트랜잭션 클러스터링을 이용한 연관규칙 생성

김의찬, 황병연
가톨릭대학교 컴퓨터공학과
e-mail: {eckim, byhwang}@catholic.ac.kr

Creation of Association Rules using Transaction Clustering

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

요 약

데이터베이스로부터 유용한 정보를 얻기 위해서 데이터마이닝을 사용하는데 많은 데이터들을 다루기 위해서는 좀 더 나은 성능의 데이터마이닝 기법이 필요하다. 연관규칙을 생성하는 기존의 Apriori 알고리즘은 많은 데이터베이스 접근과 많은 조인 횟수로 인하여 수행 속도의 저하를 가져오게 된다. 이를 개선하기 위하여 본 논문에서는 새로운 클러스터링 방법을 이용하여 클러스터링을 수행하고 각 클러스터의 연관규칙을 생성하게 된다. 본 연구의 방법을 이용하게 되면 기존 연관규칙 알고리즘으로 찾지 못했던 규칙도 생성가능하다.

1. 서론

데이터베이스에는 많은 데이터들을 가지고 있으며, 계속적으로 그 양은 늘어나고 있다. 이렇게 많은 양의 데이터에서 얻어낼 수 있는 정보는 검색을 통해 바로 확인할 수 있는 기본적이고 일반적인 정보들이다. 데이터 마이닝(Data Mining)은 이런 기본적이고 일반적인 정보들로부터 함축하거나 암시하고 있는 정보들을 추출해내는 기법이다. 데이터베이스에 저장되어 있는 수많은 데이터들로부터 기존의 단순한 검색으로부터 얻어내지 못하는 그런 정보들을 찾아내는 것이 데이터 마이닝인 것이다. 현재 연구되는 기법들은 연관규칙(Association Rules), 분류(Classification), 일반화(Generalization), 클러스터링(Clustering) 등 다양하다[1]. 이들 중에서 본 논문에서는 클러스터링과 연관규칙을 혼합한 방법을 제시하려 한다.

클러스터링은 주어진 객체들 중에서 유사한 객체들을 몇 개의 집합으로 그룹화하여 그 그룹의 특징이나 성격을 파악하는 방법이다[2]. 클러스터링의 종류는 다양하다. 계층적 클러스터링(Hierarchical Clustering), 분할 클러스터링(Partitional Clustering), 최근접 이웃 클러스터링(Nearest Neighbor Clustering) 등이 있다[3]. 그러나 여기서 주시해야 되는 부분은 유사도 측정을 어떠한 방법으로 하여 클러스터링 하느냐하는 것이다. 기존의 연구들에서는 거리기반이나 밀도기반의 유사도를 측정하여 클러스터링을 하거나, 연관규칙을 유사도 측정 방법으로 이용하여 클러스터링을 한 연구도 있다[4, 5]. 기본적인 유사도 측정방법은 거리기반의 측정방법인데 거리기반 측정방법의 대표적인 방법으로 유클리디언 거리법을 많이 사용한다. 그러나 본 연구에서 다루어지는 트랜잭션의 경우 거리기반으로 하는 유클리디언 거리법을 이용하는 것은 적절한 방법이 아니다. 따라서 새로운 클러스터링 방법을 이용하여 클러스터링하려 한다.

본 논문에서 찾고자 하는 것은 연관규칙이다. 연관규칙은 하나의 거래나 사건에 포함되어 있는 아이템들 간의 상호 연관성을 찾는 것이다[6]. 여기서 연관성이라는 것은 어떤 아이템이나 아이템 집합의 존재가 다른 아이템이나 아이템 집합의 존재를 암시한다는 것을 의미하며, 다음과 같이 표현할 수 있다.

$$X \rightarrow Y$$

이것은 "만일 X가 발생한다면 Y도 발생한다."라는 의미를 가지고 있다. 다시 말하면, "아이템 X를 소유한다면 아이템 Y를 소유할 가능성이 있다"라고 할 수 있는 것이다. 이러한 연관규칙을 사용하는 예로는 함께 구매하는 상품의 조합이나 서비스 패턴을 발견할 때 종종 사용된다.

