최대 빈발 패턴을 이용한 온라인 쇼핑객의 구매규칙에 대한 효율적인 마이닝

조재현*, Md. Rezaul Karim, 정병수
경희대학교 컴퓨터공학과
E-mail : {*whwo86, asif_karim, jeong} @khu.ac.kr

Efficient Mining E-Shopper's Purchase Behavior Based on Maximal Frequent Patterns

Jae-Hyun Jo*, Md. Rezaul Karim, Byeong-Soo Jeong
Dept. of Computer Engineering, Kyung-Hee University

요 약

온라인 쇼핑객의 구매 규칙을 예견하기 위해 기업은 데이터 마이닝 기법을 사용하는데, 최대 빈발 패턴은 특정한 고객의 구매 원칙을 드러내기 때문에, 최대 빈발 패턴에 대한 마이닝은 최근 시장 분석에서 핵심적 이슈가 되고 있다. 본 논문에서 우리는 오리지널 데이터세트로부터 널 트랜잭션(Null Transaction)을 제거한 후, 최대 빈발 패턴을 발생시키기 위한 BRE-트리(Bottom-up Row Enumeration Tree)를 적용시켰다. 다음으로 온라인 거래 데이터베이스에서 고객 구매 규칙의 마이닝을 위한 항목들 간의 거리를 계산하기 위해, SCL(Sequence Close Level)의 변형된 버전을 사용하였다. 실험결과는 합리적인 시간 내에 고객의 구매 규칙을 더 정확하게 예견할 수 있음을 보여준다.

1. 서론

온라인 쇼핑은 소비자가 판매자로부터 실시간으로 상품이나 서비스를 직접 구매하는 과정으로서 중개 서비스 없이 인터넷을 통해 이루어지는 전자 상거래의 한 형태이다. 이에 대량의 데이터에서 유용한 정보를 추출하고 확인하는데 데이터 마이닝이 중요하다. 전자상거래 기업들은 장기적으로 모든 구매 데이터베이스를 운용하기 때문에 모든 거래는 순차적으로 저장된다. 거래 데이터베이스 기록은 전형적으로 거래 날짜뿐만 아니라 기존의 거래 과정에서 구매된 제품을 포함한다. 일반적으로 각 기록은 온라인 쇼핑객의 신원(ID)를 포함하는데, 데이터 마이닝을 통해 반복적인 구매를 보이는 데이터베이스상의 온라인 쇼핑객의 구매기록을 쉽게 확인 할 수 있다. 이러한 구매기록은 온라인 쇼핑객의 장기적인 선호도 변화를 드러내기 때문에 유용하게 사용된다.

본 논문이 제안하는 것은 먼저 오리지널 데이터세트에서 널 트랜잭션을 제거한 후, 최대 빈발 패턴의 데이터세트를 생성시키기 위해 BRE-트리를 적용하였다. 다음으로 온라인 거래 데이터베이스에서 고객구매 규칙의 마이닝을 위해, SCL 에 대한 이전 접근법의 한계를 보완하였다.

2. 문제 기술

A. 최대 빈발 패턴의 마이닝

특정한 항목들의 집합를 $I = \{i_1, i_2 \ldots i_n\}$로 할 때, n 은 특정 항목들의 수를 가리킨다. $t = \{i_1, i_2 \ldots i_m\}$에서 t 는 항목들의 주문 목록을 가리키는데, 여기에서 $i1<i2<\ldots<im$ 과 $m \leq n$. 이 성립된다. 거래 데이터베이스 $T = \{t_1, t_2 \ldots t_N\}$에서 N 은 거래집합이고, |N| 는 총거래 건수를 가리킨다. 집합 $X \subseteq I$ 는 패턴으로 불린다. 만약 $X \subseteq t$ 가 성립한다면, X 는 t 에서 일어나거나 t 는 X 를 포함한다. 지지도(X)는 X 를 포함하고 있는 거래의 비율을 가리킨다. 만약 지지도 (X) ≥ min_sup 이라면, X 는 빈발 패턴이 된다. 또한 X 가 빈발 패턴이고 X 의 상위 집합이 빈발패턴이 아니라면, X 는 최대 빈발 패턴이 된다.

