Journal of the Korean Institute of Industrial Engineers
Vol. 32, No. 3, pp. 242-252, September 2006.

트랜잭션 연결 구조를 이용한 빈발 Closed 항목집합 마이닝 알고리즘

한경록† · 김재련

한양대학교 산업공학과

An Efficient Algorithm for Mining Frequent Closed Itemsets Using Transaction Link Structure

Kyong Rok Han · Jae Yearn Kim

Department of Industrial Engineering, Hanyang University

Data mining is the exploration and analysis of huge amounts of data to discover meaningful patterns. One of the most important data mining problems is association rule mining. Recent studies of mining association rules have proposed a closure mechanism. It is no longer necessary to mine the set of all of the frequent itemsets and their association rules. Rather, it is sufficient to mine the frequent closed itemsets and their corresponding rules. In the past, a number of algorithms for mining frequent closed itemsets have been based on items. In this paper, we use the transaction itself for mining frequent closed itemsets. An efficient algorithm is proposed that is based on a link structure between transactions. Our experimental results show that our algorithm is faster than previously proposed methods. Furthermore, our approach is significantly more efficient for dense databases.

***Keywords*:** Association Rule, Frequent Closed Itemset, Transaction Link Structure

1. 서 론

대용량의 데이터베이스에서 사건들이 함께 발생하거나, 또는 하나의 사건이 다른 사건을 암시하는 것과 같은 사건간의 상호관계를 나타내는 연관규칙(association rule)을 발견하는 문제는 가장 중요한 데이터 마이닝 문제들 중의 하나이다(Chen *et al*., 1996; Michael and Gordon, 1997; Pieter and Dolf, 1996). 하나의 트랜잭션을 여러 개의 항목들(items)의 집합으로 보고 이러한 트랜잭션들이 하나의 집합으로 주어지면, 연관규칙은 "$X \Rightarrow Y$"라는 형태로 표현되며 여기서 X와 Y는 항목들의 집합이다(Klemettinen *et al*., 1994). 그러한 연관규칙의 의미는 X를 포함하는 트랜잭션들이 Y 또한 포함하는 경향이 있다는 뜻이다. "모든 트랜잭션들 중에서 60%는 기저귀와 맥주를 함께 포함하고 있다."와 같은 정보에서 60%를 지지도(support)라 하고, "기저귀를 구입하는 고객 중에서 90%의 고객이 맥주를 같이 구입한다."와 같은 정보에서 90%를 신뢰도(confidence)라고 부른다.

연관규칙을 발견하는 문제는 다음의 두 가지 하위 문제로 나누어진다(Agrawal and Srikant, 1994; Agrawal *et al*., 1993).

1) 사용자가 지정한 최소지지도(minimum support) 이상의 지지도를 갖는 항목집합들(itemsets)을 찾는 단계이다. 항목집합에 대한 지지도란 전체 트랜잭션 중에서 그 항목집합을 포함하는 트랜잭션들의 수를 의미한다. 여기서 최소지지도를 만족하는 항목집합을 빈발(large or frequent) 항목집합이라 부르며, 그 이외의 항목집합은 비빈발(small) 항목집합이라고 한다.
2) 빈발 항목집합들을 이용하여 규칙을 생성하는 단계이다. 예를 들어, 항목집합 ABCD와 AB가 빈발 항목집합이면, '신뢰도=지지도(ABCD)/지지도(AB)'의 비율을 계산함으로써 "AB ⇒ CD"와 같은 연관규칙을 생성시킬 수 있다. 만약 '신

† 연락저자 : 한경록 박사과정, 133-791 서울시 성동구 행당동 17번지 한양대학교 산업공학과, Fax : +82-2-2296-0471,
E-mail : krhan516@hanyang.ac.kr
2006년 1월 접수; 2006년 5월 수정본 접수; 2006년 6월 게재 확정.

뢰도 ≥ 최소신뢰도(minimum confidence)'이면 그 규칙은 강한 연관규칙이라고 부르며 사용자에게는 잠재적 사용가치가 큰 정보로 인식된다. 이 규칙은 ABCD가 빈발이기 때문에 최소지지도를 만족할 것이다.

최근 들어서는 모든 빈발 항목집합과 연관규칙을 발견하는 대신에, closure 개념을 사용하여 빈발 closed 항목집합만을 발견하는 연구가 진행되고 있다. 모든 빈발 항목집합을 발견하는 기존의 알고리즘과 비교해 볼 때, 빈발 closed 항목집합만을 발견하는 알고리즘은 같은 결과를 얻어내면서도 실질적으로 발견되는 빈발 항목집합과 연관규칙의 수를 감소시켜준다 (Pasquier *et al*., 1999b).

본 논문에서는 주어진 데이터베이스를 한 번 스캔하고 주어진 데이터베이스보다 크기가 작아진 데이터베이스를 추가적으로 두 번 스캔하면서 효율적으로 빈발 closed 항목집합을 발견하는 알고리즘을 제시한다. 제안하는 알고리즘은 두 단계로 나누어지며 첫 번째 단계에서는 데이터베이스를 스캔하여 1-항목집합의 지지도를 계산하고, 최소지지도를 고려하여 각 트랜잭션에서 구매한 항목들의 수를 의미하는 구매항목수를 기준으로 오름차순 정렬한다. 두 번째 단계에서는 정렬된 데이터베이스를 읽으면서 트랜잭션들을 서로 교차하여 포함관계를 고려하고, 트랜잭션 연결 구조를 완성하여 빈발 closed 항목집합을 발견한다.

본 논문의 구성은 다음과 같다. 2장에서는 연관규칙, 빈발 항목집합, 빈발 closed 항목집합의 정의에 대한 설명을 하고 기존 연구를 고찰해 본다. 3장에서는 트랜잭션 연결 구조를 정의하고 제안하는 알고리즘을 단계별로 설명하며, 수치 예제와 함께 논의한다. 4장에서는 기존의 알고리즘들과 비교한 실험결과를 보여준다. 5장은 본 연구의 결론을 기술한다.

2. 연관규칙

2.1 빈발 항목집합

$I=\{i_1, i_2, \cdots\cdots, i_m\}$를 항목이라 불리는 문자들의 집합이라고 하자. D는 트랜잭션들의 집합이고 각 트랜잭션 T는 $T \subseteq I$인 항목들의 집합이라고 하자. 한 트랜잭션에서 구입하는 항목들의 수량은 고려하지 않기로 가정한다. I의 원소인 항목들의 집합을 X라 할 때 $X \subseteq T$이면 "트랜잭션 T가 X를 포함한다."라고 말한다. 연관규칙은 "$X \Rightarrow Y$"의 형태로 표시되고, 여기서 $X \subset I$, $Y \subset I$ 및 $X \cap Y = \varnothing$이다. 전체 트랜잭션들 중에서 s%가 $X \cup Y$를 포함하면 연관규칙 "$X \Rightarrow Y$"는 지지도 s%를 가지고 있음을 의미한다. 만약 X를 포함하는 트랜잭션들의 c%가 Y 또한 포함하고 있으면 연관규칙 "$X \Rightarrow Y$"는 신뢰도 c%를 가지고 있다는 뜻이다. 최소지지도 이상을 갖는 항목집합을 빈발 항목집합이라 한다. k개의 항목들로 이루어진 빈발 항목집합을 빈발 k-항목집합이라 한다. 연관규칙 문제는 사용자가 지정한 최소지지도와 최소신뢰도 이상의 지지도와 신뢰도를 갖는 모든 빈발 항목집합들과 연관규칙을 발견하는 문제이다(Han and Fu, 1995; Han and Kamber, 2001; Srikant and Agrawal, 1995).

