

함축적 지식 영역에서 부 연관규칙의 발견

박양재*

목 차

- I. 서론
- II. Boolean Analyzer를 이용한 부 연관규칙 알고리즘
- III. 실험 및 결론
 - 참고문헌
 - Abstract

I. 서론

데이터마이닝 기법 중에 많은 연구가 되고 있는 연관규칙[1]은 데이터베이스에서 항목들 간의 상관성을 찾아내는 것을 말하며, 연관규칙 알고리즘을 이용하여 발견한 규칙들은 교차판매, 상품진열, 첨부우편물, 카달로그 디자인 등 많은 방면에 활용되어 사용되고 있다.

하지만 연관규칙보다 부(否) 연관규칙이 새로운 규칙으로 발견되는 경우가 종종 있다. 기존의 연관규칙이 $A \rightarrow B$ 규칙과 같이 A상품을 사는 사람들은 B상품을 사는 규칙이 있는 것을 의미한다면, 부 연관규칙은 $\sim A \rightarrow \sim B$ 규칙과 같이 A상품을 사지 않는 사람들은 B상품을 산다는 규칙이 있는 것으로 어느 한쪽에 not이 들어간 규칙을 의미한다. 이외에도 $A \rightarrow \sim B$, $\sim A \rightarrow B$ 형태의 부 연관 규칙이 있을 수 있다.

부 연관규칙은 연관규칙과 같이 흔하게 나타나는 규칙은 아니지만 support, confidence가 부 연관규칙 쪽에서 훨씬 높게 나타난다면 오히려 부 연관규칙에서 찾아낸 규칙이 훨씬 더 가치 있다고 볼 수 있다. 또한 $\sim A \rightarrow \sim B$ 의 규칙이 있다면 B상품의 판매촉진을 위해 A를 구매하지 않은 사람들을 대상으로 마케팅을 하는 타겟마케팅이 가능할 것이다.

즉, 연관규칙이 구매한 항목에 관심을 가져 구매 항목간의 규칙을 생성하는 것이라면 부 연관규칙은 구매하지 않은 항목에 대해서도 관심을 가짐으로써 구매하지 않은 항목들 또한 규칙에 적용해 기존 전략보다 더 효과적인 규칙을 마케팅 전략에 사용할 수 있다.

1.1. 기존 연구 및 문제점

부 데이터베이스를 Apriori 알고리즘을 이용하여 부 연관규칙을 찾는 방법[2]은 구매하지 않은 데이터를 관찰하기 위해 기존의 데이터를 모두

* 가천의과학대학교 IT학부(유비쿼터스 컴퓨팅전공) 교수

부 데이터베이스로 변환하는 작업이 필요하다. 그러므로 변환된 데이터들이 전체 ITEM의 숫자에 비례하여 증가하게 되는 단점이 있다. 즉, ITEM이 100개인 매장에 4개의 ITEM을 구매한 트랜잭션을 처리하려면 $100 \times 4 = 96$ 개의 데이터를 처리해야 한다. 또한, 처리 결과로 찾게 된 규칙들이 부 데이터베이스에서 찾은 규칙들과 같으므로 $\sim A \rightarrow B$ 와 같은 규칙만을 찾게 되는 단점이 있다.

Taxonomy를 이용한 방법[3]은 Taxonomy 활용으로 정확한 규칙을 찾을 수 있다는 장점이 있다. 그러나 부 연관규칙을 찾으려면 연관규칙으로 찾은 결과를 이용하여야 하고 중복된 규칙을 제거해야 하는 과정 등이 있어 연관규칙만을 찾는 것보다 더 복잡한 알고리즘을 사용하는 단점이 있다.

1.2. 제안하는 알고리즘

Decision Making에서 사용된 Boolean Analyzer [4][5][6]는 보다 간단한 계산과정으로 부 연관규칙을 발견할 수 있다.

$\sim A \rightarrow B$ 의 규칙뿐만 아니라 $\sim A \rightarrow B$, $A \rightarrow \sim B$ 의 규칙까지도 모두 찾아낼 수가 있다. 또한, 연관이 된 정도를 수치적으로 표현이 가능하므로 연관정도의 순위까지도 알 수 있는 장점이 있다. 또한 Taxonomy를 이용한 방법[2]과는 달리 단 한 번의 과정으로 빠르게 원하는 규칙을 찾을 수 있는 장점이 있다.

그러므로 본 논문에서는 Boolean Analyzer를 사용하여 부 연관규칙을 발견하고 발견된 규칙의 연관성으로 순위도를 측정하는 과정을 설명하고자 한다.