본 논문에서는 연관규칙을 생성하는데 있어서 기존에 문제가 되었던 수행성능 부분을 클러스터링을 이용하여 해결하려 한다. 클러스터링도 기존의 클러스터링을 이용하는 것이 아니라 본 논문에서 제안하는 클러스터링을 이용하여 적용하려고 한다.

기존의 연관규칙의 문제점이라 함은 데이터베이스 접근 횟수가 많고, 트랜잭션의 수가 많을수록 조인의 횟수도 많아져서 처리하는 수행속도 및 성능에 많은 문제가 있었다. 이러한 성능에 대한 문제점을 해결하기 위한 방법도 계속 연구되고 있다. 본 논문에서는 이러한 문제를 해결하기 위한 방법으로 클러스터링을 이용하려 한다.

본 논문에서 제시한 클러스터링을 이용하여 트랜잭션들을 몇 개의 클러스터들로 생성하고, 각각의 트랜잭션 클러스터에 연관규칙 생성 방법을 적용하는 것이다. 이러한 방법을 이용한다면 다음과 같은 장점을 갖는다. 첫째, 데이터베이스 접근 횟수 및 조인 횟수의 감소가 있기 때문에 연관규칙 생성 수행속도는 당연히 기존의 알고리즘보다 빠를 것이다. 둘째, 기존의 알고리즘을 이용하는 경우에는 모든 데이터베이스 트랜잭션을 통해 연관규칙을 생성하는 것인데 이러한 경우에 최소 지지도를 통해 무시될 수 있는 트랜잭션들이 발생하게 된다. 그러나 본 논문에서 제안한 방법을 이용한다면 기존의 방법에서 찾아내지 못했던 연관규칙도 찾아 낼 수 있게 된다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 연관규칙 방법에 대해 간략히 살펴보고, 3장에서는 본 논문에서 제안하는 트랜잭션 클러스터링 방법과 이를 이용하여 연관규칙을 생성하는 방법에 대해 기술한다. 4장에서는 실험예제를 통해 좀 더 자세히 살펴보고, 마지막으로 5장에서는 결론 및 향후 연구 방향에 대해 기술한다.

2. 관련 연구

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

연관규칙을 찾는 전체 과정을 간단하게 살펴보면, 전체 데이터베이스에서 먼저 후보 아이템 항목 집합을 찾고, 이 후보 아이템 항목 집합에서 미리 제시된 최소 지지도 값을 넘는 빈발 항목 집합을 찾아낸다. 빈발 항목 집합을 찾을 때 조인연산을 반복해서 사용하게 된다. 최종적으로 나오는 아이템 집합에서 최소 신뢰도 값을 넘는 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 신뢰도(C)는 어떤 아이템 X를 포함하는 사건이나 거래 중에서 Y가 포함된 사건이나 거래가 어느 정도인가 하는 것이다. 이것을 식으로 표현하면 다음과 같다.

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

지지도를 통해 나온 빈발항목에서 신뢰도를 통해 최종 연관규칙을 얻어내는 것이다.

대표적인 연관규칙 알고리즘으로는 [7]에서 제안한 Apriori 알고리즘이 있다. 이 알고리즘은 조인(join) 단계와 가지치기(prune) 단계로 나누어지는데 앞서 살펴본 지지도 계산에서 각 항목의 발생 빈도수를 세어 빈발 항목 집합을 찾아내게 된다. 이 과정에서 데이터베이스의 전체 트랜잭션을 검색해야하기 때문에 그만큼 수행속도는 느려지게 된다. 또한, 후보항목이 많을수록 조인단계에서 많은 조인을 필요로 하므로 성능은 떨어지게 된다. 본 논문에서는 이러한 문제점을 해결하기 위한 방법으로 데이터베이스에 있는 트랜잭션들을 먼저 클러스터링 하고 클러스터링을 통해 나온 클러스터에서 연관규칙을 찾게 되면 조인의 개수도 적어지고 전체 트랜잭션보다 클러스터 내에 있는 더 적은 수의 트랜잭션을 검색하기 때문에 기존의 방법보다 성능은 나아지게 될 것이다.