<Table 1>에 데이터베이스가 주어져 있고 최소지지도를 40%라고 가정한다. 즉, 5개의 트랜잭션들 중에서 2번 이상 발생하면 빈발이다. TID는 Transaction IDentifier이다. 이 데이터베이스를 대상으로 모든 빈발 항목집합을 찾는 대표적인 알고리즘인 Apriori 알고리즘(Agrawal and Srikant, 1994)을 적용하면 <Table 2>와 같이 빈발 1-항목집합부터 빈발 4-항목집합까지 모두 20개의 빈발 항목집합을 찾을 수 있다. 편의상, 항목집합 {A, B, C}를 ABC로 표시한다.

Table 1. Database-1

TID	Items
1	A E F
2	B C E
3	A B C D E
4	A B C D
5	B C E

Table 2. Frequent itemsets

Support	Frequent itemsets
80%	B, C, E, BC
60%	A, BE, CE, BCE
40%	D, AB, AC, AD, AE, BD, CD, ABC, ABD, ACD, BCD, ABCD

2.2 빈발 closed 항목집합

연관규칙을 발견하는 데는 두 가지 문제점이 있다. 많은 빈발 항목집합들이 발견되고, 그에 따른 연관규칙 또한 많다는 것이다. 이 문제에 대한 대안으로 빈발 closed 항목집합을 발견하는 알고리즘이 있다. 예를 들어, 데이터베이스에 $\{(i_1, i_2, ..., i_{10}), (i_1, i_2, ..., i_5)\}$이라는 두 개의 트랜잭션이 있다고 하자. 최소지지도는 50%이고, 최소신뢰도도 50%라고 하자. 기존의 알고리즘으로 구하면 (i_1), ..., (i_{10}), (i_1, i_2), ..., (i_9, i_{10}), ..., $(i_1, i_2, ..., i_{10})$과 같이 2^{10}-1(약 10^3)개의 빈발 항목집합이 발견된다. 그러나 closure 개념을 사용하면 $\{(i_1, i_2, ..., i_5), (i_1, i_2, ..., i_{10})\}$과 같이 두 개의 빈발 closed 항목집합만을 발견하고, 연관규칙도 "$(i_1, i_2, ..., i_5) \Rightarrow (i_6, i_7, ..., i_{10})$"과 같이 하나만 만들어진다. 모든 빈발 항목집합과 모든 연관규칙은 빈발 closed 항목집합과 그에 따른 연관규칙으로부터 만들 수 있다.

<Table 2>에서 지지도가 60%인 빈발 항목집합들을 살펴보면 빈발 2-항목집합인 BE, CE가 빈발 3-항목집합인 BCE와 같이 속해 있다. BE와 CE는 BCE의 부분집합이면서 BE, CE, BCE가 같은 지지도를 가지고 있기 때문에 모두 찾는 것은 불필요하다. BCE의 지지도를 이용하여 부분집합인 BE와 CE의

지지도는 자연스럽게 유도할 수 있기 때문이다. 어떤 항목집합이 빈발이고 그 항목집합과 지지도가 같으면서 자신을 포함하는 다른 항목집합이 없다면 그 항목집합을 빈발closed 항목집합이라고 한다. 즉, 어떤 빈발 항목집합에 대해 그것의 부분집합들과 지지도를 비교하여 지지도가 다른 부분집합들만 closure로 채택한다. 따라서 빈발 closed 항목집합은 ABCD, BCE, AE, BC, A, E로서 6개만 발견된다.

<Table 1>에서 항목을 기준으로 보면A, B, C, D의 4개의 항목을 모두 구입한 고객은 3번과 4번이고, 반대로 트랜잭션 기준으로 3번과 4번 고객은 공통으로 구입한 항목이 A, B, C, D 이외에는 없다. 이처럼 어떤 항목집합에 대해 그 항목집합을 구입한 고객을 모두 찾아낸 후에 다시 그 고객들이 그 항목집합 이외에는 공통으로 구입한 항목이 없으면 그 항목집합을closed 항목집합이라고 한다. <Table 1>을 보면 B와 E를 같이 구입한 고객이 2번, 3번, 5번이지만 세 고객은 B와 E 이외에도 C 항목이 공통이어서 BE를 closed 항목집합이라고 할 수 없는 것이다. 즉, BE의 closure는 BCE가 된다. 또 6개의 빈발 closed 항목집합의 지지도를 이용하면 나머지14개의 다른 빈발 항목집합들의 지지도는 포함관계를 고려하여 쉽게 유추할 수 있어서 정보의 손실 또한 없다.

2.3 기존 연구

연관규칙 문제의 대표적인 알고리즘이 Apriori 알고리즘이다(Agrawal and Srikant, 1994; Jiuyong *et al*., 2002). 이 알고리즘에서 사용하는 개념을 바탕으로 항목간의 연관관계를 찾는 연구가 진행되었고, 항목의 수량을 고려하거나(Srikant and Agrawal, 1996) 항목에 제약을 주는(Jiuyong *et al*., 2002; Srikant *et al*., 1997) 알고리즘들이 개발되었다. 대부분의 알고리즘들이 이진 데이터로 이루어진 트랜잭션들 사이의 관계성을 파악하지만, 현실 세계에서는 수량적 속성의 정보를 포함한 트랜잭션들을 다루어야 할 필요가 있다. 따라서 항목의 수량 정보까지 포함하여 분석하려는 수량적 연관규칙에 대한 연구가 진행되었다(Hong *et al*., 1999; Rajeev and Shim, 2001; Srikant and Agrawal, 1996; Takeshi *et al*., 1999).

또한 Galois connection에 기초한 closure 개념을 이용하여 빈발 closed 항목집합을 발견함으로써 빈발 항목집합과 연관규칙의 수를 줄이는 알고리즘이 연구되었다(Pasquier *et al*., 1999a; Pasquier *et al*., 1999c). Apriori 알고리즘을 기반으로 하는 Close 알고리즘과 A-Close 알고리즘이 제안되었는데, A-Close 알고리즘(Pasquier *et al*., 1999b)은 join 과정을 반복하는 Apriori 알고리즘의 단계를 그대로 따르면서 generator를 생성하기 때문에 낮은 최소지지도가 주어지거나 긴 패턴을 찾는 것에 대해서는 비효율적이다. 또한 빈발 closed 항목집합을 사용하여 중복되지 않는 연관규칙을 발견하는 연구도 진행되고 있다(Pasquier *et al*., 2000).

Charm 알고리즘(Zaki and Hsiao, 1999)은 항목집합과 트랜잭션집합(tidset)을 동시에 고려하여 빈발 closed 항목집합을 발견하는데, 항목집합들 사이의 많은 교차(intersection)가 발생하고 각 항목집합마다 TID를 같이 기억하고 있어야 하는 문제가 있다. 예를 들어, 100,000개의 트랜잭션에 최소지지도가 50%라면 빈발 항목집합은 50,000개 이상의 TID를 기억하고 있어야 한다. 또한 각 항목집합의 지지도를 구하기 위해서는 나열된 TID들을 다시 세어야만 한다.

dCharm 알고리즘(Mohammed and Karam, 2003)은 diffset이라는 수직적인 데이터 표현 형식을 사용하여 빈발closed 항목집합을 발견하고 수평적인 데이터 표현 형식과 비교했는데, 수직적 데이터베이스를 기초로 하여 큰 문제를 분할해서 빈발 항목집합을 탐색하기 위해 동치류(equivalence class)라는 개념을 사용했다. 동치류 접근법의 장점은 큰 탐색범위를 작은 탐색범위로 분할하여 각 서브탐색트리를 독립적으로 탐색할 수 있다는 것이다.

