리의 탐색이 필요하다. 연관 관계 탐색은 subject layer의 각 노드의 count를 object layer에 속한 각 child의 count로 나눈 결과가 사용자가 미리 지정한 threshold 이상일 경우 연관관계가 있는 것으로 판정되는 형식으로 이루어지며 그 알고리즘은 [알고리즘 3]과 같다.

```
트리 탐색 알고리즘
n – 모든 상품의 개수
usr_ threshold – 사용자 지정 threshold
1 트리에 존재하는 subject layer의 각 노드 Sa에 대해
2   S의 child인 object layer의 각 노드 Ob에 대해
3     if Ob.count/Sa.count >= usr_ threshold
4       Sa와 Ob는 연관관계가 존재
```

[알고리즘 3] 트리 생성 알고리즘

[그림 2]에 제시된 트리에서 사용자 지정 threshold를 0.70으로 잡은 경우 비평면→4:3(연관도: 1.00), 평면→16:9(연관도:0.83)가 연관관계가 있다고 판정된다.
위 알고리즘에서 연산이 이루어지는 회수는 직관적으로 트리에 존재하는 object layer의 전체 노드 수에 선형 비례한다. 따라서 트리 탐색 알고리즘의 cost는 O(n)이 된다.

4. Support를 이용한 최적화

만약 속성값 a와 b가 전체 데이터베이스에 단 한번씩 나오며 동일한 상품에 나올 경우 이 상품간의 연관도는 100퍼센트가 되겠지만 지나치게 적은 count때문에 검색이나 유의어 추출에 유의미한 결과를 낸다고 말할 수 없다. 이 때문에 유의미한 결과만을 추출하기 위해서는 count가 일정 기준(minimum support)를 넘긴 경우에만 연관관계에 반영할 필요가 있다.

속성값들 간의 연관 관계를 추출하기 위한 트리 구조에서 parent 노드의 count값은 무조건 child 노드의 count보다 크거나 같은 성질을 가진다는 것은 직관적으로 파악할 수 있다. 이 성질을 이용하여 minimum support를 넘지 못하는 subject 노드에 대해서는 하부 구조를 탐색할 필요가 없다는 사실을 알 수 있다. 즉, 트리 탐색 시 count가 minimum support를 넘지 못하는 subject 노드는 pruning하여 탐색 공간을 줄여 수행 시간을 최적화 할 수 있다.

5. 결론과 향후 과제

본 논문에서는 상품 데이터베이스에서 연관관계를 가지는 속성값을 O(n)에 추출하는 자료 구조와 알고리즘을 제시하였다. 이 알고리즘은 속성값이 아닌 속성값을 파싱한 어휘를 이용한 적용도 가능하다.
여기서 제시한 알고리즘을 통하여 두 속성간의 연관관계가 있는 속성값 추출은 선형 시간에 가능하게 되었으나 여러 개의 속성이 존재하는 경우 속성의 수를 m이라 하였을 때 $m*(m-1)/2$번의 알고리즘 수행이 필요한 문제가 존재한다. 즉, m개의 속성을 가진 경우 전체 cost는 $O(nm^2)$이 된다. 이 경우 여러 개의 tree를 동시에 생성하여 메모리와 시간을 tradeoff하는 방법을 생각하거나 3개 이상의 속성을 동시에 비교하는 방법에 대한 연구가 앞으로 수행될 필요가 있다.

6. 참고 문헌

[1] J.Han, J.Pei, and Y.Yin, " Mining Frequent Patterns without Candidate Generation" , ACM SIGMOD 2000
[2] R.Agrawal and R.Srikant, " Fast Algorithms for Mining Association Rules" , VLDB 1994
[3] Ng, Lakshmanan, Han, Pang, " Exploratory Mining and Pruning Optimizations of Constrained Association Rules" , ACM Sigmod 1998
[4] H.Cormen, Charles E.Leiserson, Ronald L.Rivest, Clifford Stein, Introduction to Algorithms, The MIT Press