연관성규칙 발견을 위한 효율적인 데이터마이닝 알고리즘 설계

이해각
순천향대학교 공과대학 정보기술공학부

An Efficient Data Mining Algorithm For An Association Rule Discovery

Hae-Kag Lee
Division of Information Technology Engineering, College of Engineering,
Soonchunhyang University

요 약

수많은 데이터로부터 우리가 이용할 수 있는 의미 있는 연관성 규칙을 찾는 것은 대단히 중요하다. 연관성 규칙은 데이터베이스의 각 트랜잭션을 분석하여 이에 대한 각종 측정치를 수집하여 이루어지는데 대단히 많은 시간과 노력을 요한다. 본 논문에서는 통계적 추론을 이용하여 탐색도중 주어진 조건을 만족하는 항목에 대하여 의사결정을 내려 탐색시간은 단축할 수 있는 알고리즘을 제안한다. 또한 추론에 따른 오류발생을 최소화할 수 있는 기법을 제시한다.

1. 서론

지난 수십 년 동안 데이터의 저장과 검색에 관한 여러 가지 기법으로서의 데이터베이스 기술이 연구되고 발전하여 왔다. 금융거래, 병원 데이터, 신용카드 거래 등 대부분의 업무는 대규모의 데이터베이스를 구축하고 관리하며 진행된다. 이러한 데이터베이스의 데이터양은 무제한적으로 증가하고 있는데, 이는 우리가 원하는 정보를 찾아내는 일을 어렵게 만들고 있는 것이 현실이다. 왜냐하면 우리는 대용량의 데이터로부터 의미 있는 지식(knowledge)을 찾는 것이 목적인데 반하여 실제적으로는 데이터만 계속 쌓이는 상황이기 때문이다. 따라서 최근 들어서의 연구는 기존의 데이터베이스 도구로 검색되어지지 않는 숨겨진 정보를 캐내는 기법(knowledge discovery)에 초점이 맞추어져 있다. 데이터 마이닝(Data Mining)은 이러한 기법의 과정 중 대용량의 데이터에 숨겨진 연관성, 패턴, 규칙 등을 찾아내는 일련의 과정을 말한다.

데이터 마이닝 연구의 주요 분야는 연관성규칙 분석(Association Rule Analysis), 연속규칙 분석(Sequence Rule Analysis), 분류규칙분석(Classification Rule Analysis), 군집화분석(Clustering Analysis) 등이 있다. 이중에서 데이터로부터 특정한 연관성을 발견하는 것은 가장 일반적인 작업이라고 할 수 있다.[6]

연관성분석에서 기본적으로 이용되는 기법은 동시발생매트릭스 (Co-occurrence Matrix)이다. 예를 들어 어느 백화점의 거래 데이터가 <표1>과 같다고 하자.

<표 1. 거래데이터의 예>

거래번호	거래품목
T1001	A, B, D
T1002	A, B
T1003	B, C, D
T1004	A, B, C, D
T1005	A, B, E

<표1>의 예제 데이터로부터 품목A를 구입한 고

객은 품목B를 함께 구입하며, 품목B와 C를 구입한 고객은 품목D를 구입한다는 사실을 알 수가 있다. 즉 {A,B}와 {(B,C),D}의 항목은 연관성이 매우 높다는 사실을 찾을 수가 있다.

동시발생매트릭스기법은 이러한 연관규칙을 찾기 위하여 다음과 같은 정의를 한다.

품목 A와 B의 지지도(Support)는

$$P(A \cap B)$$

로 정의한다. 단,

P(X)=(항목집합 X를 포함한 거래수)/(전체거래수)

이며 전체 거래 중의 항목집합 X의 출현 비율을 의미한다. 이 값이 일정한 값(최소지지도)을 초과하면 품목 A와 B의 동시 구매성에 대한 의미 있는 규칙으로 채택하며 이러한 지지도는 두 품목의 동시 구매성에 대한 척도이다.

그러나 보다 관심있는 규칙은 품목A를 구입하면 B도 구입하는지에 관한 규칙이다. 규칙'A⇒B'를 '품목A를 구입하면 B도 구입 한다'로 정의할 때 'A⇒B'의 신뢰확률(Confidence)는

$$P(B \mid A) = P(A \cap B) / P(A)$$

로 정의되며 신뢰확률이 1에 가까울수록 규칙 'A⇒B'를 의미 있는 규칙으로 채택한다. 또한 규칙 'A⇒B'와 규칙 'B⇒A'는 상호 대칭적이지는 않다.