CLOSET 알고리즘은 FP-tree를 사용하여 후보 항목집합을 생성하지 않고 빈발 closed 항목집합을 발견한다(Han *et al*., 2004; Pei *et al*., 2000). 그러나 CLOSET은 조건부 데이터베이스(conditional database)를 계속 만들어 나가야 하고, 하나의 트랜잭션이 여러 조건부 데이터베이스에 중복 저장되는 문제가 있다. 또한 기존의 알고리즘들이 제안한 효과적인 전략들을 통합시킨 CLOSET+ 알고리즘이 개발되어 수행 시간뿐만 아니라 메모리 사용에 대해서도 비교 실험을 했다(Wang *et al*., 2003). 그리고 closure 개념을 그래프 구조에 적용하여 빈발 closed 그래프 패턴을 찾는 CloseGraph 알고리즘이 제안되었다(Yan and Han, 2003).

최근에는 두 개 이상의 데이터 마이닝 기법들의 통합에 대한 알고리즘들이 제안되고 있다. Bing *et al*.(1998)은 연관규칙과 분류를 통합하여 분류 연관규칙이라는 개념을 제시했고, Tsay and Chien(2004)은 군집화 기법을 응용하여 클러스타분해 연관규칙을 제안하였다. Hsu *et al*.(2003)은 분류 트리를 생성하기 위해 연관규칙과 유전 알고리즘의 접목을 시도하였다. 또한 데이터 마이닝 기법을 웹에 적용하여 웹에서의 개인화 서비스 향상을 위해 웹 마이닝에 대한 연구가 진행되고 있다(Lee *et al*., 2001).

3. 제안하는 알고리즘

3.1 트랜잭션 연결 구조

3.1.1 기호 및 용어 설명

TDB(Transaction DataBase) : 트랜잭션 데이터베이스

ODB-1(Object DataBase-1) : TDB를 스캔하여 구매항목수가 2 이상인 트랜잭션을 오름차순으로 정렬한 데이터베이스

ODB-2(Object DataBase-2) : 최소지지도를 고려하여 ODB-1을 스캔한 후의 데이터베이스

I : 항목들의 집합

N : 항목들의 수
n-항목집합 데이터베이스 : 구매항목수가 n인 트랜잭션들을 모아 놓은 데이터베이스
k-항목집합 : k개의 항목들을 갖는 항목집합

3.1.2 제안하는 알고리즘에 관한 정의

정의 1. Closed 항목집합
- TDB에서 어떤 항목집합 I에 대해, I를 포함하고 있는 트랜잭션들만을 대상으로 공통으로 들어있는 항목들을 찾을 때 I의 원소로 구성된 항목들 이외에는 공통으로 들어있는 항목이 없다면 그 항목집합 I를 CI(Closed Itemset)라고 한다. 특히 k개의 항목들을 원소로 갖는 CI를 k-CI라고 한다. 만약 CI의 지지도가 최소지지도보다 크거나 같으면 FCI(Frequent Closed Itemset)라고 부른다. 특히 k개의 항목들을 갖는 FCI를 k-FCI라고 한다.

정의 2. k-블록
- k-CI의 집합이다. 특히 closed 항목집합을 표현한 그림을 트랜잭션 연결 구조라고 한다. <Figure 1>에 트랜잭션 연결 구조의 예가 나와 있다.

정의 3. 상위항목집합과 하위항목집합
- 두 항목집합 X와 Y에 대해, X의 구매항목수가 Y의 구매항목수보다 클 때, X를 상위항목집합이라고 하고 Y를 하위항목집합이라고 한다.

정의 4. 진입항목집합과 기존항목집합
- 진입항목집합은 다른 기존항목집합들과의 교차를 위해 ODB-2에서 읽어들인 항목집합을 말하고, 기존항목집합은 읽어들인 진입항목집합과 교차해야 하는 대상이 되는 나머지 항목집합이다. 즉, 기존항목집합은 이미 트랜잭션 연결 구조에 존재하고 있으며 진입항목집합은 트랜잭션 연결 구조에 있는 모든 기존항목집합과 교차하여 공통항 목집합을 발견한다.

정의 5. 포함관계
- 진입항목집합 X와 기존항목집합 Y에 대해서, $X \supset Y$이면 완전포함관계이고, $X \cap Y \neq \varnothing$이면서 $X \not\supset Y$이면 불완전포함관계이며, $X \cap Y = \varnothing$이면 배반관계이다.

3.1.3 트랜잭션 연결 구조

<Table 3>을 TDB라고 하면, TDB를 스캔한 후의 ODB-1은 <Table 4>와 같다. 최소지지도는 1이라고 하자. 여기에서 최소지지도 1은 횟수를 의미한다. 최소지지도가 1이므로 이 예제에서는 결과적으로 ODB-1과 ODB-2가 같다. <Table 4>를 스캔하여 트랜잭션 연결 구조로 표현하면 <Figure 1>과 같다. 블록은 같은 구매항목수를 갖는 트랜잭션을 모아 놓은 것이다. 트랜잭션 연결 구조는 각 블록에 속한 트랜잭션들을 서로 연결해 놓은 그림이다. 트랜잭션 연결 구조를 작성하는 이유는 closed 항목집합의 지지도를 계산하기 위함이다. 트랜잭션 연결 구조에 있는 모든 트랜잭션에는 지지도가 계산되어 있다. 트랜잭션 옆에 있는 숫자가 그 트랜잭션의 지지도이다. 예를 들어, ABC3은 ABC라는 트랜잭션이 전체 데이터베이스에서 3번 나온다는 의미이다. 트랜잭션 연결 구조를 이용하여 최소지지도를 만족하는 빈발 closed 항목집합을 즉시 발견할 수 있다. 그리고 각각의 트랜잭션은 같은 구매항목수를 갖는 n-블록 ($n \geq 2$)에 저장된다.

Table 3. Database-2

TID	Items
1	A C T
2	M T R
3	A M S T R
4	A M S R
5	M T R

Table 4. ODB-1 after scanning Database-2

TID	Items	Support
1	A C T	1
2	M T R	2
3	A M S R	1
4	A M S T R	1

<Table 4>에서 먼저 ACT를 읽어서 트랜잭션 연결 구조에 삽입한다. ACT는 구매항목수가 3이므로 3-블록에 속하며 트랜잭션 수는 1이다. 2번째 트랜잭션인 MTR을 읽어서 3-블록에 넣은 후에 이미 읽어들인 ACT와 교차한다. 여기서 방금 읽은 MTR을 진입항목집합이라고 하고, 교차 대상이 되는 ACT를 기존항목집합이라고 한다. 두 트랜잭션은 불완전포함관계이면서 공통항목이 T밖에 없다. T는 1-항목집합이므로 트랜잭션 연결 구조에 표현하지 않는다. 즉, <Table 3>을 스캔하면서 1-항목집합의 지지도가 이미 계산되었기 때문에 필요가 없다. 지금까지는 ACT의 트랜잭션 수는 1이고 MTR의 트랜잭션 수는 2이다.

3번째 트랜잭션을 읽어서 4-블록에 넣고, 이미 트랜잭션 연결 구조에 삽입된 두 트랜잭션인 ACT, MTR과 차례로 교차한다. AMSR과 ACT를 교차하면 공통항목이 역시 A밖에 없으므로 의미가 없다. 그러나 AMSR과 MTR을 교차하면 공통항목이 MR로서 새로운 closed 항목집합이 생성되고, 공통항목이 두 개이므로 2-블록에 삽입한다. MR의 트랜잭션 수는 AMSR의 트랜잭션 수(1)와 MTR의 트랜잭션 수(2)를 더한 3이 된다. 그리고 새로 생성된 2-블록의 MR은 기존항목집합인 MTR과 연결한다. 즉, 2-블록의 MR은 3-블록의 MTR의 부분집합이며 완전포함관계가 된다.

