능동 기반의 점진적 데이터 마이닝

연영광°, 신예호, 류근호
충북대학교 컴퓨터과학과
e-mail : {ykyeon, snowman, khryu}@dblab.chungbuk.ac.kr

An Incremental data mining based on Active system

Yeon Young Kwang°, Shin Yae Ho, Keun Ho Ryu
Dept. of Computer Science, Chungbuk National University

요약

데이터 마이닝 작업에서 사용되는 데이터의 크기는 그 특성상 대규모를 이루고 있다. 이러한 대규모의 데이터로부터 규칙을 추출하는 작업은 많은 비용이 소모된다. 또한 급변하는 데이터는 이미 발견된 마이닝 패턴에 대하여 현저한 패턴은 약한 패턴으로, 반면 약한 패턴은 현저한 패턴으로 변화시키는 요인이 되고 있다. 이러한 동적 환경에서는 기존의 데이터베이스 특정시간의 스냅 샷 형태의 데이터를 이용하였던 마이닝 방법으로는 적당하지 못하다. 따라서 이 논문에서는 동적인 환경에서 적용할 수 있는 점진적 마이닝 방법을 제시하고, 점진적 마이닝 작업이 효과적으로 수행 가능한 능동시스템 모델을 제시한다.

1. 서론

데이터 마이닝은 이론적 시도에서 벗어나 현재 그 활용 범위가 데이터베이스 마케팅, 전자상거래, 주식시장 분석 등과 같이 널리 이용되고 있다. 이러한 마이닝 적용 분야에서 사용되는 데이터는 실제로 연속적인 형태로 빠르게 수집되어 대규모의 데이터를 이루게 된다. 이때 마이닝 관점에서 이미 발견된 규칙은 현시점에서 유효하지 않을 수 있다. 즉 현저한 패턴은 약한 패턴으로, 약한 패턴은 현저한 패턴으로 수시로 바뀔 수 있다. 따라서 이러한 동적인 환경에서 수집되는 마이닝 데이터는 기존에 수행되었던 접근 방식인 특정 시간의 데이터베이스의 스냅 샷 데이터에 대한 마이닝 방식으로는 적당하지 못하다. 이와 같은 문제점을 해결하기 위한 연구[1, 2, 3]가 최근 수년간 진행되어왔다.

이 논문에서는 동적 환경에 적용할 수 있는 마이닝 방법으로서 새로 갱신되는 데이터에 대하여 점진적으로 패턴을 유지하여 마이닝 시점에서 빠르게 응답할 수 있는 마이닝 시스템을 제안한다. 이 시스템에서는 데이터베이스에 자동 조치기능을 갖는 능동 서브시스템을 이용하여 새로 입력되는 데이터가 트랜잭션의 의미에 따라 트리거를 적용하여 빈발 항목을 유지한다. 즉 능동 데이터베이스에서 자체적으로 빈발항목을 유지함으로서 마이닝 응용프로그램에서는 단순히 데이터베이스에 있는 빈발항목을 이용하여 규칙을 생성하여 사용자에게 제공한다.

이 제안된 시스템은 동적 환경에 대응할 수 있도록 데이터베이스 스스로 마이닝 작업에 참여함으로서 자주 변경되는 마이닝 패턴을 실시간으로 탐사가 가능하다.

이 논문은 다음과 같이 구성된다. 2장에서는 점진적 마이닝작업에 대한 문제 정의를 하고, 3장에서는 동적 환경에 수행되는 능동 마이닝 알고리즘을 기술한다. 4장에서는 점진적 구조를 갖는 마이닝 시스템에 대하여 설명한다. 5장에서는 점진적 마이닝 작업에서의 트레이드 오프(tradeoff)에 대해 논하고, 마지막 6장에서 결론과 현재 진행중인 작업에 대해 설명한다.

2 문제 정의

2.1 빈발 패턴

$I = \{a_1, a_2, ..., a_n\}$를 항목 집합이라 하고, 트랜잭션 데이터 베이스 $DB = \langle T_1, T_2, ..., T_n \rangle$일 때 $T_i (i \in$

이 연구는 한국과학재단 특정 기초(1999-2-303-006-3)연구비 지원으로 수행되었음

[1...n])은 Ti가 같은 항목집합이라 하자. 어떠한 항목 A의 지지도(빈발횟수 : s)는 DB에서 A를 포함하는 트랜잭션 수이다. 빈발 항목(L)은 사용자가 정의한 최소 지지도(ξ)를 만족하는(즉, 최소지지도보다 큰) 항목을 의미하며, 빈발 패턴은 연관성 있는 빈발 항목의 집합을 의미한다.

2.2 동적 환경 하에서의 점진적 마이닝

원 데이터베이스(DB)에 새로 입력되는 트랜잭션 데이터베이스를 db이고. D는 DB의 트랜잭션 수이며, 각 A∈L에 대하여 A에 대한 지지도를 X.SUPP라 가정하자.