한편 규칙 'A⇒B'의 향상도(Lift 혹은 improvement)는 신뢰도를 독립 가정 하에서의 신뢰도로 나눈 값, 즉,

$$L(A,B) = P(B \mid A) / P(B)$$
$$= P(A \cap B) / \{P(A) \cdot P(B)\}$$

로 정의되며 리프트가 1에 가까우면 두 항목은 서로 독립적인 관계로, 1보다 크면 양의 상관관계로, 1보다 작으면 음의 상관관계로 판단한다.

<표1>의 예제 데이터에 대한 각 측정치의 예는 다음의 <표2>에 나와 있다.

<표2. 예제 데이터에 대한 각 측정치>

항목집합	지지도	신뢰도 ('항목1⇒항목2')	신뢰도 ('항목2⇒항목1')	향상도
A,B	0.8	1.0	0.8	1.0
B,C	0.4	0.5	0.4	0.5
B,D	0.6	0.75	0.6	0.75

의사 결정자는 임계값을 설정하고 탐색을 통하여 얻어진 각 측정치에 대하여 판단을 내린다.

그러나 만약데이터의 양이 대단히 많거나 거래품목이 많다면 이러한 규칙을 발견하는데 대단히 많은 시간을 소비하게 된다. 실제로 품목수가 100가지가 된다면 우리가 체크해야할 규칙의 수는 $2^{100}-1 \approx 10^{30}$가지나 되어 모든 규칙을 체크한다는 것은 불가능하다. 따라서 실제로 의미를 가지는 항목들만의 근거확률을 찾는 노력이 연구되어왔다. 이러한 항목들의 집합을 빈발항목집합(Frequent Item Set)라고하며 최소근거확률을 상회하는 항목의 집합으로 구성된다.

Pasquier[3] 등은 A-Close라고 불리우는 Apriori알고리즘을 제안하였으며 Zaki와 Hsiao[5] 는 CHARM이라는 자료구조적 접근기법을 제안하였다. 또한 Pei[4] 등은 CLOSET라고하는 방법을 제안하였는데 여기에서는 빈발 항목 중 closed set를 정의하고 이러한 집합을 찾는 알고리즘을 제안하였다. Cheung[2] 등은 기존의 탐색된 연관성규칙을 데이터베이스에 새로이 들어오는 트랜잭션에 대하여 유지하는 관리 기법을 제안하였다.

본 논문은 항목집합에 대한 통계적인 연관성 규칙 탐색기법을 제시한다. 다시 말하여 데이터베이스의 모든 트랜잭션을 탐색하여 의미 있는 규칙을 가지는 항목집합을 찾는 것이 아니라 탐색과정 중 통계적 추론을 이용하여 유의한 결론을 얻으면 바로 탐색을 종료하여 규칙탐색시간을 줄이고 오류를 최소화할 수 있는 알고리즘을 제시한다.

2. 본론

2-1. 통계적 배경

어떤 항목집합의 출현확률, 즉, 하나의 트랜잭션에 포함될 확률을 p라고 하자. X를 n개의 트랜잭션 중 그 항목을 포함한 트랜잭션의 개수라고 하고 각 트랜잭션은 확률적으로 독립이라고 가정하면, 탐색 근거확률 X/n 은 n과 p를 모수로 가지는 이항분포를 따른다. 다시말하여,

$$\frac{X}{n} \sim b(n, p)$$

이다. 출현확률을 $\frac{X}{n} = \hat{p}$ 라고 정의할 때 n이 충분히 크다면 중심극한정리(Central Limit Theorem)에 의하여

$$Z = \frac{\hat{p} - p}{\sqrt{\frac{p(1-p)}{n}}} \sim N(0,1)$$

이다. 여기에서 N(0,1)은 표준정규 확률분포를 나타낸다.

p_0를 우리가 정의한 최소 출현확률이라 하고

$$Z_0 = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}}$$

라고 하자.

여기에서 두 개의 통계적 가설(Hypothesis)을 고려하자. 즉,

$H_0 : p = p_0$

$H_1 : p > p_0$

유의 수준을 α라고 할 때 $Z_0 > Z_\alpha$ 이면 H_0를 기각한다.