<표 1> 거래 데이터베이스

TID	Itemset (Sequence of items)
10	A, B, C, F
20	C, D, E
30	A, C, E, D
40	A
50	D, E, G
60	B, D
70	B
80	A, E, C
90	A, C, D
100	B, E, D

따라서 거래 데이터베이스를 고려할 때, 최대 빈발 패턴 마이닝의 문제는 최대 빈발 패턴의 최종 집합을 발견하는 것이다. 예를들어 <표 4>에서 패턴 "CD", "DE", "CDE"의 발생은 각각 3, 3, 2 이다. 만약 최소 지

지도 한계값(min_sup)이 2 라면, 이 패턴들은 모두 빈발 패턴이 된다. 그러나 “CD”는 최대 빈발 패턴이 아닌데, 왜냐하면 상위 패턴인 “CDE”가 빈발 패턴이기 때문이다. 이 절에서 우리는 문헌들[2,3,5-7]에서 소개된 SCL 을 제시할 것이다.

온라인 고객들에 의해 구입된 항목들은 빈발 패턴을 가리킨다. 따라서 빈발 패턴 마이닝은 온라인 쇼핑객의 행동을 분석하기 위해 널리 사용된다. <표 1> 에 TID 20 에서는 C 와 D 가 서로 근접하지만 TID 30 에서는 떨어져 있다. 시퀀스에서 두 항목들 간의 거리는 그들 관계의 연광성을 결정하는데 도움을 준다. 만약 두 항목이 서로 간 멀리 떨어져 있다면 그들의 관계는 멀어진다. 반대로 두 항목들이 서로 근접하다면 그들의 관계는 연관되어 있다. 따라서 순차 패턴 마이닝은 발생 패턴의 빈도를 계산하여야 할 뿐만 아니라, 시퀀스에서 두 항목들 간의 거리를 계산해야 한다. 항목들 간 거리의 중요성을 설명하기 전에 먼저 거리가 정의 된다. 만약 두 항목이 근접하다면, 그들 간의 거리는 1 로 설정된다. 시퀀스를 위한 거리의 중요성을 계산하기 위해, SCL 이 다음처럼 정의 된다.

$$CL(T^k) = \sum_{i=1}^{k-1}\left(\frac{1}{d_i}\right) \Big/ k-1 \qquad (1)$$

여기서 k 는 최대 빈발 패턴 T^k 의 길이이고, d_i 는두 항목들 간의 거리이다. $d_i = p\ (t_{i+1}) - p\ (t_i)$ 에서 $p\ (t_i)$ 는 거래 t_i 에서 항목의 위치를 가리킨다. $CL\ (T^k)$ 의 값은 0 과 1 사이지만, $CL\ (T^k)$ 이 1 일 때 거래에서 최대 빈발 패턴에 있는 모든 항목들은 서로에게 가깝게 연관되며, 시퀀스는 매우 중요해진다. 이에 반해 $CL\ (T^k)$ 가 0 일 때, 시퀀스는 고객의 구매 규칙이나 구매 역사에 대한 유용한 정보를 포함하지 않게 된다.

사례 1: <표 3>에서 제시 된 걸러진 데이터베이스의 경우 패턴 {A,C,D}는 TID 30 과 TID 90 에서 일어나는 최대 빈발 패턴이며, 두 항목을 포함하기 때문에 3-패턴(T^3)이기도 하다. 따라서 TID 30 과 TID 90 에서 SCL 의 $CL\ (T^k)$는 공식 (1)에 의해 다음처럼 계산될 수 있고, {A, C, D}는 SCL 에서 같은 중요성을 포함하고 있지 않음을 나타낸다.

TID 30 $CL\ (T^2) = (1/1) + (1/2) / (3-1) = 0.75$
TID 90 $CL\ (T^2) = (1/1) + (1/1) / (2-1) = 1$

이러한 방법은 한계를 가지고 있다. 예를들어 <표 3> 에서 패턴 {A, C, E}는 TID 30 과 TID 80 에 최대 빈발 패턴이다. 그렇다면 우리는 TID 80 에서 {A, E, C}의 SCL 은 어떻게 계산하는가. 따라서 이러한 한계가 보완될 필요성이 제기된다. 앞서 우리는 거래에서 항목들 간의 거리 계산을 변형시킬 것(SCL 변형)을 제안하였다. 아래 공식 (2)는 {A, C, E}가 연속한다면 순서가 바뀌어도 항목 사이에 SCL 계산의 차이를 두지 않기 위해 공식 (1)에 d_i 를 보완한 공식이다.

