최신구간을 활용한 점진적 로그 분석 기법

김명순[0] 박병준
광운대학교 컴퓨터과학과
{mskim[0], bjpark}@cs.kwangwoon.ac.kr

Practical Use of Recent Section on Incremental Log Analysis technique

Myung-soon Kim[0] Byung-Joon Park
Dept. of Computer Science, Kwangwoon University

요 약

본 논문에서 최신구간을 활용하여 패턴의 최신성을 보장하고 최신구간내 패턴의 누락 없이 모든 패턴을 발견할 수 있는 점진적 로그 분석 기법을 제안한다. 즉, 주기마다 최신구간이 이동되면서, 동시에 패턴의 최신성 여부가 결정되고, 이동된 최신구간내 패턴이 될 후보 패턴을 미리 예측하여 보다 합리적인 패턴 관리할 수 있다. 따라서 일반적인 점진적 로그 분석 기법에서 간과된 대량의 로그에 숨겨진 패턴은 적어도 해당 최신구간내에서 모두 발견될 수 있고 최신성도 보증된다.

1. 서론

웹 마이닝의 한 분야인 웹 이용 마이닝은 웹 서버 로그로부터 이용자의 패턴이나 의미 있는 행동 발견을 목적으로 하고 있다. 오프라인 환경의 데이터 마이닝과 달리, 웹 마이닝은 온라인 환경에서 발생하는 실시간 데이터를 이용하여 주기적으로 패턴을 발견하고 갱신하는 것이 특징이다. 이러한 분석 기술은 폭발적으로 증가하는 웹(WWW)과 이용자 환경에서 더 중요하다. 즉 제한된 시간과 자원으로 대량의 웹 엑세스 로그로부터 연간 또는 격년간의 이용자 엑세스 패턴을 추출하기는 어렵다.[2] 그래서 현재 분석 주기마다 그 결과를 결합, 재평가, 갱신하는 단계가 필요한 주기적, 점진적 로그 분석 기법이 연구되고 있다. 이러한 연구의 결과는 단기적 패턴 발견에 초점을 두고 있어 전체 데이터 분석에 의한 패턴 발견 또는 장기적인 패턴을 기대하기는 어렵다. 즉, 빈발 행동 패턴이 로그 파일에 저장된 대량의 데이터 사이에 숨겨진 경우, 이 패턴이 특정 시간 범위에서만 빈번하게 발생하지만 전체 로그 파일에서는 그렇지 않은 경우이다.[3] 예를 들어 적어도 도메인성격상 일년 동안의 데이터에서 발견되는 패턴은 최신성이 있다라고 가정하자. 그러나 매달 실행되는 로그 분석 과정에서 제거되었거나 나타나지 않는 패턴은 일년 동안의 로그를 일괄적으로 분석했을 경우 패턴으로 발견될 수 있다.

일반적인 점진적 로그 분석 기법은 이미 제거된 패턴을 재발견 또는 하는 것이나 숨겨진 패턴을 발견하는 것을 간과해 오고 있다. 웹 환경 기반으로 모든 패턴의 발견은 축적된 전체 로그를 분석 주기마다 재분석하거나, 모든 후보 패턴을 저장, 유지해야 가능한데 현실적으로 어렵고 그 최신성조차 불확실하기 때문이다. 그래서 본 논문에서 최신구간을 활용하여 패턴의 최신성을 보장하고 최신구간내 패턴의 누락 없이 모든 패턴을 발견할 수 있게 한다. 즉, 주기마다 최신구간이 이동되면서, 동시에 패턴의 최신성 여부가 결정되고, 이동된 최신구간내 패턴이 될 후보 패턴을 미리 예측하여 합리적인 패턴 관리에 기여한다.

본 논문의 구성은 2장에서는 관련연구, 3장에서는 최신구간 설명과 후보 패턴 추출을 위한 임계값 도출, 점진적 로그 분석으로 인한 로그 분할에서 오는 문제 해결, 4장에서는 최신구간을 활용한 점진적 로그 분석 시스템에 대해 기술, 5장에서는 결론과 향후 연구에 대해 기술한다.