단, Z_α는 표준정규분포 확률변수 Z에 대하여

$$\alpha = \Pr[Z > Z_\alpha]$$

로 정의된다.

마찬가지로 통계적 가설을

$H_0 : p = p_0$

$H_1 : p < p_0$

와 같이 설정하면 유의 수준 α에 대하여 $Z_0 < -Z_\alpha$ 이면 H_0를 기각한다.

2-2. 측정치에 대한 의사 결정

일반적으로 연관성 규칙 발견을 위한 탐색 과정은 크게 2부분으로 나뉘어 있으며 이는 후보생성 단계와 후보들에 대한 지지도를 계산하는 단계이다.[1][3] 이들은 먼저 단일 항목에 대하여 탐색하고 일정한 지지도를 가지는 즉, 임계값을 초과하는 항목들의 조합 항목에 대한 탐색을 계속한다. 그러나 데이터 수의 무제한적 증가는 탐색 시간을 늘려 우리가 원하는 항목집합에 대한 결론을 내리기까지 많은 시간을 요한다. 본 논문에는 탐색 도중 지지율의 변화 과정을 계속 추적하고 통계적으로 유의한 측정치를 얻으면 바로 탐색을 중지하고 결론을 내린다. 예를 들어 현재까지 탐색된 트랜잭션에 대한 신뢰 확률을 $\hat{p}$, 의사 결정자가 정한 임계값을 p_0라고 했을 때 $Z_0 > Z_\alpha$이면 항목집합에 대한 연관성을 인정하고, $Z_0 < -Z_\alpha$이면 연관성을 기각하고 탐색을 중지한다. 그렇지 않으면 탐색을 진행하여 각 탐색 단계에서 의사결정을 내린다.

2-3. 알고리즘

문제를 단순화하기 위하여 단일 항목들에 대한 지지도의 임계값 p_0 이상을 가지는 항목집합(빈발항목집합; Frequent Set)을 찾는다고 가정하자 (두 개 이상의 항목에 대한 탐색 과정은 동일한 형태로 이루어지며 이에 대한 설명은 생략한다).

앞에서 살펴본 바와 같이 n개의 트랜잭션을 탐색했을 때

$$\hat{p} > p_0 + Z_\alpha\sqrt{\frac{p_0(1-p_0)}{n}}$$

이면 유의수준 α에 대하여 그 항목은 최소지지도를 만족한다고 결론을 내릴 수 있으며 그 항목은 유의 항목으로 결정한다.

또한

$$\hat{p} < p_0 - Z_\alpha\sqrt{\frac{p_0(1-p_0)}{n}}$$

이면 유의수준 α에 대하여 그 항목은 최소지지도를 만족하지 못한다고 결론을 내리고 그 항목은 빈발항목 집합에서 제외 한다

다시 말해서,

$$T = \left[p_0 - Z_\alpha\sqrt{\frac{p_0(1-p_0)}{n}},\ p_0 + Z_\alpha\sqrt{\frac{p_0(1-p_0)}{n}} \right]$$

의 구간은 최소지지율 p_0에 대한 만족 여부의 하한치(Lower Limit)와 상한치(Upper Limit)로 쓰일 수 있으며 오류에 대한 확률은 α 이다. 따라서 구간 T는 탐색 근거확률에 대한 미결정 구간으로 정의하고 이 구간에 포함되는 탐색 근거확률을 가지는 항목은 계속 탐색을 진행하되 탐색 근거확률이 이 구간을 벗어나게 되면 탐색을 중지하고 그 항목에 대한 빈발항목 여부의 결론을 내릴 수 있다.

이러한 결과로 다음과 같은 알고리즘을 제안한다.

Algorithm (Search Algorithm with Stopping-Rule for Minimum-Support)

Step 0 : U={Item1, Item2, ..., Item m} , F = Ø, and N = Ø where U is a Undecided Set, F is a frequent item set, and N is a non-frequent item set.

Step 1 : Read a Transaction Row

Step 2 : Calculate $\hat{p}$ for each item which still remains in U. For the items which are found at the row, make a decision on the items wether $\hat{p}$ values are greater than the upper limit or not under the given p_0 and the significance level α. If $\hat{p}$ is greater than the lower limit then the item is inserted into the frequent item set F.

