XML 문서 클러스터링을 이용한 개선된 연관규칙

김의찬° 이재민 황병연
가톨릭대학교 컴퓨터공학과
{eckim°, likedawn, byhwang}@catholic.ac.kr

Advanced Association Rules using XML Document Clustering

Euichan Kim° Jaemin Lee Byungyeon Hwang
Dept. of Computer Engineering, The Catholic University of Korea

요 약

기존의 연관규칙을 생성하는 알고리즘의 문제점을 개선하기 위해 본 논문에서는 XML 문서 클러스터링을 이용하였다. XML 문서 클러스터링을 이용하여 데이터베이스 탐색 횟수 및 조인 개수를 줄여서 수행 속도를 향상시키고, 또한 클러스터링을 통해 얻은 클러스터에서 규칙을 찾기 때문에 기존의 연관규칙 생성 방법에서는 찾지 못했던 규칙들도 찾아낼 수 있다. 본 논문에서 사용하는 클러스터링 방법은 XML문서 검색을 위한 3차원 비트맵 인덱싱인 xPlaneb를 사용하여 구현하였다.

1. 서 론

컴퓨터의 계속적인 발전과 데이터베이스 시스템 사용의 증가로 인해 데이터베이스에 저장되는 데이터의 양이 무수히 늘어나고 있다. 그러나 이렇게 많은 양의 데이터에서 얻어낼 수 있는 정보는 일반적인 정보이다. 이러한 일반적인 정보 외에 데이터베이스에 저장되어 있는 데이터만으로는 알 수 없는 정보를 찾아내는 방법이 필요하다. 그것이 바로 데이터 마이닝(Data Mining) 방법이다. 데이터 마이닝 방법은 많은 양의 데이터에 함축적으로 들어 있는 지식이나 패턴을 찾아내는 기술이며, 현재 많은 기법들이 연구되고 있다. 데이터 마이닝 기법으로는 연관규칙(Association Rules), 분류(Classification), 클러스터링(Clustering), 의사결정(Decision Making) 등이 있다[1]. 이러한 기법들 중 본 논문에서는 클러스터링과 연관규칙을 다루려 한다.

클러스터링은 주어진 객체 중에 유사한 것들을 몇몇 집단으로 그룹화하여 각 집단의 성격을 파악하는데 사용되는 기법이다[2]. 이러한 클러스터링은 통계학(Statistics), 기계 학습(Machine Learning), 패턴 인식(Pattern Recognition), 이미지 처리(Image Processing) 등 많은 분야에서 지속적으로 연구되고 있다. 연관규칙은 하나의 거래나 사건에 포함되어 있는 아이템들의 상호 연관성을 발견하는 것이다[3]. 이 때 연관성은 어떤 아이템 집합의 존재가 다른 아이템 집합의 존재를 암시하는 것을 의미하며, 다음과 같이 표시할 수 있다.

$$A \rightarrow B$$

이는 "만일 A가 발생한다면 B도 발생한다."라는 의미를 가지고 있다. 이러한 연관규칙을 사용하는 예로는 함께 구매하는 상품의 조합이나 서비스 패턴을 발견할 때 종종 사용된다.

기존의 연구들에서는 거리기반이나 밀도기반 등으로 클러스터링을 하였으며, 거리나 밀도가 아닌 연관규칙을 통해서 클러스터링 하는 연구도 있다[4, 5].

본 논문에서는 기존의 연구인 클러스터링 기법에 연관규칙을 적용하는 것이 아니라, 연관규칙을 생성하는데 클러스터링을 이용하는 방법을 제안하려 한다.

기존의 연관규칙의 문제점이라 할 수 있는 것은 데이터베이스 접근 횟수가 많고, 트랜잭션의 수가 많을수록 조인의 횟수도 많아서 성능에 많은 문제가 있었다. 이러한 성능에 대한 문제점을 해결하기 위한 방법도 연구되고 있다[6]. 본 논문에서는 이러한 문제를 해결하기 위한 방법으로 클러스터링을 이용하려 한다.

클러스터링을 하여 몇 개의 클러스터들을 생성하고, 각각의 클러스터에 연관규칙 생성 방법을 적용하는 것이다. 이러한 방법을 이용한다면 다음과 같은 장점이 있다. 첫째로, 연관규칙 생성 수행 속도는 당연히 기존의 알고리즘보다 빠를 것이다. 둘째로, 기존의 알고리즘의 경우에는 모든 데이터베이스 트랜잭션을 다루어 연관규칙을 생성하는 것이므로, 같은 항목을 가지고 있는 소수의 트랜잭션은 무시될 수 있는 문제가 있지만, 제안한 방법을 이용한다면 클러스터링을 통해서 기존의 방법에서 찾지 못했던 연관규칙도 찾아 낼 수 있을 것이다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 방법에 대해 간단히 살펴보고, 3장에서는 본 논문에서 제안하는 방법에 대해 기술한다. 4장에서는 실험예제에 대한 분석, 5장에서는 결론 및 향후 연구과제에 대해 논의하도록 하겠다.