마지막으로 5-블록으로 AMSTR을 읽어서 진입항목집합으로 두고 이미 삽입된 4-블록, 3-블록, 2블록의 기존항목집합과 차례로 교차를 시작한다. AMSTR과 AMSR은 완전포함관계이

며 따라서 연결한다. AMSTR의 트랜잭션 수는 1이지만 AMSR은 트랜잭션 수가 2가 된다. 즉, AMSR은 자신의 트랜잭션 수(1)에 AMSTR의 트랜잭션 수(1)를 더하는 것이다. 이번에는 AMSTR과 ACT를 교차하는데 공통항목이 AT로서 생성되므로 2-블록에 삽입하고 트랜잭션 수는 2가 된다. AT는 ACT와 연결한다. 또, AMSTR과 MTR은 완전포함관계이므로 연결하고 MTR의 트랜잭션 수를 2에서 3으로 증가시킨다.

여기서 주의할 점은 MTR과 이미 연결되어 있는 MR은 자동적으로 트랜잭션 수를 1만큼 증가시킨다는 것이다. MR은 MTR의 부분집합이므로 AMSTR과 따로 교차할 필요 없이 AMSTR의 트랜잭션 수를 더하기만 하면 된다. 즉, 연결 구조로 많이 표현될수록 교차의 횟수가 줄어들게 되며 연결되어 있는 하위항목집합의 트랜잭션 수만 갱신해주면 된다. 이러한 과정을 트랜잭션 연결 구조로 표현하면 <Figure 1>과 같다. 이 예제에서는 한 번만 구매해도 빈발이라고 가정했으므로 <Figure 1>의 각 블록에 있는 모든 트랜잭션들은 모두 빈발 closed 항목집합이 된다. 즉, 2-FCI는 MR, AT이고 3-FCI는 ACT, MTR이며 4-FCI와 5-FCI는 각각 AMSR, AMSTR이다. 1-FCI는 트랜잭션 연결 구조에서 구할 수 없고 트랜잭션 연결 구조의 FCI와 빈발 1-항목집합의 지지도를 비교하여 발견한다.

트랜잭션 연결 구조의 특징은 트랜잭션과 트랜잭션 사이에 연결된 형태가 많이 표현될수록 교차하는 횟수가 줄어든다는 것이다. 트랜잭션 연결 구조는 ODB-2의 모든 트랜잭션뿐만 아니라 트랜잭션간의 교차에 의해 만들어지는 새로운 트랜잭션들도 포함한다. <Figure 1>에 있는 점선으로 된 연결선은 연결된 트랜잭션간의 완전포함관계를 나타낸다. <Figure 1>에서는 모든 closed 항목집합이 최소지지도 1을 만족하기 때문에 모두 빈발 closed 항목집합이 된다. 트랜잭션 연결 구조가 완성되면, 사용자가 최소지지도를 바꾸어 가면서 빈발 closed 항목집합을 찾는 것이 용이해진다.

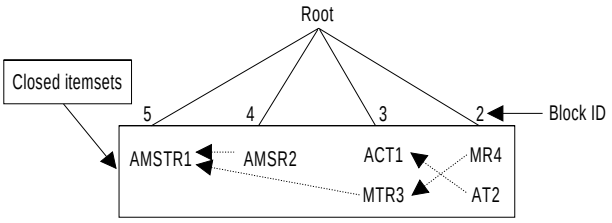

Figure 1. Transaction link structure-1.

3.2 알고리즘의 단계

제안하는 알고리즘은 FCILINK(Frequent Closed Itemsets using LINK)라고 하며, 두 단계로 나누어진다. 첫 번째 단계에서는 트랜잭션 데이터베이스(TDB)를 한 번 스캔하여 1-항목집합의 지지도를 계산함과 동시에 구매항목수가 작은 것부터 오름차순으로 정렬한 데이터베이스(ODB-1)를 생성한다. 각 트랜잭션 내에서의 항목들의 정렬과 각 블록 내에서의 트랜잭션들의 정렬은 알파벳 순서를 따르는 것으로 가정한다. 구매항목이 같은 트랜잭션들은 TID 뒤에 괄호를 사용하여 트랜잭션의 지지도를 표현한다. 하나의 항목만을 구매한 트랜잭션에 대해서는 1-항목집합 데이터베이스를 따로 구성하지 않는다. 1-항목집합은 다른 어떤 항목집합과 교차를 해도 자기 자신만이 공통인 항목으로 나오거나 배반관계만 만들어지게 되며 또한 1-항목집합의 지지도는 이미 계산이 되어 있기 때문에 고려하지 않는다.

따라서 ODB-1은 원래 주어진 TDB보다 데이터베이스의 크기가 작아지게 되며, 주어진 TDB에 항목을 하나만 구입한 고객이 많을수록 그리고 구매항목이 같은 트랜잭션들이 많을수록 더욱 효과적이다. ODB-1을 스캔하여 최소지지도보다 작은 지지도를 갖는 1-항목집합을 삭제하면서 구매항목수가 작은 것부터 오름차순으로 재정렬한다. 이렇게 만들어진 데이터베이스가 ODB-2이다. 다음부터의 데이터베이스 스캔은 TDB보다 데이터베이스의 크기가 작아진 ODB-2를 상대로 이루어지므로 스캔 시간이 줄어든다. 대용량 데이터로 인한 메모리 부하를 방지하기 위해 ODB-2를 n-항목집합($2 \le n \le N$) 데이터베이스로 분할한다.

두 번째 단계에서는 ODB-2를 스캔하여 읽어들인 트랜잭션을 해당 블록에 삽입하는데, 항목들을 나열하고 나열된 항목의 끝에 트랜잭션 수를 적는다. i-블록($i \ge 3$)으로 읽어들인 트랜잭션은 진입항목집합으로서 먼저 (i-k)-블록($1 \le k \le i$-2)의 기존항목집합과 공통항목을 찾기 위해 교차하고 마지막에 자신의 블록의 다른 기존항목집합과 교차한다. 1-항목집합 데이터베이스를 만들지 않기 때문에 1-블록은 생성되지 않으며 따라서 2-블록으로 읽어들인 트랜잭션들은 교차의 대상이 없다. 또한 2-블록의 트랜잭션은 자신의 블록에 속한 다른 트랜잭션과도 교차하지 않는다. 두 트랜잭션을 비교하여 완전포함관계이면 연결한다. 완전포함관계를 연결형으로 표현하여 교차의 횟수를 감소시킨다. 공통항목이 1개이면 무시한다.

i-블록의 진입항목집합이 (i-1)-블록의 기존항목집합과 완전포함관계이면 기존항목집합과 연결된 (i-k)-블록($2 \le k \le i$-2)의 기존항목집합과는 공통항목을 찾지 않고 연결된 모든 기존항목집합들의 트랜잭션 수를 진입항목집합의 트랜잭션 수만큼 증가한다. 공통항목은 있으나 완전포함관계를 이루지 못하는 불완전포함관계이면 공통항목집합을 공통항목의 개수에 해당하는 블록에 적고 진입항목집합의 트랜잭션 수와 기존항목집합의 트랜잭션 수를 합산한 결과를 공통항목집합의 트랜잭션 수로 취한 후 공통항목집합을 기존항목집합과 연결한다. ODB-2의 모든 트랜잭션을 트랜잭션 연결 구조에 삽입하고 스캔이 끝나면 최소지지도를 만족하지 못하는 closed 항목집합을 제거한다.

알고리즘이 종료하기 직전에, 발견된 k-FCI($2 \le k \le N$)와 빈발 1-항목집합을 비교하여 closure 개념에 위배되는 모든 빈발 1-항목집합을 제외하고 남은 빈발 1-항목집합을 빈발 closed 항목집합으로 선별한다. 즉 빈발 1-항목집합의 지지도와 발견된 k-FCI($2 \le k \le N$)의 지지도가 같으면 그 빈발 1-항목집합은

빈발 closed 항목집합이 될 수 없다. 제안하는 알고리즘은 주어진 트랜잭션 데이터베이스보다 크기가 작아진 데이터베이스를 대상으로, ODB-2의 모든 트랜잭션들을 closed 항목집합으로 가정하고 마이닝한다. <Figure 2>에 제안하는 알고리즘의 단계가 표현되어 있다.