새로운 트랜잭션 데이터베이스 db가 원 데이터베이스 DB에 더해질 때, 같은 최소 지지도 s에 대하여 X.SUPP >= s*(D+d)인 경우 항목 X는 새로 갱신된 데이터베이스 DB∪db에 대하여 빈발한다.

즉 점진적 마이닝의 문제는 새로 입력되는 db에 대한 새로 갱신된 데이터베이스(DB∪db)대하여 빈발 패턴을 찾는 작업이다.

3. 점진적 마이닝 알고리즘

이 장에서 설명되는 점진적 알고리즘은 Agrawal에 의해 제시되었던 Apriori 알고리즘[4]을 기반으로 한다. 중요한 차이점은 능동 데이터베이스 자체의 능동 규칙을 이용하여 빈발 항목이 유지된다. 빈발 패턴 유지는 준비단계, 첫 번째 연산, 그리고 k-번째 연산 단계로 구성된다.

준비단계에서는 사용자가 초기 신뢰도 값을 정하거나, 재 지정한 신뢰도 값이 기존 값보다 작을 경우 즉, SUPP > new supp 인 경우 트리거가 기동하게 되며, 데이터베이스를 전체 스캔하여 이미 존재하는 데이터베이스의 패턴을 찾는다.

첫 번째 연산 단계는 새로 입력되는 트랜잭션 데이터에 대하여 점진적으로 마이닝을 수행한다. 이 단계에 대한 처리순서가 그림 1.에 있다. k-번째 연산 단계로 넘어가기 위한 조건은 L_1에 갱신 연산이 수행되어 패턴변화가 생겼을 경우이다.

k-번째 연산 단계는 첫 번째 연산 단계 이후 새로 입력되는 항목이 지지도 값을 만족시키는 경우 그 다음 연산으로 진행된다. 이 단계에의 기본 흐름은 첫 번째 연산 단계와 비슷하며 C_{k+1}에 대한 빈발 항목 생성부분이 추가된다. 즉 빈발항목 L에 대한 갱신 사건이 발생 할 경우 어떠한 조건 검사 없이 후보 항목에 대한 재 조합이 수행된다.

다음 그림 2.은 빈발 항목 L_k에 사건이 발생할 경우 C_{k+1}의 후보항목 조합에 대한 트리거 정의이다.

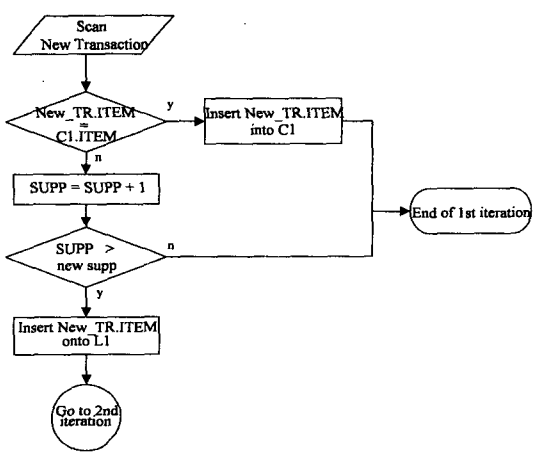

그림 1. 1st iteration의 처리

```
Event : Insert or Update on Lk
Condition : none
Action :
  Operation :
    begin
      insert into Ck+1
      select p.item1, p.item2, ..., p.itemk, q.itemk
      from Lk p, Lk q
      where p.item1 = q.item1, ..., p.itemk-1 =
            q.itemk-1,  p.itemk < q.itemk;
    end;
```

그림 2. C_{k+1}후보항목에 조합에 대한 Trigger Pseudo definition

4. 능동 시스템 구조

능동 시스템은 설계형태에 따라 데이터베이스 시스템과 통합된 형태, 레이어 형태, 그리고 컴파일형 구조 구분된다[4]. 이 시스템에서는 트랜잭션 관리자 수준에서 통합된 형태로 트랜잭션 관련 규칙처리가 정확하게 반영될 수 있다는 장점을 취한다.

그림 3.은 능동 시스템 구조를 나타내는 개념도 이다. 사용자는 능동 시스템자체에서 마이닝 작업이 가능하도록 사용자 인터페이스를 통해 지지도를 명시한다. 데이터베이스는 사용자가 정의한 지지도 보다 작거나 같게(SUPP <= new supp) 빈발 항목을 유지한다.

능동 데이터베이스에서는 빈발 패턴을 모니터 하여 패턴 변경이 일어날 경우 응용프로그램의 마이닝 규칙 생성기를 호출한다.

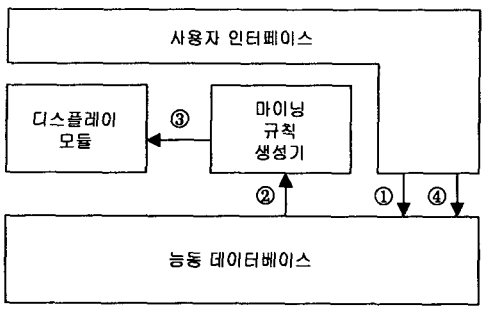

그림 3. 능동 시스템 구조

마이닝 규칙 생성기는 변경된 마이닝 패턴에 따라 규칙을 만들어 디스플레이 모듈에 의해 사용자에게 이해하기 적합한 형태로 규칙을 보여 준다.

