스트림 데이터의 효율적인 연관규칙 업데이트 알고리즘

김형주
서울대학교
dawnglow@hanmail.net

최재걸°
네이버개발실
jkchoi@nhncorp.com

HARD : Hybrid Association Rule with streaming Data

Hyung-ju Kim
Scoul National University

Jae-keol Choi °
Naver Development Department

요 약

스트림 데이터에 내재된 정보들은 시간이 흐름에 따라 변화의 가능성이 매우 높기 때문에 이러한 변화를 신속하게 업데이트할 수 있다면 유용한 정보를 제공할 수 있을 것이다. 그러나 스트림 데이터의 모든 연관규칙을 업데이트하는 것은 수행하는데 많은 부담이 있으므로 이 논문에서는 지지도의 변화가 큰 흥미있는(interesting) 데이터에 대해서 효율적으로 업데이트 하는 방법을 제시하고자 한다.

제 1 절 서론

스트림 데이터는 정적인 특성을 가지는 전통적인 데이터베이스의 데이터와는 달리 매우 빠른 시간 내에 지속적으로 데이터가 증가되는 동적인 특성을 가진다. 또한 스트림 데이터는 시간에 흐름에 따라 다양한 데이터의 분포를 가진다. 그런데 이전의 알고리즘들은 스트림 데이터의 전부를 이용하여 새로운 연관규칙을 찾아내는 방법을 취하고 있다. 그러나 스트림 데이터를 이용하여 연관규칙을 찾는 과정에서 메모리 사용량은 무한히 증가되지 않고 한정되기 때문에 이러한 방법은 그리 효율적이지 못하다. 또한 이러한 방법은 데이터의 분포 변화가 큰 경우에도 성능이 떨어질 수 있어 사용하기에 적절하지 않을 수 있다. 그래서 우리는 빠른 속도와 함께 데이터의 분포 변화가 큰 스트림 데이터 전부를 꼭 사용해야 하는가하는 질문을 던지게 되었고, 중요한 데이터만을 가지고 연관규칙을 구하는 알고리즘을 연구하게 되었다. 이 논문에서는 스트림 데이터에서 지지도의 변화가 큰 흥미있는 유효한 데이터를 먼저 골라내고 그 데이터들 사이에서 연관규칙을 찾아내는 방법을 연구하였다. 실험 결과는 이 연구가 유효함을 입증하여 주고 있다.

제 2 절 기존 알고리즘

2.1 기존 알고리즘

기존의 스트림 데이터의 연관규칙 업데이트 알고리즘 가운데 [8] 알고리즘은 전체 스트림 데이터를 통해서 새로운 연관규칙(넓은 의미에서 순차패턴이라고 할 수 있다.)을 구하고 이미 구해진 연관규칙과의 거리를 측정하여 업데이트 할 것인지 아닌지를 결정한다.

2.2 알고리즘의 문제점

이 알고리즘은 스트림으로 들어온 모든 데이터를 다루므로 수행과정에서 저장공간과 수행속도에서 많은 어려움이 발생한다. 특히 이미 구해진 연관규칙과 새로운 연관규칙 사이에 많은 차이가 있다면 성능(performance)이 많이 떨어질 수 있다는 단점이 있다. 즉, 이 알고리즘은 데이터 분포(data distribution)의 변화가 작은 경우에만 사용하기에 적절하다고 볼 수 있다. 그러나 실제 응용 분야 중 네트워크 분야 등에서는 스트림 데이터의 데이터 분포의 변화가 작은 경우 뿐만 아니라 큰 경우도 많이 있다. 그래서 우리는 데이터 분포의 변화가 큰 스트림 데이터의 경우에 효율적으로 적용할 수 있는 스트림 데이터 연관규칙 업데이트 알고리즘을 제안하고자 한다.

제 3 절 하이브리드 알고리즘

기존의 고전적인 연관규칙 알고리즘을 이용하여 오프라인(offline)에서 일정시간 간격으로 기준이 되는 빈발패턴을 추출한다. 스트림 데이터를 이용하여 실시간으로 구해진 빈발패턴은 앞서 구해진 결과물을 업데이트 하는 용도로 사용한다. 이렇게 오프라인 데이터로 찾은 빈발패턴은 스트림 데이터를 이용해 빈발패턴을 찾는데 있어서 많은 부담을 줄여준다. 스트림 데이터를 분석하여 완전한 연관규칙을 구하지 못한다 하더라도 일정간격을 두고 다시 완전한 알고리즘을 수행하게 되므로 보충이 될 것으로 기대할 수 있기 때문이다. 스트림 데이터의 이용은 매우 중요한 빈발패턴들 간의 연관규칙을 신속하게 구해서 업데이트 하는데 그 목적을 두도록 한다. 이렇게 함으로써 스트림 데이터를 처리하는 부담을 줄일 수 있고 빠른 속도의 스트림 데이터에서 중요한 데이터에 대한 연관규칙을 업데이트하고 또한 연관된 빈발패턴을 찾아내는 일이 가능할 수 있도록 한다.