3.3 수치 예제

<Table 5>와 같이 데이터베이스가 주어져 있고 최소지지도는 2라고 가정하자. 최소지지도는 비율이 아닌 횟수로 사용한다. 즉, 전체 10개의 트랜잭션 중에서 2개 이상의 트랜잭션에서 발생하면 된다. <Table 5>를 한 번 스캔하여 구매항목수가 작은 것부터 오름차순으로 정렬한 것이 <Table 6>에 나타나 있다. 구매항목이 같은 트랜잭션은 괄호를 사용하여 지지도를 표현한다. <Table 5>에서 TID 3번과 5번 트랜잭션은 같은 항목을 구매한 것이므로 <Table 6>에서는 TID 4번에 괄호를 사용하여 지지도를 2라고 표현한 것이다. 항목 A, B, C, E를 한꺼번에 구입한 고객이 두 명이라는 의미이다. 따라서 <Table 6>에서 TID 뒤에 괄호가 없는 것은 지지도가 1이다.

또한 <Table 5>를 <Table 6>으로 변환하면서 각 항목의 지지도도 계산한다. <Table 7>은 1-항목집합의 지지도를 나타낸 것이다. 1-항목집합의 지지도가 모두 계산되므로 <Table 6>에는 항목을 하나만 구입한 트랜잭션은 표시되지 않는다. <Table 5>에 항목을 하나만 구입한 고객이 많을수록 그리고 구매항목이 같은 트랜잭션들이 많이 중복해서 발생할수록 <Table 6>의 크기는 작아진다. ODB-1은 메모리를 효율적으로 관리하기 위해 n-항목집합 데이터베이스로 분할한다. <Table 6>에서는 항목을 2개 구입한 고객에서부터 5개 구입한 고객까지 있으므로 2-항목집합 DB, 3-항목집합 DB, 4-항목집합 DB, 5-항목집합 DB로 분할한다. ODB-1을 대상으로 최소지지도를 고려하여 비빈발 항목을 모두 제거하고 재정렬한 것이 <Table 8>의 ODB-2이다.

Table 5. TDB

TID	Items
1	A C D
2	B C D F
3	A B C E
4	B E
5	A B C E
6	B C E
7	A C D E
8	D F I J K
9	F I J K M
10	N

Table 6. ODB-1

TID	Items
1	B E
2	A C D
3	B C E
4(2)	A B C E
5	A C D E
6	B C D F
7	D F I J K
8	F I J K M

Input : A transaction database and minimum support
Output : The complete sets of frequent closed itemsets
Method :
1. Scan TDB
   1-1. Calculate the support of a 1-itemset
   1-2. Generate a sorted database called ODB-1 in terms of transaction size
   1-3. Remove infrequent 1-itemsets from ODB-1
   1-4. Generate a re-sorted database ODB-2
   1-5. Divide ODB-2 into n-itemset databases($2 \le n \le N$)
2. Scan ODB-2
   2-1. Insert the transactions of ODB-2 into corresponding blocks in the transaction link structure
   2-2. Intersect an entering itemset with other existing itemsets
      Case 1. Perfect inclusion relation
      - Link transactions with each other
      - Add the support of the entering itemset to the support of linked lower itemsets
      Case 2. Imperfect inclusion relation
      - Link the intersecting itemset to the existing itemset
      - Calculate the support of the intersecting itemset by adding the support of the entering itemset to the support of the existing itemset
   2-3. Construct the transaction link structure
   2-4. Find frequent closed k-itemsets($2 \le k \le N$)
3. Find the frequent closed 1-itemsets by comparing the discovered k-FCI with the frequent 1-itemsets
4. Exit

Figure 2. The FCILINK algorithm.

Table 7. The support of 1-itemsets

A	B	C	D	E	F	I	J	K	M	N
4	5	6	4	5	3	2	2	2	1	1

Table 8. ODB-2

TID	Items
1	B E
2	A C D
3	B C E
4(2)	A B C E
5	A C D E
6	B C D F
7	F I J K
8	D F I J K

최소지지도가 고려된 <Table 8>을 대상으로 구매항목수가 작은 트랜잭션부터 위에서 아래로 하나씩 읽어서 해당 블록에 삽입한다. 즉, <Table 8>에서 구매항목수가 2인 1번 트랜잭션은 2-블록에 BE1의 형태로 저장된다. 트랜잭션 수는 항목을 나열하고 마지막에 기록한다. 다음에 ACD1을 3-블록에 삽입하고 이미 2-블록에 저장된 BE1과 포함관계를 조사한다. 여기서 방금 읽어들인 ACD1을 진입항목집합이라고 하고, 공통항목이 존재하는지를 파악하기 위해 교차시킬 BE1을 기존항목집합이라고 한다. 두 항목집합의 교차 결과는 배반관계이므로 상호간의 연결을 하지 않으며 어떠한 트랜잭션 수의 증가도 없다.

이번에는 3-항목집합 데이터베이스의 BCE1을 읽어서 3-블록에 삽입하여 진입항목집합으로 정하고 이미 저장되어 있는 BE1과 ACD1을 기존항목집합으로 인식하여 각각 교차시킨다. 진입항목집합이 3-블록에 있으므로 먼저 2-블록에 있는 기존항목집합인 BE1과 교차하고 다음에 3-블록의 기존항목집합인 ACD1과 교차한다. 즉, i-블록($i \geq 3$)에 있는 진입항목집합은 (i-k)-블록($1 \leq k \leq i$-2)의 기존항목집합과 차례대로 교차시켜서 공통항목을 찾고 마지막으로 자신이 속한 블록의 다른 기존항목집합과 교차한다. BCE1과 BE1은 포함관계가 기존항목집합이 진입항목집합에 포함되는 완전포함관계이므로 두 항목집합을 연결하고 기존항목집합인 BE1의 지지도를 하나 증가하여 BE2로 바꾼다. BCE1과 ACD1은 공통항목이 C로서 하나의 항목만이 교차하므로 무시한다.

4-블록의 ABCE2를 진입항목집합으로 정하여 3-블록의 BCE1과 먼저 교차하면 완전포함관계가 되므로 연결시키고 ABCE2의 지지도 2를 BCE1에 더하여서 BCE3으로 바꾼다. BE2는 이미 BCE3에 연결되어 있으므로 교차할 필요 없이 바로 지지도 2만큼을 증가하여 BE4로 만든다. ABCE2와 ACD1을 교차하면 AC가 공통으로 나오므로 2-블록에 새로운 closed 항목집합으로 삽입하고 지지도는 ABCE2와 ACD1의 지지도를 더한 3을 취한다. 따라서 2-블록에 AC3이라고 하는 closed 항목집합이 만들어진다. 이와 같은 방식으로 <Table 8>의 모든 트랜잭션을 읽어서 트랜잭션 연결 구조에 저장한다.

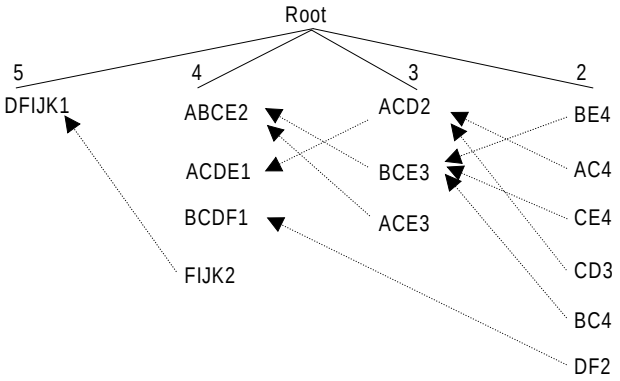

Figure 3. Transaction link structure-2.