이 능동 시스템은 시스템 제어 정보뿐만 아니라 데이터가 각 모듈로 전달된다.

그림 3.의 ①~④는 각 모듈에서 전송되는 정보이며 각 의미는 다음과 같다.

① 데이터베이스 연산 : 데이터 정의, 데이터 조작 및 트랜잭션 연산

② 응용프로그램 호출 : 능동 시스템에서 빈발 패턴을 관리하다가 변화가 생겼을 경우, 마이닝 규칙 생성기를 호출하여 새롭게 갱신된 규칙생성

③ 디스플레이 모듈 호출 : 마이닝 규칙 생성기에서 새로운 규칙 생성 후 디스플레이 모듈에서 새로운 정보를 보여줌

④ 트리거 연산 : 데이터베이스 트리거 연산(creation, disable, enable...)을 수행한다.

5. 점진적 마이닝 작업의 트레이드 오프(tradeoff)

점진적 마이닝 구조를 취하는 동적인 환경에서 마이닝 기본 환경으로 능동 데이터베이스를 이용하고 있다[1, 2, 3]. 왜냐하면 능동데이터베이스의 자동 조치기능은 실시간으로 동적 환경에 대하여 즉각적인 조치를 취할 수 있기 때문이다. 이에 대한 장점은 빈발 패턴을 데이터베이스 스스로 유지 관리하기 때문에 응용프로그램에서는 패턴을 이용하여 사용자에게 이해하기 쉬운 형태로 보여 주는 역할을 한다.

능동 데이터베이스의 자동 조치기능은 동적인 마이닝 환경에 적합하지만 효율에 대한 다음과 같은 3가지의 트레이드 오프가 존재한다.

1. 능동 규칙이 실시간으로 마이닝작업에 참여 할 경우 각 트랜잭션의 의미에 따라 규칙이 수행 되지만 데이터베이스 자체의 오버헤드가 초래된다.

2.사용자가 최소 지지도를 변경할 경우(SUPP > new supp) 원 데이터베이스 전체 스캔이 필요하다. (일반적으로 사용자의 98%의 마이닝 질의는 최소 지지도 ξ >= 20 범위에 든다[6].)

3.빈발 패턴이 데이터베이스 자체에 저장되기 때문에 공간적인 낭비가 초래한다.

따라서 동적 환경에 수행되는 마이닝 시스템은 기능적인 면에서 효과적이지만, 성능향상을 위해서는 위에 언급한 사항을 고려해야 한다.

6. 결론

데이터 마이닝의 범위가 확대되고 작업에 대한 요구사항이 다양해지고 있으며, 동적으로 수집되는 방대한 데이터를 처리하기 위해 점진적 마이닝 방법이 연구되고 있다. 이런 점진적 마이닝 작업을 효과적으로 수행하기 위해 능동 시스템이 사용된다.

이 논문은 점진적 마이닝에 적합한 능동 시스템을 제안하고 설계하였다. 이 시스템에서는 마이닝의 빈발 패턴을 동적으로 유지하도록 하였다. 또한 설계한 능동 시스템에서 수행 가능한 마이닝 알고리즘을 제시하였다.

현재 진행중인 연구로 능동 규칙과 효율적 결합 가능한 마이닝 작업, 그리고 마이닝 질의어 및 인터페이스 설계가 수행 중이다.

참고문헌

[1] H. Kawano, S. Nishio, J. Han, and T. Hasegawa, "How Does Knowledge Discovery Cooperate with Active Database Techniques in Controlling Dynamic Environment?", in Proc. 5th Int'l Conf. on Database and Expert Systems Applications (DEXA'94), Athens, Greece, September 1994, pp. 370-379.

[2] J. Han, S. Nishio and H. Kawano, "Knowledge Discovery in Object-Oriented and Active Databases", F. Fuchi and T. Yokoi (eds.), Knowledge Building and Knowledge Sharing, Ohmsha, Ltd. and IOS Press, 1994, pp. 221-230.

[3] R. Agrawal, G. Psaila "Active Data Mining", Proc. of the 1st Int'l Conference on Knowledge Discovery and Data Mining, Montreal, August 1995.

[4] R. Agrawal, K Srikant "Fast Algorithms for Mining Association Rules. In Proc. of the 20th Int'l conference on VLDB, 1994.

[5] J. Widom, S. Ceri, "Introduction to Active Database Systems", Chapter 1, Active Database Systems, Triggers and Rules for Advanced Database processing, Morgan Kaufman Pub, 1996.

[6] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation", Proc. 2000 ACM-SIGMOD Int. Conf. on Management of Data (SIGMOD'00), Dallas, TX, May 2000.