$$d_i = |p\ (t_i) - p\ (t_{i+1})| \qquad (2)$$

사례 2: 패턴 {A, C, E}는 TID 30 과 TID 80 에서 일어나는 최대 빈발 패턴이며, 세 항목들을 포함하고 있기 때문에 3-패턴이다. 따라서 TID 30 과 TID 80 의 SCL $CL\ (T^k)$는 공식 (1)+(2)에 의해 다음처럼 계산될 수 있다 :

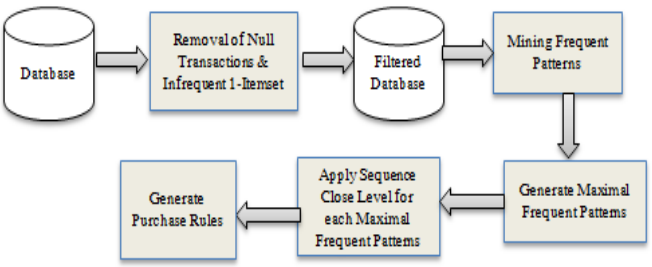
TID 30 $CL\ (T^3) = (1/1) + (1/1) / (3-1) = 1$
TID 80 $CL\ (T^3) = (1/1) + (1/1) / (3-1) = 1$

마찬가지로 최대 빈발 패턴 {C, D, E}의 경우에도 유사한 방식을 사용하여 SCL 을 계산할 수 있다. SCL 에서 볼 때, 최대 빈발 패턴 {A, C, E}와 {A, C, D}는 모든 거래에서 똑 같은 중요성을 포함하고 있지 않은데, 예를 들어 TID 30 에서 패턴 {A, C, E}는 패턴 {A, C, D} 보다 더 중요하다. 따라서 위의 사례에서 볼 때 CL (T^k)값이 커질수록 거래 (T^k)에서 온라인 고객들의 항목들을 구입할 가능성은 증가한다.

3. 제안하는 접근

우리는 빈발 1-항목집합과 2-항목집합를 삽입하기 위해 BRE-트리(Bottom-up Row Enumeration Tree)를 사용하였으며, 다음으로 BRE-트리로부터 최대 빈발 패턴의 최종 세트에 대한 마이닝을 실행 하였다.

오리지널 데이터베이스에 대한 반복적인 접근을 피하기 위해, 우리의 접근은 다음처럼 이루어졌다.
(i) 널 트랜잭션은 연관성 규칙이나 SCL에 기여하지 않기 때문에, 첫 데이터베이스 스캔에서 널 트랜잭션이 제거 되었으며, 수직적 포맷으로 데이터베이스를 나타냈다.
(ii) 걸러진 데이터베이스에 대한 두 번째 스캔에서 우리는 빈발 1-항목집합을 BRE-트리에 삽입시켰다.
(iv) 빈발 패턴에 대한 마이닝이 실행되었고, 그런 다음 최대 빈발 패턴이 트리로부터 생성되었다. 불필요한 패턴들을 줄이기 위해, 트리의 모든 레벨에서 가지치기(Pruning)가 적용되었다.



(그림 1) 제안하는 구매 규칙 마이닝의 작업 흐름도

<표 2> 수직적 포맷 데이터베이스

Items	TID Sets
A	10, 30, 80, 90
B	60, 100
C	20, 30, 80, 90
D	20, 30, 50, 60, 90, 100
E	20, 30, 50, 80, 100
F	10
G	50

입력: i) 거래 데이터 베이스, ii) 지지도 임계값 'min_sup', iii) SCL 임계값 'min_seq_close_lev'

출력: i) 지지도 임계값 'min_sup' 기준에 만족하는 최대 빈발 패턴의 최종세트, ii) SCL 임계값 'min_seq_close_lev'을 만족하는 구매규칙의 최종세트

1단계: 데이터를 스캔하여 수직적 배치 포맷을 사용하여 나타낸 후, 모든 '널 트랜잭션'을 제거한다.