이렇게 만들어진 트랜잭션 연결 구조가 <Figure 3>이다. <Figure 3>의 각 블록에 저장된 항목집합들은 모두closed 항목집합이다. 항목집합끼리의 연결이 많아질수록 교차하는 빈도가 줄어든다. 연결되어 있는 두 개의 항목집합들은 상호간에 지지도가 같을 수가 없으며 구매항목수가 더 큰 블록의 항목집합이 구매항목수가 더 작은 항목집합의 지지도보다 작다. 주어진 최소지지도 2를 만족하는 모든 빈발 closed 항목집합이 <Figure 4>에 있다.

1-FCI : B5, C6, D4, E5, F3
2-FCI : BE4, AC4, CE4, CD3, BC4, DF2
3-FCI : ACD2, BCE3, ACE3
4-FCI : ABCE2, FIJK2

Figure 4. Frequent closed itemsets.

빈발 closed 1-항목집합은 <Figure 3>과 <Table 7>을 사용하여 지지도와 포함관계를 동시에 고려하면서 발견한다. <Figure 3>의 어떤 closed 항목집합이 <Table 7>의 특정한 1-항목집합을 부분집합으로 가지면서 서로 지지도가 같으면 그 1-항목집합은 빈발 closed 항목집합이 될 수 없다. 예를 들어, <Table 7>의 {A}는 지지도가 4인데, <Figure 3>의 {AC4}와 지지도가 같기 때문에 1-항목집합인 {A}는 빈발 closed 항목집합이 아니다. 즉, {AC4}만 알고 있으면 {A}의 지지도가 4라는 것을 알 수 있기 때문에 발견할 필요가 없다.

3.4 계산량

항목들의 수를 m이라고 할 때, 모든 빈발 항목집합을 나열하기 위해 필요한 탐색 공간은 2^m이며 이는 지수적으로 증가함을 의미한다(Tan *et al*., 2006). 특정한 크기의 빈발 항목집합을 발견하는 문제는 NP-complete로 알려져 있다(Fabrizio *et al*., 2004). 복잡성 이론에서는 알고리즘의 속도가 다항식으로 표현되는

문제들을 'P'라고 부르고, 다항식으로 표현될 수 있는지 여부가 알려지지 않은 문제들을 'NP'라고 부른다. 제안하는 알고리즘의 계산량은 다음과 같은 요인에 의해 영향을 받게 된다.

1) 최소지지도 : 최소지지도가 낮게 주어질수록 더 많은 빈발 항목집합이 발견된다. 빈발 항목집합의 최대 크기 역시 낮은 최소지지도에서 더 커지는 경향이 있다.
2) 항목들의 수 : 항목들의 수가 증가하면 저장하는 공간이 더 많이 필요하게 되고 입출력(I/O) 비용도 증가하게 된다.
3) 트랜잭션들의 수 : 트랜잭션들의 수가 증가하면 저장 공간이 많이 필요하게 되고 트랜잭션끼리 교차하는 횟수가 늘어나게 된다.
4) 평균 트랜잭션 크기 : 특히 밀집한 데이터 집합에서는 평균 트랜잭션 크기가 커질 수 있다. 평균 트랜잭션 크기가 커지면 빈발 항목집합의 최대 크기도 커지는 경향이 있다. 이는 지지도를 계산할 때 더 많은 탐색을 필요로 한다.

주어진 트랜잭션 데이터베이스에서 트랜잭션의 수를 n이라고 하고, 가장 큰 구매항목수를 갖는 빈발 closed 항목집합의 크기를 k라고 하고, 빈발 closed 항목집합의 수를 r이라고 할 때 제안하는 알고리즘을 사용하여 빈발 항목집합을 발견하는 계산량은 $O(r \cdot n \cdot 2^k)$가 된다. 어떤 항목집합의 크기가 p일 때 $2^p - 2$개의 연관규칙이 생성된다. 공집합과 전체집합을 제외한, 그 항목집합의 모든 부분집합들을 조건부에 위치시킬 수 있기 때문이다. 따라서 연관규칙을 생성하는 과정에서의 계산량은 빈발 항목집합의 수를 f라고 하고 가장 긴 빈발 항목집합의 길이를 l이라고 할 때 $O(f \cdot 2^l)$가 된다.

일반적인 연관규칙 생성 방법으로 접근하면 연관규칙의 수는 다음과 같다.

$$\sum_{i=0}^{l}\binom{l}{i}\cdot 2^{l-i} \le \sum_{i=o}^{l}\binom{l}{i}\cdot 2^{l} = 2^{l}\sum_{i=0}^{l}\binom{l}{i} = 2^{l}\cdot 2^{l} = O(2^{2l})$$

반면에 closure 개념을 사용하여 연관규칙의 수를 구한다면 최악의 경우 2^l개의 빈발 closed 항목집합을 가정할 수 있다. 이 경우에는 신뢰도가 100%인 연관규칙이 생성되지 않는다. 이 때 생성되는 연관규칙의 수는 다음과 같다.

$$\sum_{i=0}^{l}\binom{l}{i}\cdot (l-i) \le \sum_{i=o}^{l}\binom{l}{i}\cdot l = O(l\cdot 2^{l})$$

4. 실험 결과 및 분석

제안하는 알고리즘의 성능을 평가하기 위해<Table 9>와 같은 데이터 집합을 사용하여 실험을 수행했다. C언어로 프로그램을 코딩하고, 실험은 CPU 550MHz, 메모리 256MB를 가진 컴퓨터를 사용했다. 트랜잭션의 수와 항목의 수를 증가시켜 보고, 최소지지도를 바꾸어 가면서 실험했다. <Table 9>에서 T는 트랜잭션들의 평균 크기이고, I는 항목들의 수이며, D는 주어진 데이터베이스의 전체 트랜잭션들의 수를 말한다.

Table 9. Database characteristics

Database	T	I	D
T20I100D100K	20	100	100,000
T20I100D10K	20	100	10,000
T10I1000D100K	10	1,000	100,000
T3I500D60K	3	500	60,000
T3I20000D100K	3	20,000	100,000
T3I40000D800K	3	40,000	800,000

예를 들어, 'T20I100D10K'는 평균 트랜잭션 크기가 20이므로 고객들이 평균 20개의 항목을 한꺼번에 구입하고, 고객들이 구매할 수 있는 항목들이 100가지 종류가 있으며, 전체 트랜잭션들의 수는 10,000개라는 것을 의미한다. 여기서 K는 1,000 단위를 뜻한다. <Figure 5>부터 <Figure 10>까지 각 데이터 집합을 사용하여 실험한 결과를 보여준다.

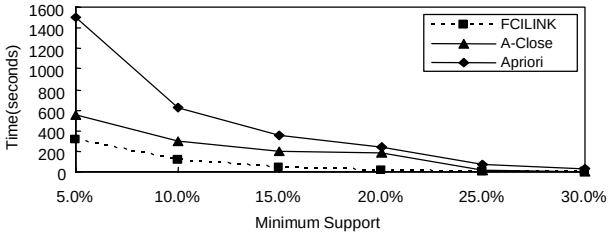

Figure 5. Scalability with the support threshold for dataset T20I100D100K.

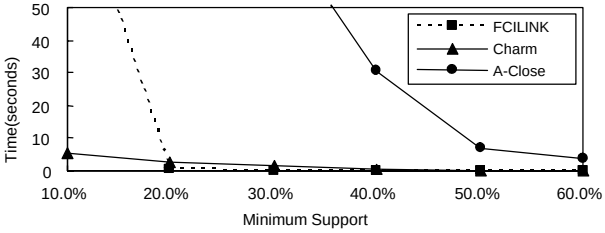

Figure 6. Scalability with the support threshold for dataset T20I100D10K.

