단계 선형 배치 트리를 이용한 순차 패턴 추출

최현화[O] 이동하 이전영
한국전자통신연구원,㈜넷스루,포항공과대학교
hyunwha[O] @etri.re.kr
dongha@nethru.co.kr
jeon@postech.ac.kr

Mining Sequential Patterns Using Multi-level Linear Location Tree

Hyun Wha Choi[O]
Electronics and Telecommunications Research Institute
Dong Ha Lee
Nethru inc.
Jeon Young Lee
Intelligent Information Systems Lab., Pohang University of Science and Technology

요 약

대용량 데이터베이스로부터 순차 패턴을 발견하는 문제는 지식 발견 또는 데이터 마이닝(Data Mining) 분야에서 주요한 패턴 추출 문제이다. 순차 패턴은 추출 기법에 있어 연관 규칙의 Apriori 알고리즘과 비슷한 방식을 사용하며 그 과정에서 시퀀스는 해쉬 트리 구조를 통해 다루어 진다. 이러한 해쉬 트리 구조는 항목들의 정렬과 데이터 시퀀스의 지역성을 무시한 저장 구조로 단순 검색을 통한 다수의 복잡한 포인터 연산 수행을 기반으로 한다.본 논문에서는 이러한 해쉬 트리 구조의 단점을 보완한 다단계 선형 배치 트리(MLLT, Multi-level Linear Location Tree)를 제안하고, 다단계 선형 배치 트리를 이용한 효율적인 마이닝 메소드(MLLT-Join) 를 소개한다.

1. 서 론

대용량 데이터베이스로부터 순차 패턴(Sequential Pattern)[1][2] 을 발견하는 문제는 지식 발견(knowledge discovery) 또는 데이터 마이닝(data mining)분야에서 주요한 패턴 추출 문제이다. 순차 패턴은 트랜잭션 안에서 발생된 항목들 간의 연관 규칙에 시간 변이를 추가한 지식의 한 형태로, "CPU를 바꾼 사람의 80%는 다음에 메인 보드를 바꾼다" 와 같은 구매 패턴이 그 대표적인 예이다.

이러한 순차 패턴은 1995년 Agrawal에 의해 처음 소개되어, 고객의 구매 패턴 분석을 통한 구매 예측은 물론 의료 분야의 질병 발생 순서 패턴에 따른 진료 및 투약에 관한 정보 제공에 이르기까지 그 활용 범위가 점차 다양해져 가고 있다.

연관 규칙(Association Rule)과 비슷한 알고리즘을 사용하는 순차 패턴은 아직 상용화에 있어, 다른 데이터 마이닝 작업에 비해 초기 단계에 있는데, 이는 큰 후보 집합(Candidate Set)을 전제로 하기 때문에, 상용화에 있어 중요한 관건이 되는 성능에 문제점을 지니고 있기 때문이다.

초기 연구 단계에 있는 순차 패턴은 추출 과정에서의 성능 향상을 위해 많은 연구가 이루어져 왔으며, 대부분이 후보 집합의 축소 및 Join 알고리즘[2][4][5] 에 집중되고 있다.

본 논문에서는 순차 패턴 추출 시에 데이터 시퀀스(data sequence)의 저장 구조로 널리 사용되고 있는 해쉬 트리(Hash Tree)구조의 대표적인 단점인 데이터의 지역성 배제를 해결한 새로운 데이터 구조, 다단계 선형 배치 트리(MLLT, Multi-level Linear Location Tree)를 소개하고, 이를 이용한 마이닝 메소드(Join)를 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 간략히 살펴보고, 3장에서는 다단계 선형 배치 트리에 대해서 서술한다. 4장에서는 다단계 선형 배치 트리를 이용한 마이닝 메소드, Join 알고리즘에 대해 설명하고, 5장에서 실험 결과 및 결론을 맺는다.