2단계:
널 트랜잭션의 제거 후 걸러진 데이터베이스를 스캔하며, BRE-트리를 구성하고, 트리로부터 빈발 패턴의 최종세트를 발견한다.
[BRE-트리에서 P가 후보 항목세트라고 가정하면]
1. If Sup(P) < min_sup 이면, 트리에서 P를 제거한다. // Level1 가지치기
2. 그 외의 P는 빈발 패턴이다.

3단계: 최대 기준을 점검하여 최대 빈발 패턴의 최종세트를 발견한다.

4단계: 두 번째로 거래 데이터베이스를 스캔한 후, 3단계에서 발견된 최대 빈발 패턴 TID를 사용하여 SCL을 계산한다.
[X가 최대 빈발 k-패턴이라고 가정하면]
1. If ($CL(T^k x)$ > min_seq_close_lev) 이면, X는 중요한 구매규칙이다.
2. 그 외의 X는 중요한 구매 규칙이 아니다. // Level2 가지치기

(그림 2) 온라인 쇼핑객의 구매규칙 마이닝 알고리즘

4. 예시적 사례

온라인 고객들의 구매 규칙은 최대 빈발 패턴에 마이닝에 의해 발견될 수 있는데, 최대 빈발 패턴은 온라인 고객의 구매 규칙을 드러내기 때문이다[5,7]. 이제 <표 1>에서 주어진 데이터베이스에 대해 <표 2>의 수직적 배치 포맷을 적용해보자. <표 1>에서 TID 10과 TID 50은 널 트랜잭션이 될 것으로 예상된다. TID 40과 TID 70 또한 널 트랜잭션이며, 마이닝을 위해 이미 고려하지 않고 있다.

널 트랜잭션은 단지 1-항목집합에서만 포함하고 있기 때문에 중요하지 않을 뿐만 아니라 연관성 패턴과 규칙 마이닝에 기여하지 않기 때문에 마이닝에서 제거되었다. 걸러진 거래 데이터베이스는 <표 3>에서 제시하고 있다. <표 3>으로부터 우리는 빈발 1-항목집합 A, B, C, D, E의 지지도가 각 3, 2, 4, 5, 4 라는 사실을 알 수 있다. 다음으로 우리는 BRE-트리를 구성하였다. (그림 3)은 지지도를 가지고 최종적으로 가지치기가 된 트리와 TID집합를 보여 주고있다.

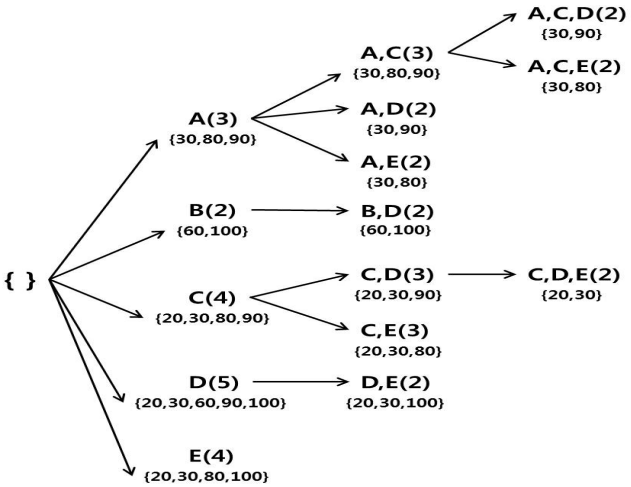
<표 3> 걸러진 데이터베이스

TID	Itemset
20	C, D, E
30	A, C, E, D
60	B, D
80	A, E, C
90	A, C, D
100	B, E, D

<표 4> min_sup에의해 추출된 빈발 패턴

Frequent Pattern	Support	Frequent Pattern	Support
AC	3	CE	3
AD	2	DE	3
AE	2	ACD	2
BD	2	ACE	2
CD	3	CDE	2

