테이블 형식의 데이터베이스에 대한 규칙의 효율적 발견

석 현 태°
동서대학교

An Efficient Discovery of Rules for Database Table

Sug Hyontai
Dongseo University
E-mail : sht@dongseo.ac.kr

요 약

데이터마이닝 작업의 대표적 방법 중의 하나인 의사결정목의 자료 단편화 및 소집단 자료에 대한 경시성 문제를 보완할 수 있는 방법으로 연관규칙 알고리즘을 활용한 기술적 규칙집합을 찾는 방법을 기술한다. 이를 위해 연관규칙 발견 알고리즘의 원리를 다루고 이를 테이블 형태의 데이터베이스에 효율적으로 적용하는 방법을 기술한다. 아울러 이러한 방법은 원 연관규칙 알고리즘을 이용할 때보다 효율적 작업이 가능함을 실험 데이터에 대한 분석을 통해 살펴보았다.

Abstract

In order to compansate the problem of fragmentating data and disdaining small group of data in decision trees, a descriptive rule set discovery method is suggested. The principle of association rule finding algorithm is presented and a modified association rule finding algorithm for efficiency is applied to target database which has condition and decision attributes to see the effect of modification.

I. 서론

예측(prediction)과 기술(description)은 데이터마이닝의 서로 다른 두 목표이다[1]. 예측은 현재의 데이터에 근거해 알려지지 않은 경우를 미루어 짐작하는 데 목적이 있고, 기술은 데이터의 특징을 사람이 해석하기 쉽도록 규칙 등의 형태로 보여주는 것이다. 예를 들어 신경망에 근거한 주식 가격 예측 시스템은 예측이 주목적이고[2], '인스탄트 커피를 사는 사람의 80%는 커피 프림도 함께 산다.'와 같은 연관 규칙을 발견하려는 시스템은 기술이 주목적이다. 연관 규칙 발견 시스템[3]은 원래 수퍼마켓 등에서 보유하고 있는 트랜잭션 데이터베이스에 대해 어떤 항목들이 주로 함께 팔리나 하는 연관 패턴을 발견하기 위해 개발되었다. 이런 패턴은 트랜잭션 데이터베이스에 내재하고 있는 규칙성에 대한 정보를 제공한다. 예를 들어, 어느 대형 수퍼마켓에서 일정 기간 동안 판매한 기록을 모아 둔 데이터베이스로부터 발견한 위와 같은 규칙은 상품 진열 등, 영업 전략의 수립에 유용할 것이다. 본 논문에서는 대형의 고차원 데이터베이스에 대해 최적의 기술적 규칙을 발견하기 위해 연관규칙 알고리즘을 효율적으로 적용하는 방법에

대해 논한다.

II. 관련 연구 동향

AIS는 첫 연관규칙 발견 알고리듬으로 1993년 발표되었다.[4] 이어 SETM이 트랜잭션 데이터베이스가 관계형일 경우에 대해 개발되었고, 효율성에 중점을 둔 Apriori[3]가 소개되었다. 덧붙여서, 가용한 하드웨어 자원에 따라 주기억 용량이 충분히 크면 AprioriTid 알고리듬 또는 해쉬표에 근거한 연관규칙 발견 알고리듬[5]을 적용할 수 있다. 처리 속도를 더욱 빠르게 하기 위해 병렬처리 버전[6]이라든가 표본 추출에 의한 방법[7], 트리 구조를 이용한 방법[8] 등이 발표되었다. 최근에는 다차원 연관규칙 알고리즘이란 이름으로 일반 데이터베이스에도 적용되고 있다.[9, 10] 그러나 일반 데이터베이스에 적용할 경우 훈련 데이터가 충분히 크지 않으면 과적합 (overfitting) 문제가 발생할 수도 있으므로 주의하여야 한다.

III. 연관 규칙 발견 알고리듬

1. 알고리듬의 원리

$I = \{ i_1, i_2, \ldots, i_m \}$을 수퍼마켓에서 판매하는 항목(item)에 대한 집합이라고 하자. T는 트랜잭션 레코드를 모아 놓은 것으로 각 트랜잭션은 고객이 한번에 구매한 항목집합(itemset) $X \subseteq I$로 구성된다.