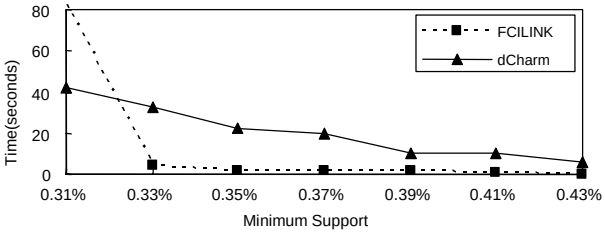

Figure 7. Scalability with the support threshold for dataset T10I1000D100K.

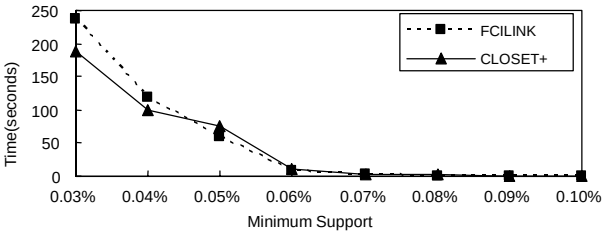

Figure 8. Scalability with the support threshold for dataset T3I500D60K.

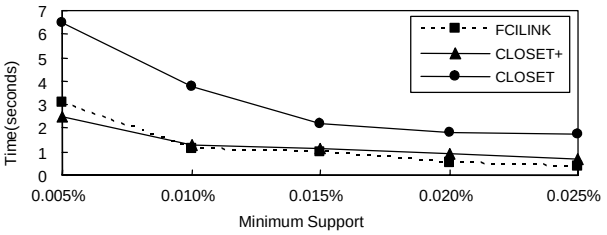

Figure 9. Scalability with the support threshold for dataset T3I20000D100K.

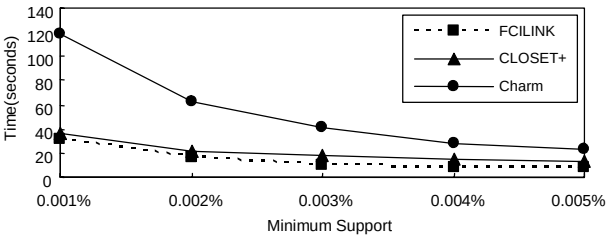

Figure 10. Scalability with the support threshold for dataset T3I40000D800K.

<Figure 5>는 밀집한(dense) 데이터 집합의 특성을 가지고 있는 T20I100D100K 데이터 집합에 대하여 FCILINK와 기존의 Apriori, A-Close 알고리즘을 비교했다. FCILINK는 A-Close에 비해서 최소지지도에 따라 2배에서 18배까지 수행시간을 단축시킨다. <Figure 6>은 UCI 기계 학습 데이터베이스 실험실의 'mushroom' 데이터 집합과 유사한 데이터 집합인 T20I100D10K 데이터 집합으로 실험한 결과이다. 'mushroom' 데이터 집합은 평균 트랜잭션 크기가23이고 항목들의 수가120이고 트랜잭션들의 수는 8,124개이다. 반면에 T20I100D10K는 트랜잭션들의 수가 10,000개로서 조금 더 많지만 평균 트랜잭션 크기는 20이고 항목들의 수는 100으로 약간 더 적다. <Figure 6>에서는 FCILINK를 A-Close, Charm 알고리즘과 비교한다. 최소지지도가 증가할수록 FCILINK는 A-Close와 Charm에 비해서 더 좋은 수행도를 보여준다.

<Figure 7>은 IBM Almaden 연구소에서 만들어진 희박한(sparse) 데이터 집합인 T10I1000D100K 데이터 집합을 가지고 dCharm 알고리즘과 비교했다. 희박한 데이터 집합에서는 거의 대부분의 빈발 항목집합들이 빈발closed 항목집합이 된다. 이 경우에는 교차가 많이 발생하기 때문에 최소지지도가 낮을수록 dCharm 알고리즘에 비해 성능이 떨어지는 경향을 보인다. 연관규칙을 활용하는 전형적인 분야가 백화점이나 대형마트를 대상으로 하는 장바구니 분석인데(Brin *et al.*, 1997; Silverstein *et al.*, 1998), 여기에서 항목은 상품을 의미하고 트랜잭션은 판매시점(point-of-sales) 데이터가 된다. 이러한 데이터베이스는 일반적으로 희박한 데이터 집합이어서 빈발 항목집합의 크기가 상대적으로 작다.

<Figure 8>은 클릭스트림 데이터로 이루어진 데이터 집합을 사용하여 FCILINK와 CLOSET+ 알고리즘을 비교한 실험결과를 보여준다. 데이터 집합의 특성으로 인해 특정한 최소지지도에서 급격히 빈발 closed 항목집합의 수가 줄어들었다. 비교하는 두 알고리즘은 전체적으로 비슷한 수행도를 나타내고 있다. <Figure 9>와 <Figure 10>의 실험에 사용된 데이터 집합은 항목이나 트랜잭션의 수가 많지만 트랜잭션들의 평균 크기가 3개로 작기 때문에 트랜잭션들끼리 교차하는 데 걸리는 시간보다는 데이터베이스를 최초에 한 번 스캔하는 데 걸리는 시간이 전체 수행시간의 대부분을 차지했다. 기존의 CLOSET+, CLOSET, Charm 알고리즘과 비교하여 제안하는 알고리즘이 보다 좋은 수행도를 보여준다.

실험 결과에 의하면 제안하는 알고리즘은 특히 밀집한 데이터 집합에서 좋은 수행도를 보여주었다. 밀집한 데이터 집합에서는 트랜잭션들 사이의 연결이 많아지므로 교차의 횟수가 줄어들게 되어 단지 지지도의 증가분만 계산하면 된다. 일반적으로 FCILINK는 최소지지도를 증가할수록 효율적으로 동작한다. 주어진 데이터베이스를 한 번 스캔하면서 빈발1-항목집합을 발견한 다음, 비빈발 항목들을 제거하는 과정 없이 진행되고, 알고리즘의 마지막에 최소지지도를 고려한다. 그러므로 낮은 최소 지지도에서는 알고리즘의 수행도가 떨어질 수 있다.

실험 결과는 제안하는 알고리즘이 기존의 알고리즘보다 전반적으로 더 좋은 성능을 나타냄을 보여준다. 첫째, ODB-2는 TDB보다 작아지게 되고 ODB-2를 한 번 스캔함으로써 트랜잭션 연결 구조를 만들 수 있기 때문에 스캔 시간을 줄일 수 있다. 제안하는 알고리즘은 하나의 항목만을 구매한 고객이 많거나 구매한 항목이 같은 고객이 많은 데이터베이스에서 효과적이다. 둘째, 트랜잭션 자체를 이용하여 트랜잭션 연결 구조를 만든다. 트랜잭션들 간의 연결이 많아질수록 교차의 횟수는 줄어든다. 하지만 Charm 알고리즘은 항목집합과 트랜잭션 번호집합을 동시에 고려한다. 셋째, 빈발 closed 항목집합을 찾기 위해 트랜잭션 자체를 이용하기 때문에 항목들 사이의join 과정이 없다. 그러나 A-Close 알고리즘은 Apriori 기반의 join 과정이 필요하다.