2. 관련 연구

연관규칙은 대용량 데이터베이스 내의 단위 트랜잭션에서 빈번하게 발생하는 사건의 유형을 발견하는 것이다[7]. 예를 들어, "전체 고객 중에 빵과 우유, 그리고 주스를 구매한 고객이 10% 이상이고, '빵과 우유' 를 구매한 고객의 50%가 주스도 함께 구매한다." 이것이 하나의 발견된 사건의 유형이다. 여기서 10%는 연관규칙의 지지도(support)가 되고, 50%는 신뢰도(confidence)가 된다.

연관규칙을 찾는 전체 과정은 전체 데이터베이스에서 후보 아이템 항목 집합을 찾고, 후보 아이템 항목 집합에서 최소 지지도 값을 넘는 빈발 항목 집합을 찾아낸다. 다음으로 빈발 항목 집합에서 최소 신뢰도 값을 넘는 아이템 집합으로 연관규칙을 찾아내게 되는 것이다. 여기서 지지도(S)란, 전체 사건 또는 거래 중에서 어떤 아이템 X와 아이템 Y를 동시에 포함하는 사건 또는 거래가 어느 정도 되는가 하는 것이다. 이것을 식으로 표현하면 다음과 같다.

$$S = \frac{|X \cap Y|}{N} \text{ (N은 전체 트랜잭션의 개수)}$$

그리고, X→Y는 지지도 S를 갖는다고 말한다. 신뢰도(C)는 어떤 아이템 X를 포함하는 사건이나 거래 중에서 Y가 포함된 사

건이나 거래가 어느 정도인가 하는 것이다. 이것을 식으로 표현하면 다음과 같다.

$$C = \frac{|X \cap Y|}{|X|}$$

그리고, X→Y는 신뢰도 C를 갖는다고 말한다. 신뢰도를 통해서 최종 연관규칙을 얻어낼 수 있다.

기존의 연관규칙 생성방법으로 [7]에서 제안된 Apriori 알고리즘이 있는데, 이는 연관규칙을 생성하는 대표적인 알고리즘이다. 이 알고리즘은 조인(join) 단계와 가지치기(prune) 단계로 나누어진다. 앞서 살펴본 지지도 계산에서 각 항목의 발생빈도수를 세어 빈발 항목 집합을 찾아내는데, 전체 트랜잭션을 검색하기 때문에 그만큼 수행속도는 느려지게 된다. 또한, 후보항목이 많을수록 조인단계에서 많은 조인을 필요로 하므로 성능은 떨어지게 된다. 본 논문에서는 이러한 문제를 해결하기 위한 방법으로 데이터베이스에 있는 트랜잭션들을 먼저 클러스터링 하고자 하는 것이다. 클러스터링을 통해 나온 클러스터에서 연관규칙을 찾게 되면 조인의 개수도 적어질 뿐만 아니라 전체 트랜잭션보다 더 적은 수의 트랜잭션을 검색하기 때문에 기존의 방법보다 성능은 나아지게 될 것이다.

3. 클러스터링을 이용한 연관규칙

본 논문에서는 클러스터링을 이용하여 연관규칙을 생성하는 방법을 제안한다. 제안한 내용을 설명하기 전에, 2가지 가정을 하도록 한다. 첫 번째로 아이템의 개수는 정해져 있으며, 두 번째로 각 트랜잭션에 속해 있는 아이템의 유무를 0 또는 1로 표현한다는 것이다. 예를 들어, 트랜잭션이 2개가 있고 아이템은 각각 T1={a, b, c, e}, T2={a, c, f}가 있다고 가정하면, 그림 1과 같이 표현하는 것이다.

트랜잭션 Item	a, b, c, d, e, f
T1 = {a, b, c, e}	→ (1, 1, 1, 0, 1, 0)
T2 = {a, c, f}	→ (1, 0, 1, 0, 0, 1)

그림 1 아이템의 유무를 0 또는 1로 표현

이러한 정보를 이용하여 BitCube[8]에서 사용한 유사도를 기반으로 클러스터링을 한다. 유사도($Sim(d_i, d_j)$)의 정의는 다음과 같다. xOR(d_i,d_j)는 그림 1과 같이 표현된 두 개 문서(d_i, d_j)의 비트들을 xOR 계산하여 나온 1의 개수를 의미하고 MAX([d_i],[d_j])는 두 개의 문서들 중 비트개수가 최대인 값을 의미한다.