연관 규칙이란 $Y \subset I$, $Z \subset I$, $Y \cap Z = \Phi$일 때 $Y \Rightarrow Z$와 같은 규칙을 말하며, $Y \Rightarrow Z$의 신뢰도가 C%란 항목집합 Y를 포함하는 트랜잭션 중 C%는 항목집합 Z도 역시 포함함을 나타낸다. 한 항목집합 $X \subset I$에 대해 X의 지지율(support ratio)이란 전체 트랜잭션 중 X를 포함하는 비율을 말한다. $Y \Rightarrow Z$와 같은 규칙의 신뢰도는 $\frac{(Y \cup Z)\text{의 지지율}}{Y\text{의 지지율}}$로 계산될 수 있다. 지지율 대신 해당 항목집합이 전체 트랜잭션 중에 몇 번 나타났나를 나타내는 지지수(support number)를 사용하기도 한다. 어떤 지지율 이상 자주 나타나는 항목을 빈번 항목집합(frequent itemset)이라 한다.

2. 정의 및 알고리듬

실세계 데이터베이스는 관계형, 계층형, 망형 등 다양한 구조로 구현되어 있다. 이런 데이터베이스에 본 논문에서 기술한 방법을 적용하려면 규칙 발견 대상 자료를 하나의 테이블 형태로 만들어야 한다. 속성(attribute)은 조건(condition part) 및 판정 속성(decision part)으로 나눠지게 되며 이것은 테이블의 각 열의 제목에 해당한다. 여기서 판정 속성은 조건 속성에 의존 관계를 갖고 있어야 하며 조건 속성끼리는 의존 관계를 갖지 않는 것이 바람직하다. 다음은 이러한 표 형식의 데이터베이스로부터 조건 및 판정 값 사이에 있는 연관규칙을 발견하는데 필요한 용어의 정의이다.

- 데이터베이스 : 테이블 형식의 데이터베이스로 조건 및 판정 속성으로 구성되고 각 속성은 유일한 이름을 가짐.
- 항목(item) : 테이블 내 속성-값(attribute-value) 의 쌍. 다시 말해 각 속성-값의 쌍은 하나의 항목을 구성하고 모든 값은 명칭 값임.
- 항목집합(itemset) : 데이터베이스 내의 각 행(레코드)에서 나타날 수 있는 항목의 조합. 데이터베이스의 특성상 한 항목집합 내의 각 속성은 유일해야 함.

이러한 데이터베이스 정의의 가장 큰 효과는 속성을 조건 및 판정 속성으로 나눔으로써 판정 속성은 규칙의 오른 쪽에만 나타날 수 있게 되어 아무런 제한이 없는 원 연관규칙 발견 알고리듬에 비해 생성 가능한 규칙의 수가 상당히 줄어들 수 있게 되는 것이다. 원 알고리듬과 또 다른 차이점은 항목 집합을 만드는 방법이다. 각 항목 집합 내 속성-값 쌍에서 각 속성은 한번 밖에 나타날 수 없다는 것이다. 다음 알고리듬은 연관 규칙 발견 알고리듬의 대표격인 Apriori 알고리듬[3]이다.

```
association_rule_finding_alg
// k: 찾으려는 항목집합의 길이
// D: 데이터베이스
// Fn = { 크기 n의 빈번 항목집합 }
    Ck = generate_candidates ( Fk-1 )
    ∀ record r ∈ D do {
```

```
    C = generate_k_itemsets(r, k)
    ∀ itemset c ∈ C do {
      Count for c' when (c'∈Ck) = (c∈C)
    }}
    Fk = { c' ∈ Ck | count of c'≥최소지지수}
End association_rule_finding_alg
```

generate_k_itemsets(r, k)는 데이터베이스의 한 레코드 r로부터 모든 가능한 길이 k인 항목집합을 생성시킨다. 만일 레코드 길이가 m이라면 mCk 개의 항목집합을 구할 수 있다. *generate_candidates* (F_{k-1}) 부분은 원 알고리즘에 비해 수정이 필요한 부분으로 다음과 같다.

```
generate_candidates ( Fk-1 )
  ∀ itemset ∈Fk-1  do {
    //Iitem(k-1)은 항목집합내 (k-1)번째 항목이다.
    Ck' = {k-itemset |
          Items Iitem1, Iitem2,··, Iitem(k-1), Jitem(k-1)
          are Selected From F I,(k-1), F J,(k-1)
                //F I,(k-1): Fk-1의 I번째 항목집
합
          Where Iitem1=Jitem1,··,Iitem(k-2)=Jitem(k-2),
                Iitem(k-1)<Jitem(k-1), and
                속성(Iitem(k-1))≠속성(Jitem(k-1))
          }
    }
  ∀ itemset c'' ∈ Ck' do {
    //t는 (k-1)-항목집합의 집합
    t = generate_(k-1)_itemsets(c'', k-1)
    If ∀itemsets ∈t Exist in Fk-1 Then
      c'' is a candidate
    }
  Ck = { c'' ∈ Ck' | c''is a candidate}
End generate_candidates
```