3. 트랜잭션 클러스터링과 연관규칙

본 논문에서는 기존의 거리 기반의 클러스터링 방법을 이용하지 않고 아이템들 간의 유사도 측정에 적합한 트랜잭션 클러스터링 기법을 제안하여 사용하려 한다.

3.1 트랜잭션 클러스터링

본 논문에서 제안하는 클러스터링 방법은 각 트랜잭션에 포함되어 있는 아이템들을 그대로 이용하는 것이 아니라 아이템의 유무에 따라서 0과 1로 표현을 하여 사용한다[8, 9]. 즉, 비트형태로 변환하여 사용하는 것이다. 다음으로 각 트랜잭션들의 유사도 값을 구하기 위하여 다음과 같은 유사도 식을 사용한다.

$$t\_sim(t_i, t_j) = 1 - \frac{|xOR(t_i, t_j)|}{\text{total of items}}$$

여기서, t_sim(t_i, t_j) 함수는 각 트랜잭션들 간의 유사도 값을 구하는 함수이며, t_i, t_j는 트랜잭션이 된다. 다음으로 xOR 연산을 이용하여 두 트랜잭션의 차이를 구한다. |xOR(t_i, t_j)|의 의미는 두 트랜잭션이 서로 다른 아이템을 가지고 있는 개수가 몇 개가 되는지 나타내는 것이다. 그것을 아이템의 총 개수로 나누게 된다. 다음으로 각 클러스터의 대푯값을 구하기 위하여 사용되는 식은 다음과 같다.

$$pos(i_i) = \frac{|i_i|}{[i_i]}$$

여기서, pos(i_i) 함수는 각 아이템들의 소유가능성을 구하는 함수이며, i_i는 아이템이 된다. |i_i|의 의미는 아이템을 소유하고 있는 즉, 1의 개수가 된다. [i_i]의 의미는 아이템을 소유하고 있거나, 소유하고 있는 않는 즉, 1과 0의 개수가 된다.

다음으로 위 pos(i_i)식을 통해 대푯값을 구하는 방법을 설명한다. 클러스터링을 하기 위해서 유사도 임계값(sn)과 소유 가능성 임계값(pn)이 필요하다. 대푯값을 구할 때 소유 가능성 임계값을 사용하게 되는데 먼저 몇 가지를 정의를 바탕으로 클러스터의 대푯값을 찾아내게 된다.

정의 1. pos(i_i) >= pn 이 식을 만족한다면 이 아이템은 possessive item이라 한다.

정의 2. pos(i_i) <= 1-n 이 식을 만족한다면 이 아이템은 약한 소유 가능성을 갖는 아이템이라 한다.

정의 3. 1-n < pos(i_i) < n 이 식을 만족한다면 이 아이템은 강한 소유 가능성을 갖는 아이템이라 한다.

정의 4. 해당 아이템이 possessive item이거나 강한 소유 가능성을 갖는 아이템인 경우에는 1, 약한 소유 가능성을 갖는 아이템인 경우에는 0으로 간주한다.

3.2 연관규칙

본 논문에서 사용하는 알고리즘은 기존의 연관규칙 알고리즘인 Apriori 알고리즘을 기반으로 한다. Apriori 알고리즘과의 차이는 Apriori 알고리즘을 사용할 때 전체 데이터베이스를 스캔하게 되지만 본 연구에서는 클러스터링된 클러스터의 트랜잭션들만 스캔하는 것이 다르다.