$$Sim(d_i, d_j) = 1 - \frac{xOR(d_i, d_j)}{\mathrm{MAX}([d_i],[d_j])}$$

유사도 측정을 하여 클러스터링을 하는 부분은 xPlaneb라는 유사한 XML 문서를 클러스터링하기 위한 인덱싱을 이용하였다 [9]. xPlaneb는 유사한 구조를 갖는 XML 문서들을 문서와 경로 그리고 단어를 축으로 하는 3차원 인덱스로 구성한다. xPlaneb는 XML 문서가 동일한 경로를 얼마나 포함하는지의 정도로 유사도를 결정하는데 위에 언급한 유사도 정의를 사용한다. xPlaneb는 Bit-wise연산이 가능한 3차원 인덱스를 통해 XML 문서들의 유사도를 측정하고 XML 문서들을 클러스터링한다.

본 논문에서의 적용방법은 xPlaneb에서 하나의 문서를 하나의 트랜잭션으로 보고, 각 문서를 구성하는 경로를 트랜잭션을 구성하는 각각의 아이템들로 보았으며, 단어는 고려하지 않았다. 식을 좀 더 살펴보면, 유사도 식에서 d_i, d_j는 각각 XML 문서인데, 이것을 각각 트랜잭션 t_i, t_j로 보고 분모에 있는 MAX([d_i],[d_j])는 앞서 가정하였듯이 아이템의 개수는 정해져 있기 때문에 아이템의 총 개수로 표현할 수 있다. 그러면 다음과 같은 식으로 정의할 수 있게 된다.

$$Sim(t_i, t_j) = 1 - \frac{xOR(t_i, t_j)}{Number\ of\ Total\ Item}$$

이러한 식을 통해서 클러스터링된 각 클러스터들은 데이터베이스에 저장되게 되고, 각 클러스터들에 대해서 연관규칙을 적용하는 것이다.

본 논문에서 제안하는 모듈의 전체적인 구조는 그림 2와 같다.

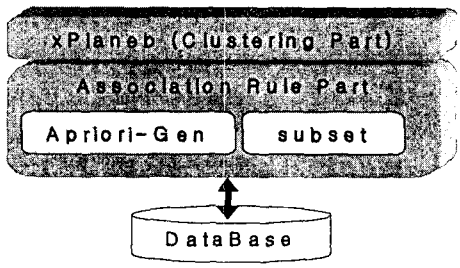

그림 2 클러스터링을 이용한 연관규칙 모듈 구조

각 클러스터들에 연관규칙을 적용하는 알고리즘은 그림 3과 같다. 입력으로 최소지지도 값과 클러스터링된 트랜잭션 데이터 즉, 유사한 몇 개의 트랜잭션이 포함되어 있는 하나의 클러스터를 사용하고, 출력으로 입력된 클러스터에 대한 최종 빈발 항목 집합이 나오게 된다.

```
입력: 최소지지도(min_sup), 클러스터링된 트랜잭션 데이터(CluTran)
출력: 클러스터링된 트랜잭션에서 나온 빈발항목 집합(Li)
방법:
L1=1-빈발항목;
for(k=2; Lk-1 ≠ Ø; k++)
{
  Ck = Apriori_Gen(Lk-1, min_sup);
  // Apriori_Gen 함수로부터 후보항목집합 생성
  for each t ∈ CluTran {
  Ct = subset(Ck, t);
  //CluTran에 포함되어있는 트랜잭션 t들 중 빈발 집합만 찾아냄
    for each c ∈ Ct
    c.count++; // Ct에 포함되어있는 후보항목 c에 대한 개수를 셈
  }
  Lk = { c ∈ Ct | c.count ≥ min_sup};
}
return UkLk;
```

그림 3 클러스터에 대한 연관규칙 적용 알고리즘

일반적으로 연관규칙을 생성하는데 있어서의 성능은 빈발 항

목 집합을 구하는 부분이 전체 성능을 좌우하기 때문에 신뢰도를 바탕으로 최종 규칙을 생성하는 부분은 생략하였다.

4. 실험 예제

표 1과 같은 데이터베이스 예제를 살펴보자. 표 1에 나타나 있는 예제 데이터베이스의 트랜잭션은 모두 6개이고 아이템 항목은 모두 9개이다. 이것을 각 트랜잭션의 아이템 집합에 대해서 유, 무를 결정하여 0 또는 1로 표시하면 그림 4와 같이 나오게 된다.