For the items which are not found at the row, make a decision on the items wether $\hat{p}$ is less than the lower limit If $\hat{p}$ is greater than the lower limit then the item is inserted into the non-frequent item set N.

Step 3 : Check if U is the null set or not. Stop the algorithm if U is the null set. Otherwise, go to step 1.

2-4. 성능평가를 위한 실험결과

위 알고리즘에 대한 성능 평가를 위하여 SAS사의 Enterprise Miner도구에 있는 은행 예제 데이터를 적용하여 시뮬레이션을 수행하였다. 총 트랜잭션의 개수는 7991개이며 대상 항목은 13개이며 실험결과를 요약하면 다음 <표3>과 같다.

<표3. 예제데이터의 시뮬레이션 수행 결과>

p_0	유의 수준	항목	CKING	SVG	ATM	MMDA	TRUST	IRA	CKCRD
		실제빈도	6855	4944	3073	1394	390	866	903
0.05	5%	의사결정	2	2	2	2	1	2	2
		탐색회수	31	31	31	33	575	214	62
	1%	의사결정	2	2	2	2	0	2	2
		탐색회수	31	31	31	33	7991	313	71
0.10	5%	의사결정	2	2	2	2	1	** 1	2
		탐색회수	31	31	31	38	141	120	924
	1%	의사결정	2	2	2	2	1	2	2
		탐색회수	31	31	31	107	218	6733	1031
0.12	5%	의사결정	2	2	2	2	1	1	1
		탐색회수	31	31	31	110	89	120	4794
	1%	의사결정	2	2	2	2	1	1	0
		탐색회수	31	31	31	195	141	120	7991
0.15	5%	의사결정	2	2	2	2	1	1	1
		탐색회수	31	31	31	811	69	120	188
	1%	의사결정	2	2	2	2	1	1	1
		탐색회수	31	31	31	888	89	120	386

<표3> (계속)

p_0	유의 수준	항목	CD	CCRD	HMEQLC	MTG	PLOAN	AUTO	평균
		실제빈도	1960	1237	1316	594	101	742	
0.05	5%	의사결정	2	2	2	2	1	2	-
		탐색회수	32	31	69	128	184	102	117
	1%	의사결정	2	2	2	2	1	2	-
		탐색회수	32	31	70	130	334	107	708
0.10	5%	의사결정	2	2	2	1	1	1	-
		탐색회수	32	355	105	1770	114	4833	656
	1%	의사결정	2	2	2	1	1	0	-
		탐색회수	32	576	111	2186	114	7991	1,476
0.12	5%	의사결정	2	2	2	1	1	1	-
		탐색회수	32	938	111	48	114	364	524
	1%	의사결정	2	2	2	1	1	1	-
		탐색회수	32	1080	355	617	114	509	865
0.15	5%	의사결정	2	1	1	1	1	1	-
		탐색회수	32	403	49	48	114	69	154
	1%	의사결정	2	0	2	1	1	1	-
		탐색회수	32	7991	1903	332	114	294	942

- 실제빈도 : 7,991개의 트랜잭션 중 각 항목의 실제 출현 회수
- 의사결정
  0: 추론에 의한 결정 불가
  1: 비빈발항목으로 결정
  2: 빈발항목으로 결정
- 탐색회수 : 의사결정이 내려질 때까지의 탐색회수

13개 항목에 대하여 4개 수준의 p_0 값, 총 52개 영역에 대하여 실험한 결과 의사 결정의 오류는 1개 영역(IRA항목, p_0=0.1, 유의수준 0.5)에 대해서만 발생하였고(오류확률 1.92%), 두개의 영역에 대하여 의사결정을 내리지 못하였다. 위 테이블에서 보듯이 유의수준 5%의 경우, 전체 트랜잭션 대비 3.6%, 유의 수준 5%의 경우 9.9%의 검색만으로 빈발항목에 대한 추론을 마쳐 탐색 시간을 줄일 수 있었다. 또한 유의수준이 높을수록 오류확률은 줄어들지만 탐색회수가 많아짐을 알 수 있다.

2-5. 최소 탐색횟수의 결정

위에서 제안한 통계적 추론은 탐색 트랜잭션의 개수가 많아질수록 구간의 길이가 짧아져 보다 정확한 결론을 내릴 수 있다. 따라서 최소 탐색회수에 대한 제한을 두어 너무 적은 탐색으로 인한 판단 오류를 방지하기 위한 방법을 제안한다.