2. 관련연구 [1]

패턴 마이닝의 일반적인 기술로서 이용자 엑세스 패턴을 발견하기 위한 Heikki Mannila의 이벤트열에서 빈발 에피소드(frequent episode)를 추출하는 기법을 사용하였다. 이를 웹 로그 분석에 적용하면 이벤트열은 로그의 집합, 에피소드는 로그의 조합, 즉 이벤트간의 조합이다. 따라서 이벤트열의 스캔 결과에 따라 빈발 에피소드의 여부가 결정되고 빈발 에피소드는 패턴으로 이용된다.

다음 식에 의해 각 에피소드의 빈도를 계산한다.

$$\text{에피소드의 빈도 (fr)} = \frac{\text{에피소드가 나타난 윈도우의 수 (occur)}}{\text{전체 윈도우의 수 (windows)}}$$

에피소드가 나타날 일정 시간 간격이 윈도우이며, 즉 윈도우 내에 나타난 이벤트 조합만이 에피소드이다. 그리고 이벤트열의 스캔은 윈도우 단위로 진행되며, 그 에피소드는 자신이 나타난 윈도우의 수로 빈도를 얻는다. 결국 분석시 주어진 임계값 최소 빈도(*min_fr*)를 넘는 에피소드만이 '빈발 에피소드'가 된다.

전체 윈도우의 수(windows)는

$$windows = (T_e - T_s) + win - 1$$

로 윈도우의 폭(*win*)을 이용하여 이벤트열의 시작시점(*Ts*)을 포함하는 윈도우로부터 종료시점(*Te*)을 포함하는 윈도우까지 계산된다. 또한 임계값 최소 빈도(*min_fr*)를 이용하여

$$min\_fr = \frac{min\_occur}{(T_e - T_s) + win - 1}$$

빈발 에피소드의 윈도우들 내 최소 발생수(*min_occur*)를 얻을 수 있다.

3. 최신구간과 후보 패턴을 추출을 위한 임계값

3.1 최신 구간(recent section)과 이동

일단 최신구간으로 정해지면 그 구간내에서 발견된 패턴은 최신 패턴(up-to-date pattern)이며, 축적된 전체 데이터에 숨겨진 패턴이라 할지라도 최신 구간외의 패턴은 급변하는 웹 환경에서 이용가치가 없는 구식 패턴(out-of-date pattern)이라 판단하는 기준 구간이 된다. 그래서 점진적 로그 분석 시 최신구간내의 패턴은 최신의 패턴으로서 효용성 있고, 적어도 최신 구간 내에서는 패턴 누락없이 모든 패턴을 찾을 수 있다. 이것은 웹 이용자의 수와 다양한 엑세스 패턴으로 인한 데이터의 대량성은 전체의 숨겨진 패턴을 발견하기에 현실적으로 어렵게 만들고, 데이터의 다변성은 숨겨진 패턴의 이용가치를 감소시키므로 최신 구간의 제안에 그 의의가 있다.

최신구간의 범위는 도메인과 적용분야에 따라 달라지며 분석이 거듭될수록 최신 구간은 이동한다.

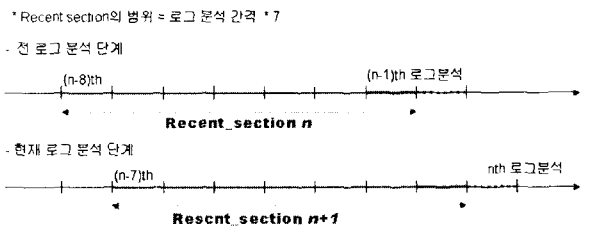

그림 1 최신 구간(recent section)의 이동

예를 들어 그림1과 같이 만약 최신 구간의 범위가 로그 분석 간격의 일곱 배로 주어졌다. 위와 같이 최신 구간의 이동은 로그 분석 시마다 수행된다. 다시 말해 *n*번째 새로운 이벤트열이 추가되는 동시에 최신 구간내의 *n-8*번째의 가장 오래된 이벤트열이 삭제된다. 이 과정에서 가장 오래된 이벤트열 내에서 나타난 에피소드의 빈도는 해당 이벤트열에서 계산된 빈도만큼 다시 감소되어 패턴 여부가 설정된다. 이를 위해 에피소드의 빈도는 로그 분석 단계별로 계산되어 저장된다.