표 1 예제 데이터베이스

TID	Item Set
T1	a, b, c
T2	a, b, c, d
T3	a, b, d, e
T4	a, b, f
T5	d, h
T6	d, g, h, i

그림 4와 같이 표현된 데이터를 바탕으로 앞서 설명하였던 유사도 식을 바탕으로 클러스터링을 하게 되면 {T1, T2, T3, T4}와 {T5, T6}으로 나누어지게 된다. 여기서 유사도 값의 임계값은 0.7로 가정하였다.

```
        a, b, c, d, e, f, g, h, i
T1 = (1, 1, 1, 0, 0, 0, 0, 0, 0)
T2 = (1, 1, 1, 1, 0, 0, 0, 0, 0)
T3 = (1, 1, 0, 1, 1, 0, 0, 0, 0)
T4 = (1, 1, 0, 0, 0, 1, 0, 0, 0)
T5 = (0, 0, 0, 1, 0, 0, 0, 1, 0)
T6 = (0, 0, 0, 1, 0, 0, 1, 1, 1)
```

그림 4 좌표로 표현

2개의 클러스터 중에 첫 번째 클러스터에 대하여 빈발 항목을 구하게 되면 그림 5와 같이 나오게 된다. 최소지지도는 50%로 한다.

Item Set	지지도
{a}	100%
{b}	100%
{c}	50%
{d}	50%

L1

Item Set	지지도
{a, b}	100%
{a, c}	100%

L2

Item Set	지지도
{a, b, c}	100%

L3

그림 5 빈발 항목 집합

L3에 나온 빈발 항목 집합을 바탕으로 신뢰도에 따른 연관규칙을 생성하게 된다. 그리고 두 번째 클러스터에서는 빈발 항목 집합이 {d, h}가 나오게 된다.

기존의 연관규칙 알고리즘을 통해서 얻을 수 있는 규칙으로는 첫 번째 클러스터에서 얻어낸 규칙을 찾을 수 있겠지만 두 번째 클러스터에서 얻어낸 규칙은 찾지 못한다. 따라서 본 논문에서 제안한 방법으로 버려질 수 있는 규칙도 찾을 수 있는 장점을 가지고 있다.

5. 결론 및 향후 연구

본 논문에서는 기존의 연관규칙 방법보다 좀 더 개선된 방법을 제시하였다. 기존의 연관규칙 방법은 많은 트랜잭션 검색과 많은 조인으로 인해 성능 저하가 나타났는데 이를 개선하기 위하여 본 논문에서는 클러스터링 방법을 이용하였다. 클러스터링을 먼저 하여 유사도가 비슷한 것끼리 묶어낸 클러스터를 찾은 다음 각 클러스터에 대한 연관규칙을 찾아내는 것인데 클러스터링을 하기 위해 xPlaneb를 이용하였다.

본 논문에서 제안한 방법을 이용하면 보다 나은 성능으로 연관규칙을 생성해 낼 수 있으며, 기존의 방법으로 찾지 못했던 규칙도 찾아낼 수 있게 된다.

추후 연구 방향으로 실제 데이터를 바탕으로 실험을 하고 분석을 하며, 다양한 클러스터링 방법을 적용하여 연관규칙을 생성하는 연구를 할 것이다.

[참고문헌]

[1] M.S. Chen, J. Han, and P.S. Yu, "Data Mining: An Overview from a Database Perspective," IEEE Transaction on Knowledge and Data Engineering, Vol. 8, No. 6, pp. 866-883, Dec, 1996.

[2] H. Wang, W. Wang, J. Yang, and P.S. Yu, "Clustering by Pattern Similarity in Large Data Sets," Proceedings of ACM SIGMOD, Wisconsin, pp. 394-405, June, 2002.

[3] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Database," Proceedings of ACM SIGMOD, Washington DC, pp. 207-216, May, 1993.

[4] E.H. Han, G. Karypis, V. Kumar, and B. Mobasher, "Clustering Based On Association Rule Hypergraphs," Workshop on Research Issues on Data Mining and Knowledge Discovery, Tucson, Arizona, 1997.

[5] W.A. Kosters, E. Marchiori, and A.J. Oerlemans, "Mining Clusters with Association Rules," Proceedings of Intelligent Data Analysis, Amsterdam, The Netherlands, pp. 39-50, 1999.

[6] J.S. Park, M. Chen, and P.S. Yu, "An Effective Hash-Based Algorithm for Mining Association Rules," Proceedings of ACM SIGMOD, San Jose, pp. 175-186, June, 1995.

[7] R. Agrawal and R. Srikant, "Fast Algorithm for Mining Association Rules in Large Databases," Proceeding of the 20th International Conference on Very Large Databases, Santiago de Chile, pp. 487-499, 1994.

[8] J. Yoon, V. Raghavan, and V. Chakilam, "BitCube: Clustering and Statistical Analysis for XML Documents," 13th International Conference on Scientific and Statistical Database Management, Virginia, July, 2001.

[9] 이재민, 황병연, "xPlaneb: XML 문서 검색을 위한 3차원 비트맵 인덱스," 정보과학회논문지, 제31권, 제3호, pp. 331-339, 2004년 6월.