II. Boolean Analyzer를 이용한 부 연관규칙 알고리즘

2.1. 알고리즘 과정

Boolean Analyzer는 확률을 이용하여 각 아이템들 사이의 의존성을 계산하여 서로 얼마나 연관이 되어 있는가를 계산하는 방법이다[3][4][5]. Boolean Analyzer 알고리즘은 사건의 확률에 근거해서 의존성(dependency)의 정도를 나타내는 PIM(Probabilistic Interstingness Measure)을 만들어 내고 PIM을 이용하여 dependency rules를 생성한다.

이 알고리즘의 과정을 간단히 요약하면 다음과 같다.

1. Dataset으로부터 State Occurrence Matrix를 만든다.
2. State Occurrence Matrix로부터 State Linkage Matrix를 계산한다.

2.2. State Occurrence Matrix

State Occurrence Matrix는 동일한 구매 패턴의 수를 Matrix로 정리한 것이다. [표 1]에서 A, B, C, D 는 ITEM을, 10, 20, 30, 40은 TID를 나타낸다. "0"은 해당 아이템을 구입하지 않은 것을, "1"은 해당 아이템을 구입했음을 표시한다.

[표 1] DataSet

TID \ Item	A	B	C	D
10	0	1	1	0
20	1	0	0	1
30	0	0	1	1
40	1	1	0	1

ITEM(A, B, C, D)을 두 개의 집합 $X=(A,B)$ 와 $Y=(C,D)$ 로 나누면, row는 집합 X의 모든 가능한 조합이고, column은 집합 Y의 모든 가능한 조합이다. $X \cap Y$ 는 \emptyset 이고 $X \cup Y$ 는 전체 ITEM의 개수이다. [표 1]의 DataSet을 이용해 State Occurrence Matrix를 작성하는데 [표 2]의 각 값은 $X \rightarrow Y$ 의 support이다. 4개 변수의 100개의 DataSet에 대한 예가 [표 2]의 State Occurrence Matrix에 있다.

다음과 같은 결과를 얻을 수 있다.

$$\frac{P(X)P(Y)}{(1-P(X))P(Y)} = \frac{P(X)P(1-P(Y))}{(1-P(X))(1-P(Y))}$$

위의 table의 각 값을 a, b, c, d 라고 가정하면

구분	Y	Y'
X	a	b
X'	c	d

[표 2] State Occurrence Matrix

구분	CD	CD'	C'D	C'D'
AB	10	2	16	9
AB'	2	2	14	6
A'B	2	2	2	4
A'B'	4	4	19	2

2.3. State Linkage Matrix

State Linkage Matrix은 확률을 이용해 ITEM들에 대한 dependency를 계산한 PIM값을 Matrix로 표현한 것이다. X와 Y가 서로 독립이라고 가정하면 X가 일어날 확률은 P(X), X'의 확률(X가 일어나지 않을 확률)은 $1-P(X)$ 이고, Y가 일어날 확률은 P(Y), Y'의 확률(Y가 일어나지 않을 확률)은 $1-P(Y)$ 이다.

a, b, c, d는 서로 독립이므로 $ad - bc = 0$ 이고, 이 식을 이용해서 PIM을 정의할 수 있다. $PIM = ad - bc$ 이고, ITEM X와 Y의 의존의 정도를 나타내는 측정치이다.

음수값은 사건 X와 Y의 inverse dependency를 나타내고, 양수값은 사건 X와 Y의 강한 dependency 관계를 나타낸다. "0"이나 "0"값에 가까우면 두 사건은 서로 독립이므로 아무런 연관 관계가 없음을 나타낸다.

그러면 다음과 같은 식이 얻어진다.

$$P(X \wedge Y) = P(X)P(Y)$$

$$P(X \wedge Y') = P(X)P(1-P(Y))$$

$$P(X' \wedge Y) = (1-P(X))P(Y)$$

$$P(X' \wedge Y') = (1-P(X))(1-P(Y))$$

이것을 다음과 같이 table로 표현할 수 있고,

집합 X와 Y의 원소가 여러 개의 경우에는 PIM을 구하는 일반적인 식이 필요하다.

구분	column j	(column j)'
row i	a_{ij}	$r_i - a_{ij}$
(row i)'	$c_j - a_{ij}$	$N - r_i - c_j + a_{ij}$

State Column Matrix에서

a_{ij} = row i와 column j의 값

N = 전체 DataSet의 크기

r_i = row i에 있는 전체 값들의 합

c_j = column j에 있는 전체 값들의 합

$$PIM = m_{ij} = a_{ij}N - r_i c_j$$

위 식을 이용하여 PIM을 계산해서 State Lin-

구분	Y	Y'
X	$P(X)P(Y)$	$P(X)P(1-P(Y))$
X'	$(1-P(X))P(Y)$	$(1-P(X))(1-P(Y))$

kage Matrix라 불리는 새로운 Matrix [표 3]을 만들 수 있다. State Linkage Matrix에서의 각 값은 row와 column의 dependency 관계를 나타낸다.