3.2 후보 패턴 추출을 위한 임계값

위의 그림 1에서 최신구간의 다음 이벤트열을 빨간 점선으로 나타내었다. 즉 다음 단계 로그 분석 구간이라 할 수 있는데 현재 발견된 패턴은 아니지만 다음 로그 분석 단계에서 패턴이 될 후보 패턴 예측하여 별도로 보관하여야 한다. 이것은 최신구간내의 패턴 누락을 방지할 수 있다. 즉 반복적 데이터 분석을 피하고도 최신구간 내에서 숨겨진 패턴을 발견, 또는 올바른 패턴 제거작업을 수행할 수 있는 장점이 있다. 우선 최신구간내의 패턴을 찾기 위한 각 에피소드의 빈도는 위의 관련 연구와 유사한 방법으로

$$\text{에피소드의 빈도(fr)} = \frac{\text{에피소드가 나타난 윈도우의 수(occur)}}{recent\_section + (win - 1)}$$

로 계산된다. 또한 주어진 최소 빈도(*min_fr*)를 이용하여

$$min\_fr = \frac{min\_occur}{recent\_section + (win - 1)}$$

으로 빈발 에피소드가 윈도우들 내의 최소 발생수(*min_occur*)를 얻을 수 있다. 그래서 후보 패턴을 추출하기 위한 임계값을 *next_min_fr*이라 하고 그 계산은 다음 순서와 같다.

a. $min\_occur = min\_fr * (recent\_section + (win - 1))$

b. $next\_min\_occur = ans\_interval * min\_fr$

c.

$$next\_min\_fr = \frac{min\_occur - next\_min - occur}{recent\_section + (win - 1)}$$

최신구간내 에피소드의 최소 발생수(*min_ocurr*)에서 다음 단계로 그 분석의 최소 발생수(*next_min_occur*)만큼을 감산하면 최소 임계값(*min_fr*)은 낮아진다. 이 임계값 *next_min_fr*으로 다음 로그 분석에서 빈발 에피소드의 가능성을 미리 예측할 수 있다.

따라서 그림 2와 같이 최소 임계값(*min_fr*)에 의해 빈발 에피소드가 추출되고 잉여 에피소드 중, 다음 단계 로그 분석 시에 빈발 에피소드의 가능성이 있는 후보 에피소드가 추출된다.

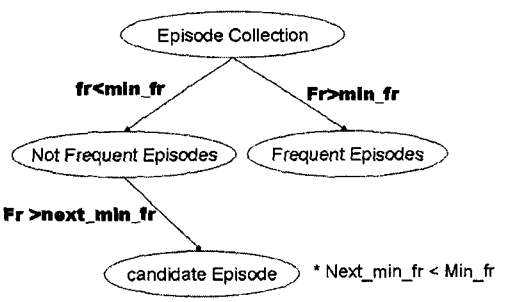

그림 2 빈발 에피소드와 후보 에피소드

3.3 로그 분할에서 오는 문제 해결

점진적 로그 분석으로 인한 로그 분할에서 오는 문제 해결로 이벤트 보정이 이루어져야 한다. 최신구간내의 로그 분석 회수는 초기값에 따라 로그가 분할되어 여러 이벤트열이 생긴다. 즉 최신구간내에는 여러 이벤트열이 연결된 구조로 되어 있다. 점진적 로그 분석시 이벤트열 사이의 연결 형태는 그림 3과 같은 세 경우가 발생한다.

관련 연구에서 설명한 바와 같이 에피소드의 빈도는 윈도우 스캐닝을 통하여 계산되어지고 이벤트열의 첫 이벤트가 포함되는 윈도우부터 스캐닝을 시작한다. 점진적 로그 분석으로 인하여 위 그림 3의 2와 3의 경우는 win-1만큼의 중복된 윈도우 스캐닝 일어나고 있다. 그러나 이벤트열 *Sn-1*의 로그 분석시 이벤트 B가 인식되지 못하고, 반면 이벤트열 *Sn*의 로그 분석시 이벤트 A가 인식되지 못하여 결국 에피소드 A-B의 빈도 계산은 실패하게 된다. 따라서 이벤트열 *Sn-1*의 뒷부분 win-1만큼 이벤트열 *Sn*에 앞부분에 보정하여 올바른 빈도의 계산으로 패턴 누락을 방지한다.