2. 관련 연구

순차 패턴은 기본적으로 장바구니 분석(Market Basket Analysis)이라 불리우는 연관 규칙의 많은 개념을 기반으로 하고 있다. 항목 집합(set of items, itemset)들로 이루어진 데이터베이스에서 이들 항목의 동시 출현 성향 ㅁ 항목 a와 b가 같이 팔리는 성향-에 대한 관계성을 표현하는 연관 규칙은 트랜잭션에 함께 출현하는 관련성을 표현한 것으로, 지지도(support)와 신뢰도(confidence)의 두 매개변수를 사용한다. 지지도는 연관 규칙을 구성하는 항목 집합을 포함하는 트랜잭션이 전체 트랜잭션의 몇 퍼센트가 되는지를 나타내고, 신뢰도는 규칙의 성립 정도를 뜻한다.

$A \rightarrow B$

$$support = \frac{\text{number of transactions containing } A \cup B}{\text{number of total transactions}}$$

$$confidence = \frac{\text{number of transactions containing } A \cup B}{\text{number of transactions containing } A}$$

연관 규칙 추출 과정은 순차 패턴 추출 과정과 같으며, 입력 시퀀스가 아닌 항목 집합이라는데 차이가 있을 뿐이다.

연관 규칙의 빈번항목 집합(frequent item set)을 저장하기 위한 데이터 구조로 해쉬 트리를 대신하는 다단계 선순열 트리(MLST, Multi-level Linear Sequence Tree)[3]에 관한 연구가 있었다. MLST는 항목 집합들이 사전적인 순서로 정렬되어 있으며, 항목 집합의 앞부분이 다른 항목 집합과 일치하면 그 정보를 공유한다. MLST에서 항목 집합을 표현하는 모습은 그림 1과 같다.

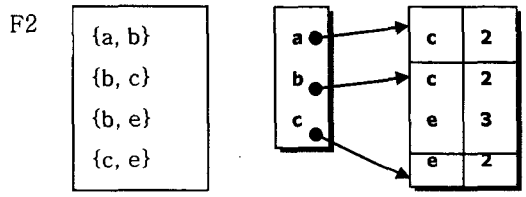


그림 1 F2를 표현한 MLST의 모습

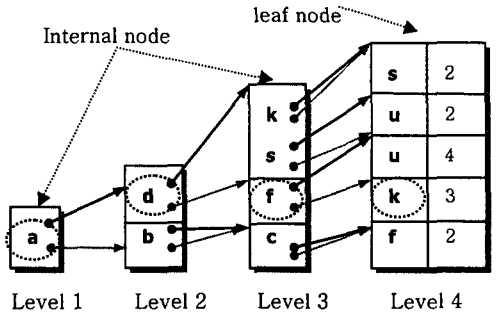
연관 규칙과 비교해 순차 패턴 추출은 후보 시퀀스 발생에서 시퀀스 안의 항목들의 중복을 허용하고, 원소들의 가능한 모든 순열을 포함하기 때문에 후보 항목 집합의 수가 훨씬 많이 발생한다. 그로 인해 오버플로우시 확장하는 동적 트리인 해쉬 트리는 주기억 장치에 저장되는 것을 전제로 하며, 버킷의 크기에 따라 기억 장소의 낭비 또는 버킷 내부에서의 긴 검색 시간을 초래한다. 뿐만 아니라 데이터에 관한 사전 지식이 없는 사용자에게 검색 시간을 좌우하는 버킷 크기에 관한 결정 권한이 주어지기 때문에 순차 패턴 추출의 성능이 수행시마다 많은 차이를 나타내기도 한다.

3. 다단계 선형 배치 트리 (MLLT)

다단계 선형 배치 트리(MLLT, Multi-level Linear Location Tree)는 순차 패턴을 추출하는 과정에서 k-번째 시퀀스 F_k , 또는 k-후보 시퀀스 C_k를 저장하는 데이터 구조이다. MLLT는 사전적인(lexicographical) 순서로 정렬된 항목들의 집합인 원소(element)들의 리스트, 즉 시퀀스를 표현하는데 사용된다.

MLLT의 구조를 보면, 시퀀스 간의 동일한 항목을 갖고 있는 경우 그 정보를 공유한다. <(a,d), f, u > <(a, d), f , k>의 두 시퀀스는 그림 2에서 보는 것과 같이 시퀀스의 앞부분 <(a, d), f> 을 공유하게 함으로써, 메모리 요구량은 물론 포인터 연산이 줄이는 효과를 가져온다.