위 알고리즘에서는 속성이 다른 항목집합만으로 새로운 항목집합을 만들기 위해 속성을 검사하는 굵은 글씨로 표시한 부분이 원 알고리즘에 비해 추가되었다. 함수 generate_(k-1)_ itemsets(c'', k-1) 은 길이가 k인 항목집합 c''로부터 k-1개 항목을 뽑아 만들 수 있는 모든 항목집합을 생성시켜 준다. 결국, 함수 *generate_candidates*(F_{k-1})는 모든 빈번 (k-1) 항목 집합 F_{k-1}을 입력으로 각 항목이 서로 다른 속성을 갖는 모든 가능한 빈번 k 항목 집합을 반환시켜 준다. F_1은 데이터베이스를 한번 스캔함으로써 구해질 수 있고 이는 초기치로 사용된다.

빈번항목집합을 구할 때 중요한 점은 원 연관규칙 알고리즘처럼 구하는 항목집합의 길이의 제한을 주지 않으면 많은 계산시간이 소요될 수 있다는 점이다. 따라서 데이터의 성질에 따라 구하려는 항목집합의 길이의 한도를 적당히 정해주어야 하는 점이 중요하다. 이것은 또한 과적합 문제를 피할 수 있는 방법이 되기도 하다.

3. 규칙의 신뢰도

찾아낸 항목집합으로 규칙을 만들 수 있으며 일반 연관규칙에서처럼 Y => Z와 같은 규칙의 신뢰도는 $\frac{(Y \cup Z)\text{의 지지율}}{Y\text{의 지지율}}$로 계산될 수 있다. 여기서 Z는 판정 속성만이 될 수 있다. 단, 이와 같은 신뢰도는 지지율 또는 지지수 값이 충분히 클 경우 사용하면 좋다. 지지수가 충분히 크지 않으면 에러 수의 기대치를 구하는 $U_{CF}(E, N)$의 식에 기초를 둔 Quinlan의 Pessimistic error rate[11]를 사용할 수도 있다. 여기서 E는 |Y| - |Y ∪ Z|, N는 |Y|가 되며 CF는 certainty factor의 약자로 디폴트 값으로 25%가 주어진다.

VI. 실 험

본 알고리듬의 타당성을 실세계 자료로 점검해 보고자 캘리포니아 대학교의 자료(UCI machine learning repository)[12]를 이용했다. 비교적 대형이면서 데이터베이스의 구조와 잘 맞는 조건 및 판정 속성으로 되어 있는 Adult라는 데이터베이스를 사용하였는데, 원래 그 자료는 1994년 인구조사통계로부터 발췌된 것으로 총 32,561개의 개체(레코드)로 구성되어 있다. 조건 속성은 6개의 연속치 속성 및 8개의 이산치 속성을 갖고 있다. 판정 속성은 class로 속성 값으로는 년 소득 5만 달러 미만의 class1과 년

소득 5만 달러 이상의 class2가 있다. 최소지지율 20%와 크기가 2인 규칙으로 한정하여 알고리즘을 적용한 결과 다음과 같은 규칙 21개를 발견할 수 있었다. 발견된 규칙에서 주목할 점은 지지수가 모두 수천 단위의 큰수로 규칙의 신뢰도가 확률에 가까운 수라는 것이다.

age=20 => class1 (빈도수=7524, 신뢰도=93.67%)
age=30 => class1 (빈도수=6304, 신뢰도=73.19%)
work_place=private => class1
(빈도수=17733, 신뢰도=78.13%)
. . . . . (생략)

다음은 원 연관규칙발견 알고리듬을 수정없이 테이블 형식의 데이터베이스에 그대로 적용했을 때에 비해 얼마나 절약이 되는지 비교 분석을 하여보자. 총 15개의 속성으로 구성된 Adult 데이터베이스는 각 속성 당 평균 11.86개의 속성값을 갖고 있다. 원 연관규칙발견 알고리듬으로 발견 가능한 규칙의 수를 근사 추정하기 위해 각 속성은 각각 12개씩의 속성값을 가질 수 있는 것으로 하고, 규칙을 구성하는 항목 수가 10 보다 큰 규칙은 길이가 지나치게 길어 흔한 경우가 아니므로 총 항목 수가 10개인 규칙까지 발견한다고 가정한다.