3.1 오프라인 데이터 연관규칙

오프라인 데이터로 빈발패턴을 추출할 때는 기존에 잘 알려진 여러 연관규칙 알고리즘들을 이용할 수 있다. 전통적인 연관규칙 알고리즘인 Apriori 알고리즘 [1] 등을 사용하면 된다.

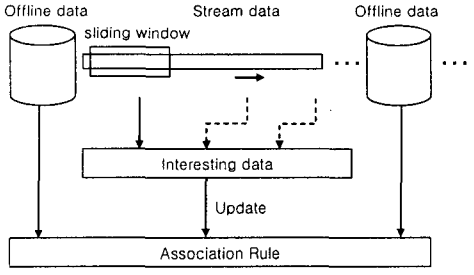


그림 1: 전체 프로세스

3.2 스트림 데이터 연관규칙

스트림 데이터를 이용하여 검색어를 추출하는 단계는 두 단계로 이루어져 있다. 먼저, 스트림 데이터에서 흥미있는 데이터를 찾는 단계, 그리고 흥미있는 데이터들 간의 연관법칙을 찾는 단계이다.

3.2.1 흥미있는(interesting) 데이터 추출

스트림 데이터는 그림 1 과 같이 슬라이딩 윈도우를 이용하여 처리된다. 사용자가 정한 단위 시간 동안 스트림 데이터를 움직이는 슬라이딩 윈도우를 통해 각각의 항목에 대한 지지도를 계산한다. 그리고 스트림 데이터 가운데 오프라인에서 계산된 지지도와 비교해서 지지도가 일정한 기준 값보다 크게 상승된 데이터가 있을 때 우리는 이 데이터를 흥미있는 데이터라고 생각한다. 오프라인에 비하여 지지도가 크게 상승한 것을 판단하기 위해서서는 오프라인에서 구해진 기준 데이터가 필요하다. 이를 위하여, 오프라인에서 각 항목의 지지도의 평균값과 표준편차를 구한다. 지지도의 변화가 정규분포를 따른다고 가정할때, 평균값과 표준편차를 이용하여 슬라이딩 윈도우내에서 각 항목들의 지지도가 나타낼 수 있는 값을 예측 할 수 있다. 만약 이 예측값을 크게 상회한 다면, 특이하게 지지도가 상승하고 있다고 판단 할 수 있을 것이다. 항목 q에 대하여 오프라인에서 일정 시간 간격으로 측정한 지지도를 $p_t(q)$ 라고 할 때, n 개의 시간간격에 대한 평균($E(q)$)과 표준편차($\sigma(q)$) 는 다음과 같이 구할 수 있다.

$$E(q) = \frac{1}{n}\sum_{t=1}^{n} p_t(q) \tag{1}$$

$$\sigma(q) = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(E(q) - p_t(q))^2} \tag{2}$$

슬라이딩 윈도우에서 측정한 항목 q의 지지도 $p_w(q)$ 의 값이 수식 1에서 측정된 평균값과 흡사하다면 이 항목은 지지도가 크게 변하지 않았다고 판단할 수 있을 것이다. 만약 지지도 값이 평균값을 크게 상회하고 있다면 지지도가 상승한 경우이므로, 우리가 찾고자 하는 흥미있는 항목이라고 판단한다. 이를 판단하는 기준은 수식 1 과 수식 2에서 구해진

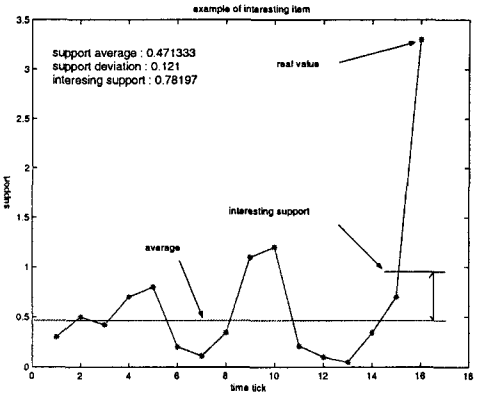


그림 2: 흥미있는 항목 예제

평균과 표준편차의 조합으로 다음과 같이 표현할 수 있다.