4. 실험 예제

표 1과 같은 데이터베이스가 있다고 가정하자.

표 1 예제 데이터베이스

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

표 1의 아이템들을 유무에 따라 1또는 0, 즉 비트로 표시하면 표 2와 같이 바꿀 수 있다.

표 2 비트로 변환한 데이터베이스

TID	a	b	c	d	e	f	g	h	i	j
T1	1	1	1	0	0	1	0	0	0	0
T2	1	1	0	1	1	0	0	0	0	0
T3	1	1	1	1	0	1	1	0	0	0
T4	1	1	0	1	0	1	0	0	0	0
T5	0	0	0	0	0	0	1	1	1	1
T6	0	0	0	1	1	0	1	1	0	1

이제 유사도 임계값을 0.6, 소유 가능성 임계값을 0.6으로 정한다. T1과 T2의 유사도 값을 구하면 0.6이 나오므로 T1과 T2는 첫 번째 클러스터로 묶인다. 다음으로 첫 번째 클러스터의 대푯값을 구한다. 각 아이템의 소유 가능성 값을 구하면, a=1, b=1, c=0.5, d=0.5, e=0.5, f=0.5, g=0, h=0, i=0, j=0이 나온다. 앞서 얘기했듯이 a, b는 possessive item이 되고, c, d, e, f는 강한 소유가능성 아이템, g, h, i, j는 약한 소유 가능성 아이템이 된다. 그러므로 첫 번째

클러스터의 대푯값은 표3과 같이 나오게 된다.

표 3 첫 번째 클러스터의 대푯값

TID	a	b	c	d	e	f	g	h	i	j
T1	1	1	1	0	0	1	0	0	0	0
T2	1	1	0	1	1	0	0	0	0	0
C1	1	1	1	1	1	1	0	0	0	0

위와 같은 과정을 반복해서 진행하여 클러스터링을 하면 {T1, T2, T3, T4} 클러스터와 {T5, T6} 클러스터가 생성된다.

생성된 클러스터들을 이용하여 연관규칙을 생성하게 되는데 첫 번째 클러스터에 대한 연관규칙을 생성해 보자. 최소 지지도는 60%로 한다. 그러면 그림 1과 같은 결과가 나오게 된다.

Item Set	지지도
{a}	100%
{b}	100%
{d}	75%
{f}	75%

L1

Item Set	지지도
{a, b}	100%
{a, d}	75%
{a, f}	75%
{b, d}	75%
{b, f}	75%

L2

Item Set	지지도
{a, b, d}	75%
{a, b, f}	75%

L3

그림 1 빈발 항목 집합

그림 1에 나온 빈발 항목 집합을 바탕으로 최종 연관규칙을 생성하게 되는 것이다. 또한 두 번째 클러스터를 이용하여 연관규칙을 생성하면 기존의 방법으로 찾을 수 없었던 규칙도 발견하게 된다.

5. 결론 및 향후 연구

본 논문에서는 기존의 연관규칙 방법의 단점이었던 전체 데이터베이스 트랜잭션 검색과 많은 조인으로 인한 성능 저하를 개선하기 위해 클러스터링을 이용하여 연관규칙을 생성하는 방법을 제시하였다. 또한, 클러스터링은 거리기반이 아닌 비트를 이용한 클러스터링 방법을 제안하였다.

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

추후 연구 방향으로 실험과 분석을 할 예정이며, 본 연구에 아이템의 횟수 및 순서와 가중치를 적용하는 문제도 연구 중에 있다.

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] A.K. Jain, M.N. Murty, and P.J. Flynn, "Data Clustering: A Review," ACM Computing Surveys, Vol. 31, No. 3, pp. 264-323, 1999.

[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] 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.

[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] 김의찬, 이재민, 황병연, "XML 문서 클러스터링을 이용한 개선된 연관규칙," 제 32회 한국정보과학회 추계학술대회논문집, 2004년 10월.