(그림 3)에서 제시된 트리를 사용하여 <표 5>와 같이 생성된 네 가지 최대 빈발 패턴들은 BD와 ACD, ACE, CDE이며 각각의 지지도는 2이다. BRE-트리로부터 우리는 최대 빈발 패턴들이 발생하는 TID집합을 발견할 수 있었다. 예를들어 패턴 BD는 TID 60과 TID 100에서 발생했고 패턴 ACE는 TID 30과 TID 80에서 발생했으며, 패턴 ACD는 TID 30과 TID 90에서 발생했고 패턴 CDE는 TID 20과 TID 30에서 발생했다. 따라서 거리 기록에 대한 최대 빈발 패턴의 중요성을 이해하기 위해, 발견된 최대 빈발 패턴을 위한 CL (T^k) 계산이 이루어졌다. SCL의 min_sup을 0.60이라고 가정하자. CL (T^k) 의 결과는, <표 5>에서 제시된 최대 빈발 패턴에서 CL (T^k) 의 값이 클수록 시퀀스에 있는 모든 항목들이 더 근접하게 되고, 온라인 고객이 k-트리 빈발 패턴에 따라 항목들을 구매할 가능성이 더 높아진다는 사실을 보여준다[7]. 예를 들어 TID 30에서 {A, C, E}와 {A, C, D} 같은 구매 규칙이나 구매 시퀀스가 매우 중요한 의미를 가진다. 이에 반해 TID 90에서는 {A, C, D}가, TID 80에서는 {A, C, E}가, TID 30에서는 {C, D, E}가 각각 중요하지만 다른 것들과 비교하여 덜 중요한 규칙들이다



(그림 3) 가지치기 된 BRE-트리

MFP	CL(T^k)	TID	Items
{B, D}	1	60	B, D
	0.5	100	B, E, D
{A, C, D}	1	30	A, C, E, D
	0.75	90	A, C, D
{A, C, E}	1	30	A, C, E, D
	0.75	80	A, E, C
{C, D, E}	1	20	C, D, E
	0.75	30	A, C, E, D

<표 5> 최대 빈발 패턴으로부터 SCL CL (T^k) 계산

5. 실험결과

A. 프로그래밍 환경

실험은 Intel Core2 Duo 2.4GHz CPU, 4GB 메모리이며, 하드웨어는 500GB이다 운영체제는 윈도우 XP에서 수행하였다. 본 연구의 알고리즘은 Mircosoft Visual C++ 6.0으로 구현되었다.

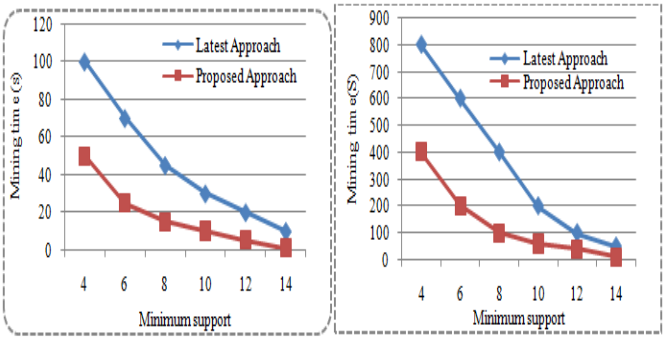
B. 데이터세트의 설명

우리는 '머쉬룸(mushroom)'과 '커넥트-4(connect-4)' 데이터세트를 사용하여 연산을 수행하였다. 이 데이터세트는 웹사이트 http://archive.ics.uci.edu/ml/에서 다운로드 되었다. 머쉬룸 데이터세트는 8124회의 거래 건수를 포함하고 있으며 평균 길이는 23이다. 이에 반해 커넥트-4 데이터세트는 135,115회의 거래 건수를 포함하고 있으며 평균 길이는 8~35이다. 머쉬룸과 커넥트-4 데이터세트의 크기는 각각 5MB와 20MB이다.

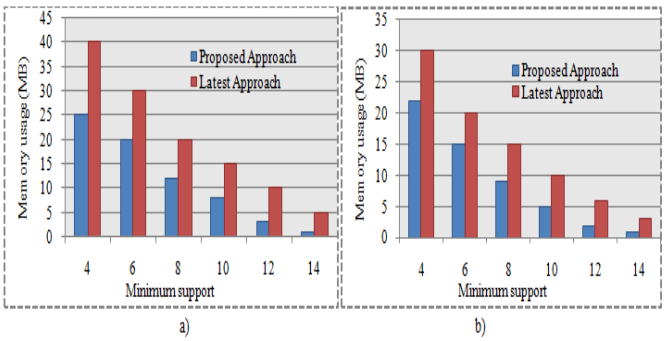
C 성과분석

우리는 0부터 1의 범위 값에 SCL min_sup을 0.65로 설정했으며, 이전의 알고리즘[7]과 우리의 결과를 비교하였다. 이전의 알고리즘은 'Latest Approach'이며, 우리의 제안은 'Proposed Approach'로 명명했다.