[표 3] State Linkage Matrix

구분	CD	CD'	C'D	C'D'
AB	334	-170	-287	123
AB'	-232	-40	176	96
A'B	20	100	-310	190
A'B'	-122	110	421	-409

2.4. 연관규칙 생성

[표 3]의 값 중에 가장 큰 양수값은 421이고 row는 A'B', column은 C'D이다. 이것으로부터 $\sim A \wedge \sim B \rightarrow D$ 라는 부 연관규칙을 얻을 수 있다. 또한, 값이 가장 크므로 모든 경우 중에서 가장 큰 dependency를 가진다.

III. 실험 및 결론

본 논문의 실험은 MS Windows XP, 펜티엄 2 GHz CPU, 1GB RAM의 환경에서 부 데이터베이스를 이용한 Apriori 알고리즘과 Boolean Analyzer를 C++ Programming으로 구현하여 비교하였다.

Synthetic Data를 트랜잭션 수와 ITEM수를 늘려가며 실험하였으며, 각 알고리즘에서 생성된 규칙과 규칙의 종류를 각각 비교하였다.

ITEM의 개수가 10개이고 트랜잭션이 500개인 data에서 생성된 규칙들을 비교해 보면 [표 4]에서 부 데이터베이스를 이용해 Apriori로 생성된 규칙은 $\sim A \rightarrow \sim B$ 형태의 규칙만 찾는 반면, [표 5]

의 Boolean Analyzer로 생성된 규칙은 $\sim A \rightarrow \sim B$ 의 규칙뿐 만 아니라 $A \rightarrow \sim B$, $\sim A \rightarrow B$ 형태의 규칙까지도 찾아낼 수 있으며 dependency의 순위까지도 알 수 있다.

[표 4] Apriori로 생성된 규칙

minimum support	생성된 규칙(support, confidence)
60 %	$\sim 2 \rightarrow \sim 3$ (60.6%, 80.0%)
30 %	$\sim 1 \wedge \sim 2 \rightarrow \sim 3$ (36.4%, 83.3%) $\sim 1 \wedge \sim 7 \rightarrow \sim 2$ (34.8%, 82.6%)

[표 5] Boolean Analyzer 로 생성된 규칙

순위	PIM 값	생성된 규칙
1	282	$1 \rightarrow \sim 5$
2	246	$\sim 2 \rightarrow \sim 3$
3	209	$\sim 4 \rightarrow 5 \wedge 7$
4	135	$1 \wedge 2 \rightarrow \sim 4$
5	114	$1 \wedge 2 \wedge 3 \rightarrow \sim 5$

향후의 연구 방향으로 Boolean Analyzer가 3원 분류 이상에서 정확도가 현저히 떨어지는 단점을 극복할 수 있는 분류기법에 대한 연구와 Boolean Analyzer의 많은 학습 시간의 문제점을 해결한 병렬 시스템에 대하여 연구가 필요하다.

참고문헌

- [1] Altman, E.I., Financial ratio, discriminant analysis and the prediction of corporate bankruptcy, *Journal of Finance*, 23, 1968, pp.589-609.
- [2] Atiya, Amir, F., Bankruptcy prediction for credit risk using neural networks: a survey and new networks, *IEEE transactions on neural networks*, 12(4), 2001,

pp.929-935.

- [3] Joachims, T., Text categorization with support vector machines: learning with many relevant features, *In Proc. of the 10th European Conf. On Machine Learning*, 1998, pp.137-144.
- [4] Thomas G. Dietterich, Machine learning research: Four current direction, *The AI Magazine*, 18(4), 1998, pp.97-136.
- [5] Vapnik, V., *The nature of statistical learning theory*, Springer-Verlag, 1995.
- [6] Han et al. *DMQL: A Data Mining Query Language for Relational Database*. In: *ACM-SIGMOD' 96 Workshop on Data Mining*
- [7] Dunhan M.H., *Data mining: Introductory and advanced topics*. Prentice Hall, 2003.

Finding Negative Association Rules in Implicit Knowledge Domain

Yang-Jae Park*

Abstract

If is interested and create rule between it in item that association rules buys, by negative association rules is interested to item that do not buy, it is attempt to do data Maining more effectively. It is difficult that existent methods to find negative association rules find one part of rule, or negative association rules because use more complicated algorithm than algorithm that find association rules. Therefore, this paper presents method to create negative association rules by simpler process using Boolean Analyzer that use dependency between items. And as Boolean Analyzer through an experiment, show that can find negative association rules and more various rule through comparison with other algorithm.

Key Wrods: Boolean Analyzer

* Professor, Gachon University of Medicine and Science, Dept. of IT(Ubiquitous Computing)