또한, 시퀀스를 표현하는데 있어서 항목간의 관계에 따라 두 부분으로 나누어 처리한다. 두 항목이 한 원소의 일부로 연결되는 경우와 두 항목이 다른 원소로 연결되는 경우인데, 이러한 두 경우에 해당하는 항목들을 모아 사전적인 순서로 정렬한 다음 각각의 첫번째 항목을 가리키도록 한다. 예를 들어, <a, b> <a, c>는 두 항목이 서로 다른 원소에 속하므로, 두번째 항목 b, c를 모아 하나의 노드에 넣고 포인터가 맨 처음 항목 b를 가리키도록 한다. 그리하여, MLLT의 내부 노드(internal node)들은 모두 (item, cPtr, dPtr)의 자료 구조를 가진다. 즉, 한 개의 연속 항목(consecutive item)을 가리키는 포인터와 이산 항목(discrete item)을 가리키는 포인터를 가지는 것이다. 반면, 말단 노드(leaf node)는 내부 노드의 자료 구조에 support의 수를 저장하는 공간을 두어 (item, cPtr, dPtr, support) 의 형태를 가진다.



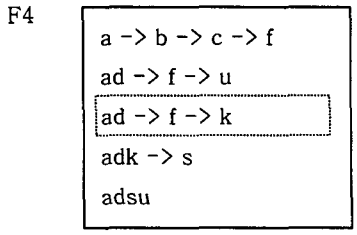


그림 2 4-빈번 시퀀스에 해당하는 MLLT 의 구조

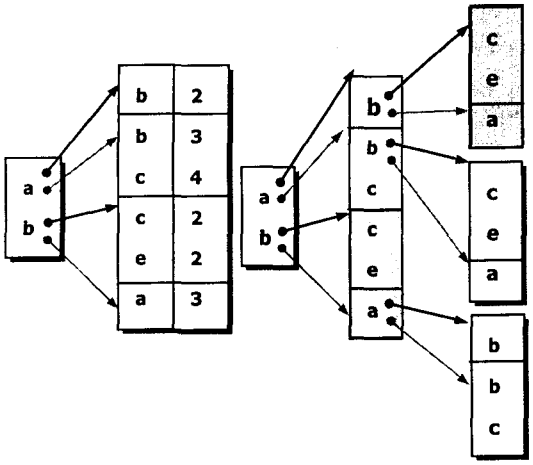
4. 마이닝 메소드(Mining Method)

MLLT를 이용한 순차 패턴 추출 과정에서의 조인(Join)은 기존의 조인 방법을 그대로 따른다. 그러나 그림 3에서 보는 것과 같이 훨씬 단순하다는 것을 확인할 수 있다.

비슷한 시퀀스들을 모아 놓은 결과를 갖게 되는 MLLT에서는 한번의 검색을 통해 S1과 조인하는 S2를 모두 찾을 수 있게 되고 두 노드를 복사하는 것만으로 조인 연산이 이뤄진다. 즉,S1 <(a,b)> 을 기반으로 한 C3을 형성하기 위한 S2는 b로 시작하는 시퀀스가 그 대상이 된다. 이산 항목 <b, a>와 연속 항목 <(b,c)>, <(b, e)>은 단 한번의 검색을 통해 이뤄지며, 이 두 노드를 복사함으로써 조인 연산이 완성된다.

가지치기(Pruning) 연산은 기존 방식과 마찬가지로, 후보 시퀀스들의 모든 부분 시퀀스를 구하고, 빈번 시퀀스인지에 대한 검사를 통해 이뤄지므로, 해쉬 트리와 똑같이 수행된다.

지지도 계산(Support Counting)에서는 비슷한 시퀀스들을 모아 놓은 효과를 갖는 MLLT을 이용하였을 경우, 해쉬 트리를 이용하는 것보다 검색 공간이 줄어 성능 향상을 가져온다.