FCILINK는 A-Close나 Apriori보다 더 좋은 수행도를 보여주지만 <Figure 6>과 <Figure 7>에서 볼 수 있는 것처럼 낮은 최소지지도를 가지고 패턴을 발견하는 것은 여전히 비용이 많이 든다. 이는 많은 수의 2-항목집합이 생성되기 때문이다 2-항목집합의 수가 많아지면 트랜잭션들 간의 교차의 횟수가 증가하게

된다. 이러한 문제를 해결하기 위해 추후 연구가 필요하다 추후 연구 방향은 해시 테이블을 이용하여2-항목집합의 수를 효과적으로 줄이는 연구를 진행함과 동시에, 제안하는 알고리즘을 웹에서 활용하여 웹 마이닝으로의 응용을 모색할 예정이다

5. 결 론

본 논문에서는 트랜잭션 자체를 이용하여 효율적으로 빈발 closed 항목집합을 발견하는 알고리즘을 제시했다. 트랜잭션들을 연결 구조로 표현하여 공통항목을 찾기 위한 교차의 횟수를 감소시키며 최종적으로 완성된 트랜잭션 연결 구조에는 모든 closed 항목집합이 지지도까지 전부 계산되어 저장된다. 트랜잭션 연결 구조가 완성되면 최소지지도를 알고리즘의 마지막에 고려하여 다양한 최소지지도 기준에 의한 빈발closed 항목집합을 발견할 수 있다. 동일하거나 비슷한 트랜잭션들이 많은 데이터베이스나 하나의 항목만을 구입한 트랜잭션들이 많은 데이터베이스에 특히 효과적이다. 실험 결과는 제안하는 알고리즘이 기존의 방법보다 마이닝 시간을 단축시키며 특히 밀집한 데이터에서 보다 나은 수행도를 보여준다.

참고문헌

Agrawal, R. and Srikant, R. (1994), Fast Algorithms for Mining Association Rules, In Proc. of the 20th Int'l Conference on Very Large Databases, Santiago, Chile, September, 487-499.

Agrawal, R., Imielinski, T., and Swami, A. (1993), Mining Association Rules between Sets of Items in Large Databases, In Proc. of the ACM SIGMOD Int'l Conference on Management of Data, Washington D.C., May, 207-216.

Bing, L., Wynne, H., and Yiming, M. (1998), Integrating Classification and Association Rule Mining, In Proc. of 4th Int. Conf. on Knowledge Discovery and Data Mining (KDD-98), New York, USA.

Brin, S., Mptwani, R., Ullman, J. D., and Tsur, S. (1997), Dynamic itemset counting and implication rules for market basket data, Proc. of the ACM SIGMOD Conference, May, 255-264.

Chen, M. S., Han, J., and Yu, P. S. (1996), Data Mining : An Overview from a Database Perspective, *IEEE Transactions on Knowledge and Data Engineering*, **8**(6), 866-883.

Fabrizio, A., Giovambattista, I., and Luigi, P. (2004), On the complexity of inducing categorical and quantitative association rules, *Theoretical Computer Science*, **314**, Issues 1-2, 25 February, 217-249.

Han, J. and Fu, Y. (1995), Discovery of Multiple-Level Association Rules from Large Databases, In Proc. of 1995 Int'l Conf. on Very Large Data Bases (VLDB'95), Zurich, Switzerland, September, 420-431.

Han, J. and Kamber, M. (2001), *Data Mining* : *Concepts and Techniques*, Morgan Kaufmann Publishers, San Francisco, CA.

Han, J., Pei, J., Yin, Y., and Mao, R. (2004), Mining Frequent Patterns without Candidate Generation : A Frequent-Pattern Tree Approach, *Data Mining and Knowledge Discovery*, **8**(1), 53-87.

Hong, T. P., Kuo, C. S., and Chi, S. C. (1999), Mining association rules from quantitative data, *Intelligent Data Analysis*, **3**, Issue 5, November, 363-376.

Hsu, P. L., Lai, R., and Chiu, C. C. (2003), The hybrid of association rule algorithms and genetic algorithms for tree induction : an example of predicting the student course performance, *Expert Systems with Applications*, **25**, Issue 1, July, 51-62.

Jiuyong, L., Hong, S., and Rodney, T. (2002), Mining the optimal class association rule set, *Knowledge-Based Systems*, **15**, Issue 7, 1 September, 399-405.

Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., and Verkamo, A. I. (1994), Finding interesting rules from large sets of discovered association rules, In Proc. 3rd Int'l Conf. on Information and Knowledge Management, Gaithersberg, Maryland, Nov., 401-408.

Lee, C. H., Kim, Y. H., and Rhee, P. K. (2001), Web personalization expert with combining collaborative filtering and association rule mining technique, *Expert Systems with Applications*, **21**, Issue 3, October, 131-137.

Michael, J. A. B. and Gordon, L. (1997), *Data Mining Techniques*, WILEY, U.S.A.

Mohammed, J. Z. and Karam, G. (2003), Fast Vertical Mining Using Diffsets, 9th International Conference on Knowledge Discovery and Data Mining, Washington, DC, August.

Pasquier, N., Bastide, Y., Taouil, R., and Lakhal, L. (1999a), Closed Set Based Discovery of Small Covers for Association Rules, Proc. BDA conf., October, 361-381.

Pasquier, N., Bastide, Y., Taouil, R., and Lakhal, L. (1999b), Discovering Frequent Closed Itemsets for Association Rules, In Proc. 7th Int. Conf. Database Theory (ICDT'99), Jerusalem, Israel, January, 398-416.

Pasquier, N., Bastide, Y., Taouil, R., and Lakhal, L. (1999c), Efficient mining of association rules using closed itemset lattices, *Information Systems*, **24**(1), 25-46.

Pasquier, N., Bastide, Y., Taouil, R., and Lakhal, L. (2000), Mining Minimal Non-Redundant Association Rules using Frequent Closed Itemsets, Proc. 1st conf. on Computational Logic, LNCS 1861, July, 972-986.

Pei, J., Han, J., and Mao, R. (2000), CLOSET : An Efficient Algorithm for Mining Frequent Closed Itemsets, In Proc. 2000 ACM-SIGMOD Int. Workshop on Data Mining and Knowledge Discovery (DMKD'00), Dallas, TX, May.

Pieter, A. and Dolf, Z. (1996), *DATA MINING*, Addison-Wesley, Harlow, U.K.

Rajeev, R. and Shim, K. S. (2001), Mining optimized support rules for numeric attributes, *Information Systems*, **26**, Issue 6, September, 425-444.

Silverstein, C., Brin, S., and Motwani, R. (1998), Beyond market baskets : Generalizing association rules to dependence rules, *Data Mining and Knowledge Discovery*, January, **2**(1), 39-68.

Srikant, R. and Agrawal, R. (1995), Mining Generalized Association Rules, In Proc. of the 21st Int'l Conference on Very Large Databases, Zurich, Switzerland, September, 407-419.

Srikant, R. and Agrawal, R. (1996), Mining Quantitative Association Rules in Large Relational Tables, Proc. of the ACM SIGMOD Conference on Management of Data, Montreal, Canada, June.

Srikant, R., Vu, Q., and Agrawal, R. (1997), Mining Association Rules with Item Constraints, In Proc. of the 3rd Int'l Conference on Knowledge Discovery in Databases and Data Mining, Newport Beach, California, August, 67-73.

Takeshi, F., Yasuhiko, M., Shinichi, M., and Takeshi, T. (1999), Mining Optimized Association Rules for Numeric Attributes, *Journal of Computer and System Sciences*, **58**, Issue 1, February, 1-12.

Tan, P. N., Steinbach, M., and Kumar, V. (2006), *INTRODUCTION TO DATA MINING*, Addison Wesley.

Tsay, Y. J. and Chien, Y. W. C. (2004), An efficient cluster and decomposition algorithm for mining association rules, *Information Sciences*, **160**, Issues 1-4, 22 March, 161-171.

Wang, J., Han, J., and Pei, J. (2003), CLOSET+ : Searching for the Best Strategies for Mining Frequent Closed Itemsets, Proc. 2003 ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'03), August.

Yan, X. and Han, J. (2003), CloseGraph : Mining Closed Frequent Graph Patterns, Proc. 2003 ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'03), August.

Zaki, M. J. and Hsiao, C. (1999), Charm : An efficient algorithm for closed association rule mining, *In Technical Report 99-10*, Computer Science Dept., Rensselaer Polytechnic Institute, October.

http://www.ics.uci.edu/~mlearn/MLRepository.html

http://www.almaden.ibm.com/software/projects/iis/hdb/Projects/data_mining/mining.shtml