첫 번째 실험에서 우리는 'Proposed Approach'와 'Latest Approach'의 실행 시간을 비교하였다. (그림 4)에 a)는 머쉬룸, b)는 커넥트-4 데이터세트의 실행 시간을 보여준다. 커넥트-4 데이터세트는 더 많은 거래들이 있을 뿐만 아니라 그 거래들의 길이가 가변적이기 때문에 시간이 a)에 비해 더 오래 소요된다. 두 번째 실험에서는 메모리 사용을 비교하여, (그림 5)와 같이 우리가 제안한 접근이 이전 것에 비해 적은 메모리를 소비한다는 사실을 발견하였다.



(그림 4) min_sup를 이용한 마이닝 실행시간 비교
a) Mushroom b) Connect-t dataset



(그림 5) min_sup를 이용한 마이닝 메모리사용 비교
a) Mushroom b) Connect-4 dataset

6. 결론

본 논문에서 우리는 최대 빈발 패턴과 SCL 을 사용하여 온라인 쇼핑객의 구매 규칙을 예견할 수 있는 개선된 솔루션을 제안하였다. 실험 결과는 온라인 쇼핑객의 구매 규칙을 발견하기 위해 더 정확한 척도를 제공할 뿐만 아니라 온라인 마케팅 결정에 유용한 척도를 제공한다는 사실을 보여 주었다. 지금까지 사업은 건실한 비즈니스 모델이 없거나 이러한 모델을 이해하지 않은 채 자주 온라인 쇼핑 기법을 채택하였으며, 고객의 기대를 충족시키지 못한 채 조직 문화와 브랜드 명을 지원하는 웹 스토어를 만들었다. 따라서 장래에 우리는 전자상거래 모델과 관련된 다양한 변수들과 요인들을 고려하여 이 작업을 확대시킬 계획이다.

참고문헌

[1] Pang-Ning TAN, M. Steinbach: "Introduction to Data Mining", Addison Wesley, 2007.
[2] S. Liao, Y. Chen: "Mining customer knowledge for electronic catalog marketing", Expert Systems with publications, no.27, pp.521–532, 2004.
[3] P. Giudici, and G. Passerone: "Data mining of association structures to model consumer behavior", Computational Statistics and Data Analysis, no.38, pp.533–541, 2002.
[4] F. Masseglia, P. Poncelet, and M. Teisseire: "Incremental mining of frequent patterns in large databases", Data & Knowledge Engineering, vol.46, no.1, pp.97–121, 2003.
[5] L. Jian, and W. Chong: "Prediction of E-shopper's Behavior Changes Based on Purchase Sequences" International Conf. on Artificial Intelligence and Computational Intelligence (AICI), 2010.
[6] A. Meenakshi, and Dr. K. Alagarsamy: "Efficient Storage Reduction of Frequency of Items in Vertical Data Layout" International Journal on Computer Science and Engineering, Vol. 3 No. 2 Feb 2011.
[7] C.Wang, J. Liu, and Y. Wang: "Mining Online customers Purchase Rules Based on K-trees Frequent Pattern" 7th International Conf. on Fuzzy Systems and Knowledge Discovery (FSKD), 2010.
[8] H. Cheng, X. Yan, J. Ha: "incremental mining of frequent patterns in large database", In Proceedings of International Conference on Knowledge Discovery and Data Mining, KDD, 2004.
[9] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto: "Mining frequent patterns by pattern-growth: The PrefixSpan approach", IEEE Transactions on Knowledge and Data Engineering, vol.16, 2004.
[10] H. Song, and J. Kim: "Mining the change of customer behavior in an Internet shopping mall", Expert System with Applications, Vol.21, no. 3, pp. 157–168, 2001.
[11] R. Agrawal, and R. Srikant: "Fast algorithms for mining association rules", In Proceedings of the International Conference on Very Large Data Bases, pp, 1995.
[12] J. Han, and M. Kamber, Data Mining: Concepts and Techniques, 2nd Edition: Morgan Kaufmann, 2006.