2-빈번 시퀀스 조인으로 생긴 3-빈번 시퀀스

그림 3 MLLT를 이용한 조인 모습

```
Join( _cur_level, _start, _cons_end, _end,
      s2_start, s2_cons_end, s2_end,
      SequentialPatterns& next_sequential_patterns,
      unsigned long& _join_result_cons_size)
{
      if( cur_level == m_depth -1)   // leaf level
            for( e1 in S1) {
                  if( e2 in S2 such an e1 is founded)
                        each consecutive and discrete node of
                        e2 is copied and attached  on e1 in
                        _next_sequential_patterns
            }
      else // internal node
            for( e1 in S1 ) {
                  if( e2 in S2 such an e1 is founded )
                        recur join with children of e1 and
                        children of e2 such it and attach e1
                        on      m_depth-1      node      of
                        NextSequentialPattern
            }
}
```

그림 4 MLLT 조인 알고리즘

4. 결 론

고객 당 평균 트랜잭션 수 10, 트랜잭션 당 평균 항목 집합 수 2.5, 최대 빈번 시퀀스의 평균 길이 4, 최대 빈번 항목 집합의 평균 길이를 1.25로 고정시키고 고객수를 달리하여 GSP와 MLLT의 총 수행 시간을 비교하였다. 그 수행 결과는 그림 6 과 같다.

Dataset	C	T	S	I	D
C10-T2.5-S4-I1.25-D100K	10	2.5	4	1.25	100,000
C10-T2.5-S4-I1.25-D200K	10	2.5	4	1.25	200,000

그림 5 데이터 집합 크기

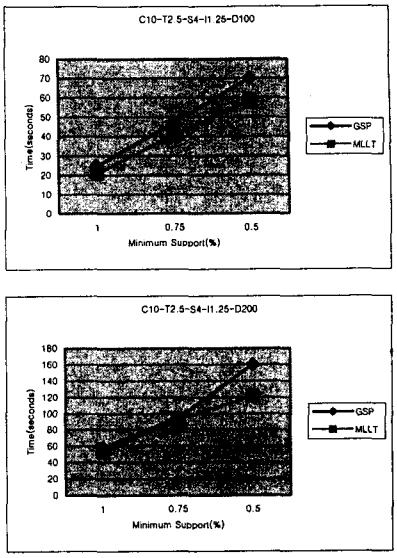


그림 6 순차 패턴 추출 수행 시간 비교

본 논문에서는 순차 패턴을 추출하는 과정에서 후보 시퀀스 또는 빈번 시퀀스를 저장하는 새로운 데이터 구조 MLLT를 제시하였다. MLLT는 시퀀스들이 공유하는 항목들을 모아 사전적인 순서에 맞춰 정렬하여 노드에 저장하고, 이러한 노드들을 파일 구조를 염두한 순차적인 배치를 통해 균형 트리를 이루도록 하였다. 데이터의 선형적인 저장은 데이터의 지역성을 이용한 것으로, 포인터 수와 메모리 요구량을 줄이고, 패턴 추출 과정에서 빈번이 일어나는 조인 연산을 단순 연산으로 대치함으로써,기본의 해쉬 트리를 통한 패턴 추출 방법보다 좋은 성능을 보였다. 저장 구조와 연산에 관한 고찰을 통한 순차 패턴의 성능 향상은 앞으로 순차 패턴 추출의 상용화에 도움이 되리라 생각된다.

참고 문헌

[1] Rakesh Agrawal and Ramakrishnan Srikant. "Mining Sequential Patterns". In Proceedings of the Eleventh International Conference on Data Engineering, 1995

[2] Ramakrishnan Srikant and Rakesh Agrawal. "Mining Sequential Patterns: Generalizations and Performance Improvements", In Proceedings of the Fifth International conference on Extending Database Technology(EDBT;96), March 1996

[3] 이동하, "다단계 선순열 트리와 개념 계층을 이용한 대용량 관계형 데이터베이스에서의 연관 규칙 추출 기법", 박사학위 논문, 포항공과대학교, 2000

[4] Mohanned J. Zaki . "Efficient enumeration of frequent sequences", In Proceeding of the Seventh International Conference on Information and Knowledge Management, 1998

[5] Jiawei Han, Jian Pei, Behzad Mortzavi-Asl, Qiming Chen Umeshwar Dayal, and Mei-Chun Hsu. "FreeSpan: Frequent Pattern-Projected Sequential pattern Mining", In Proceeding of the 2000 International Conference on Knowledge Discovery and Data Mining, 2000