$$p(q) > E(q) + \alpha \cdot \sigma(q) \tag{3}$$

α 값은 조정이 가능하며, 실험적으로 2.57 (99% 표준정규분포 위치) 이 적당함을 알 수 있다. 그림 2 은 이 장에서 제시한 방법으로 선택된 항목의 일례이다. 수식 3에서 제시한 값을 크게 상회하여 흥미있는 항목으로 선택되었다. 단, 예외적으로 오프라인에서 존재하지 않았던 항목이 새롭게 나타나는 경우에는 미리 측정된 항목의 지지도 값이 없으므로 최소지지도 보다 높으면 흥미있는 항목이라고 판단한다.

3.2.2 흥미있는 데이터에 대한 연관규칙 업데이트

3.2.1 에서 흥미있는 항목이 선택되면 이 항목에 대해서만 연관규칙을 추출하도록 한다. 연관규칙을 추출하는 방법은 [1] 등 이미 잘 알려진 방법들을 사용할 수 있다. 이미 흥미있는 데이터만을 골라내어 탐사공간을 줄인 상태이므로 시간복잡도와 공간복잡도에 대해 강점을 갖는다. 흥미있는 데이터에서 연관규칙이 추출되면 오프라인에서 구해진 연관규칙과 비교하여 업데이트를 한다. [8] 에서 두 연관규칙 사이의 거리를 정의하고 거리가 멀어질 경우 업데이트 하는 방법 등을 제안하고 있으나, 이 경우에는 현재의 슬라이딩 윈도우 안에서 만들어진 결과가 가장 적합하다고 판단하여 차이가 발생한 모든 규칙을 적용 업데이트 하도록 한다.

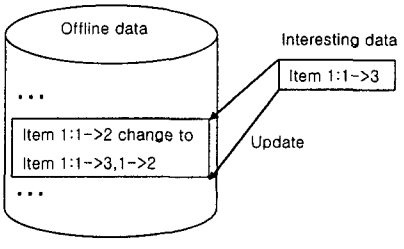
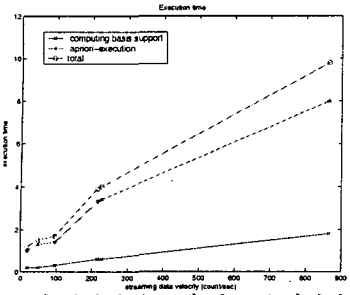
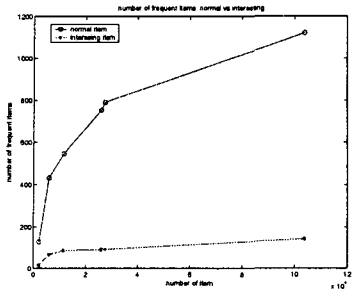


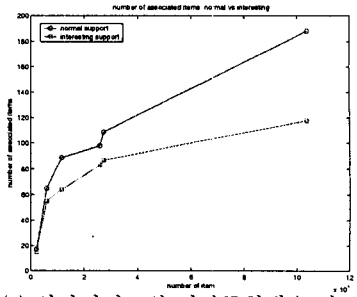
그림 3: 흥미있는 항목의 연관규칙 업데이트



(a) 스트림 데이터 속도에 따른 수행시간(Sec)