위의 알고리즘을 통하여 발생할 수 있는 오류는 두 가지이다. 미결정 집합 U에 남아 있어야할 항목이 빈발항목집합 F 혹은 비빈발항목집합 N에 포함되는 경우와 F 혹은 N에 포함되어야할 항목이 그대로 U에 남아 있는 경우(제2종 오류라고 하자)이다. 전자의 오류를 범할 확률은 유의 수준으로 정해져 있으며 후자의 경우에는 미리 정한 오류의 크기에 따라 달라질 수 있다. 만일 빈발 항목이 아니라고 확신을 가지는 근거확률을 p_1이라 하자. 만일 어느 항목이 가진 근거 확률이 p_1보다 작음에도 불구하고 $\hat{p}$가 하한치보다 크게 나타난다면 이것이 제2종 오류가 되는데 이 값의 최대한계를 β로 정한다. 탐색 트랜잭션의 수를 n이라 할 때, 이를 수식으로 정리하면 다음과 같다.

$$\Pr\left[\hat{p} > p_0 - Z_\alpha\sqrt{\frac{p_0(1-p_0)}{n}}\right]$$

$$= \Pr\left[\frac{\hat{p}-p_1}{\sqrt{\frac{p_1(1-p_1)}{n}}} > \frac{(p_0-p_1)-Z_\alpha\sqrt{\frac{p_0(1-p_0)}{n}}}{\sqrt{\frac{p_1(1-p_1)}{n}}}\right]$$

$$< \beta$$

위 식에서

$\frac{\hat{p}-p_1}{\sqrt{\frac{p_1(1-p_1)}{n}}}$ 는 실제 근거확률이 p_1 이라면 표준정규분포를 따르므로

$\frac{(p_0-p_1)-Z_\alpha\sqrt{\frac{p_0(1-p_0)}{n}}}{\sqrt{\frac{p_1(1-p_1)}{n}}} > Z_\beta$ 이며 이를 다시 정리하면

$$(p_0-p_1) > Z_\alpha\sqrt{\frac{p_0(1-p_0)}{n}}+Z_\beta\sqrt{\frac{p_1(1-p_1)}{n}}$$

이다.

따라서

$$n > \left[\frac{Z_\alpha\sqrt{p_0(1-p_0)}+Z_\beta\sqrt{p_1(1-p_1)}}{p_0-p_1}\right]^2$$

이다. 이것은 최소탐색횟수에 대한 하한값으로 이용할 수 있다.

3. 결론

본 논문에서는 연관성규칙 발견을 위한 효율적인 알고리즘을 제시하였다. 제시된 알고리즘은 검색되는 항목에 대한 최소지지도 만족 여부를 모든 트랜잭션의 검색으로 판단하는 것이 아니라 검색 도중 통계적 추론에 따라 의사 결정을 내림으로써 주어진 유의도 내에서 탐색시간을 단축하였다. 주어진 유의도에 따라 오류의 발생 가능성이 있지만 그 오류는 정해진 한도 내에서 다루어질 수 있다. 다만, 유의도가 작아질수록 검색시간이 길어져 적절한 유의도의 설정이 필요하다. 또한 본 논문에서는 1종과 2종의 오류에 대한 한계를 가지고 최소 탐색 횟수를 결정하는 방법을 제시하였다. 이를 통하여 통계적 추론에 의한 결론의 오류 발생 가능성을 최소화할 수 있다.

참고문헌

[1] R. Agrawal and R. Srikant, Fast Algorithms for Mining Association Rules, Proceedings of the 20th VLDB

[2] D. W. Cheung, J. Han, V.T. Ng, and C.Y. Wong, Maintenance of Discovered Rules in Large Databases: An Incremental Updating Technique. Proc. od 12th Int. Conf. on Data Engineering. 1996.

[3] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed item sets for association rules. In Proc. 7th Int. Conf. Database Theory (ICDC'99), 398-416, 1999

[4] J. Pei, J. Han, and R. Mao. CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets. ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 21-30, 2000

[5] M. J. Zaki and C. Hsiao. CHARM: An Efficient Algorithm for Closed Association Rule Mining. In Technical Report 99-10, Computer Science, 1999

[6] 장남식, 홍성완, 장재호, 데이터마이닝, 대청, 2000.