크기 2인 항목집합부터 규칙이 될 수 있으므로 총 15개의 속성 중 2개씩 뽑는 가짓수 즉, ${}_{15}C_2$ 개의 속성 조합이 가능하고, 각 속성이 취할 수 있는 값은 각각 12개이므로 ${}_{15}C_2 \times 12 \times 12$ 개의 후보 항목집합이 가능하다. 후보 항목집합 중 실제로는 많은 수가 데이터베이스에 존재할 것이나 이들 중 극히 보수적으로 1개만 유효하다고 가정한다. 그리고 x, y가 항목이라 할 때 $\frac{xy}{x}$ 즉, x => y의 규칙과 $\frac{xy}{y}$ 즉, y => x의 두 가지 규칙이 생성될 수 있다. 다시 말해 ${}_2C_1$ 개의 규칙이 만들어 질 수 있다. 따라서 크기 2인 항목 집합으로부터 나올 수 있는 규칙수는 극히 보수적 가정에도 불구하고 ${}_{15}C_2 \times 1 \times {}_2C_1 = 210$이다. 이 수는 앞서 실험에서 구한 규칙수의 10배이다. 따라서 원 알고리즘을 수정 없이 바로 대형 데이터베이스에 적용할 경우 규칙을 구하는데 더 많은 시간이 소요될 것을 예상할 수 있다.

V. 결 론

본 논문에서는 연관규칙 발견법을 일반적 다차원 데이터베이스 상에 적용하여 기술규칙을 발견하는 방법을 기술하였다. 이러한 방법은 주어진 최소지지율에 관해 모든 가능성을 고려하여 빈발항목집합을 발견하게되나 생성되는 규칙은 판정속성을 갖는 빈발항목집합이 주 대상이 되므로 일반적 연관규칙 발견 알고리듬을 바로 적용했을 때보다 적은 수의 규칙을 생성하므로 계산 시간이 절약될 수 있다. 다른 장점으로는 대상이 된 데이터베이스 자체가 큼으로써 생성된 기술규칙을 지지하는 개체의 수가 통계적으로 의미 있을 만큼 크게 잡을 수 있으므로 예측의 정확성 또한 높을 가능성이 많다는 점이다.

참 고 문 헌

[1] Payyad, U.M., Piatetsky-Shapiro, G., and Smith, P., "From Data Mining to Knowledge Discovery: An Overview," In *Advances in Knowledge Discovery and Data Mining*, Payyad, U.M., Piatetsky-Shapiro, G., Smith, P., and Uthurusamy, R. ed., AAAI Press/The MIT press, pp.1-34, 1996

[2] Fu, L.M., *Neural Networks in Computer Intelligence*, McGraw Hill, Inc., New York, 1994

[3] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, A.I., "Fast Discovery of Association Rules," In *Advances in Knowledge Discovery and Data Mining*, Fayyad, U.M., Piatetsky-Shapiro, G., Smith, P., and Uthurusamy, R. ed., AAAI Press/The MIT Press, pp.307-328, 1996

[4] Agrawal, R., Imielinski, T., and Swami, A., "Mining Association Rules between Sets of Items in large Databases," In

Proceedings, ACM SIGMOD Conference on Management of Data, Washington, D.C., pp.207-216, 1993

[5] Park, J.S., Chen, M., and Yu, P.S., "Using a Hash-Based Method with Transaction Trimming for Mining Association Rules," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 9, No. 5, pp.813-825, Sept. 1997

[6] Savesere, A., Omiecinski, E., and Navathe, S., *An Efficient Algorithm for Mining Association Rules in Large Databases*, College of Computing, Georgia Institute of Technology, Technical Report No.: GIT-CC-95-04, 1995

[7] Toivonen, H., "Sampling Large Databases for Association Rules," In *Proceedings of 22th International Conference on Very Large Databases(VLDB'96)*, Mumbay, India, Morgan Kaufmann, pp.134-145, Sept. 1996

[8] Han, J., Pei, J., and Yin, Y., "Mining Frequent Patterns without Candidate Generation," In SIGMOD'00, Dallas, TX, May 2000

[9] Wenmin, L., Han, J., and Pei, J., "CMAR: Accurate and Efficient Classification Based on Multiple Class Association Rules," In Proceedings of 2001 International Conference on Data Mining(ICDM'01), San Jose, CA, Nov. 2001

[10] Han, J., and Kamber, M., Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2001

[11] Quinlan, J.R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, Inc., 1993

[12] Murthy, P.M., and Aha, D.W., *UCI Repository of Machine Learning Databases*, http://www.ics.uci.edu/~mlearn/MLRepository.html, Dept. of Information and Computer Science, University of California at Irvine, CA