(b)일반지지도와 빈발항목개수 비교



(c) 일반지지도와 연관규칙개수 비교

그림 4: 실제 데이터 집합들에 대한 성능 평가

제 4 절 실험 결과

모든 실험은 LINUX 운영체제에 메모리 1GigaByte, CPU Intel(R) Xeon(TM) CPU 3.00GHz인 Server장비에서 실행되었다. 사용된 데이타는 검색 서비스를 통해 얻어진 엑세스 로그의 스트림 데이터를 이용하였다. α 값은 모두 2.57을 사용하였다. 그림(a)는 스트림 데이터의 속도에 따른 수행시간을 나타낸 그래프이다. 지지도가 증가하는 흥미있는(interesting) 항목을 찾아내는 시간과 찾아진 아이템의 연관규칙을 구하는 시간을 나타내었다. 그림에서 볼 수 있듯이 각 프로세스는 모두 선형으로 나타나며 수행시간은 수 초 단위이므로 스트림데이터 처리에 적합하다고 할 수 있다. 그림 (b)는 흥미있는(interesting) 항목의 개수와 지지도를 만족하는 모든 항목의 개수를 비교해 보여준다. 기존의 알고리즘을 사용하면 지지도를 만족하는 모든 항목이 대상이 되어 그 수가 많으나, 이 논문에서 제시한 방법을 사용하면 그림에서 보는 바와 같이 그 대상을 현격히 줄일 수 있다. 스트림 속도에 따라서 대상의 개수에 차이가 생기나 평균적으로 약 80% 이상 줄어드는 것을 확인할 수 있다. 그림 (c)는 흥미있는(interesting) 항목만을 가지고 연관규칙을 구한 경우와 모든 항목으로 연관규칙을 구한 경우 새롭게 찾아지는, 즉 변경되어야 하는 연관규칙의 개수를 나타내고 있다. 그림에서 볼 수 있듯이 이 논문에서 제시한 방법으로 연관규칙을 찾을 경우, 기존의 방식에서 찾아내는 모든 연관규칙을 찾아낼 수는 없다. 그러나, 그 차이가 크지 않았고, 또한 중요한 항목의 연관규칙은 모두 찾아내었으므로 효율성의 측면에서 우수하다고 할 수 있다. 이 알고리즘이 적용된 검색환경의 경우 이러한 상황은 더욱 명백하다.

제 5 절 결론

본 논문에서는 스트림 데이터에서 연관규칙을 찾을때 탐사공간을 줄이는 방법으로 지지도가 상승한 항목(item)만을 선별적으로 찾아내는 알고리즘을 연구하였다. 지지도가 상승한 항목만을 사용하여 연관규칙을 구함으로써 스트림 데이터 처리의 시간복잡도와 공간복잡도가 개선되었다. 실험결과는 이러한 항목만을 사용하여도 결과의 질이 크게 하향되지 않음을 보여주고 있다. 앞으로 선별적으로 항목을 골라내는 방법을 더욱 연구할 가치가 있을 것이다.

참고 문헌

[1] R. Agrawal and R. Srikant. "Fast algorithms for mining association rules," In *Proc. 1994 Int'l Conf. Very Large Data Bases (VLDB'94)*, pages 487-499, Santiago, Chile, September 1994.

[2] R. Agrawal and R. Srikant. "Mining sequential patterns," In *Proc. 1995 Int'l Conf. Data Engineering (ICDE'95)*, pages 3-14, Taipei, Taiwan, March 1995.

[3] R. Agrawal and R. Srikant. "Mining Generalized Association Rules," In *Proc. 21st Int'l Conf. Very Large Data Bases (VLDB'95)*, pages 3-16, Zurich, Swizerland, September 1995.

[4] M. Garofalakis, R. Rastogi, and K. Shim. "Spirit: Sequential pattern mining with regular expression constraints," In *Proc. 1999 Int'l Conf. Very Large Data Bases (VLDB'99)*, pages 223-234, Edinburgh, UK, September 1999.

[5] J. Han, J. Pei, and Y. Yin. "Mining frequent patterns without candidate generation," In *Proc. 2000 ACM-SIGMOD Int'l Conf. Management of Data (SIGMOD'00)*, pages 1-12, Dallas, TX, May 2000.

[6] J.S. Park, M.S. Chen, and P.S. Yu. "An effective hash-based algorithm for mining association rules," In *Proc. 1995 ACM-SIGMOD Int'l Conf. Management of Data (SIGMOD'95)*, pages 175-186, San Jose, CA, May 1995.

[7] S.D.Lee and D.W.Cheung. "A note on Maintenance of Discovered Association Rules: When to update?," In *Proc. 1997 ACM-SIGMOD Workshop on Data Mining and Knowledge Discovery(DMKD'97) in cooperation with ACM-SIGMOD'97*, Tucson, Arizona, May 11, 1997.

[8] Q.Zheng, K.Xu, W.Lv, S.Ma. "A note on The Algorithms of Updating Sequential Patterns," In *Proc. The Second SIAM (Society for Industrial and Applied Mathematics) Data mining,2002:workshop HPDM (High Performance Data Mining, Washington, USA, April 2002 available at http://arXiv.org/abs/cs.DB/0203).*