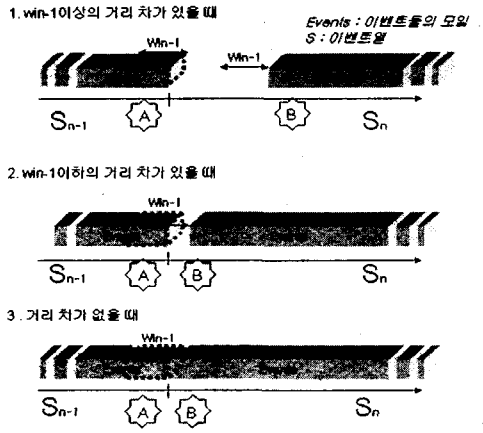

그림 3 이벤트 간의 세 연결 형태

4. 점진적 로그 분석 시스템의 제안

웹 서버는 계속적으로 로그 파일을 생성한다. 제안하는 점진적 로그 분석 시스템은 윈도우 폭(*win*), 최소 빈도수 (*min_fr*), 로그 분석 간격(*ans_interval*), 최신 구간(*recent_section*)을 파라미터로 빈발 에피소드를 발견하는 패턴 마이닝 엔진을 실행한다.

점진적 로그 분석 시스템은 그림 4와 같이 세 컴포넌트로 구성된다.

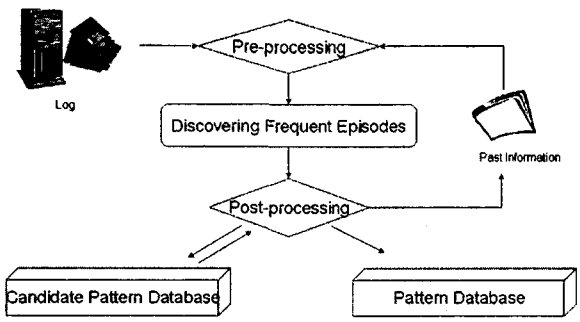

그림 4 점진적 로그 분석 시스템

우선 전처리(pre-processing) 컴포넌트의 역할은 기록된 로그는 마이닝 엔진의 효율을 높이기 위해 필요한 로그만을 정제한다. 또한 지난 정보(past information)의 기록을 통하여 위에서 언급한 3.3 이벤트열의 보정의 작업을 수행한다. 빈발 에피소드 발견을 위한 마이닝(discovering frequent episodes) 컴포넌트는 이벤트열의 윈도우 단위 스캔으로 각 에피소드의 빈도를 계산한다. 각 에피소드의 계산된 빈도는 후처리(post-processing) 컴포넌트를 통하여 최신 구간의 이동이 수행된다. 그리하여 빈발 에피소드 발견 위한 컴포넌트 수행 결과와 후보 패턴 데이터베이스의 병합으로 최종 패턴 데이터베이스를 구성하고, 임계값(*next_min_fr*)을 이용하여 후보 패턴데이터베이스도 재구성한다. 마지막으로 이벤트열 보정을 위한 지난정보(past information)를 재작성한다.

5. 결론

수많은 이용자와 다양한 엑세스 패턴이 존재하는 로그 파일에서 모든 패턴을 발견, 유지하는 것은 어렵다. 따라서 본 논문에서 최신구간을 활용하여 패턴의 최신성을 보장하고 최신구간내 패턴의 누락없이 모든 패턴을 발견할 수 있는 점진적 로그 분석 기법을 제안하였다. 이것은 주기마다 최신구간이 이동되면서, 동시에 패턴의 최신성 여부가 결정되고, 이동된 최신구간내 패턴이 될 후보 패턴을 미리 예측하여 합리적인 패턴 관리에 할 수 있다. 즉 최신구간을 고려한 특정 임계치로 후보 패턴을 추출하고, 적절한 패턴 제거 작업을 수행한다. 이러한 후보 패턴을 유지함으로서 분석 주기내의 패턴뿐만 아니라 최신구간내에 존재하는 모든 패턴을 발견할 수 있다. 따라서 일반적인 점진적 로그 분석 기법에서 간과된 대량의 로그에 숨겨진 패턴은 적어도 해당 최신구간내에서 모두 발견될 수 있고 최신성도 보증된다.

한편 앞으로 최신 구간 영역의 범위를 합리적으로 결정될 수 있는 방법에 대한 연구와 현실적인 실험이 필요하다.

참고 문헌

[1] Heikki Mannila, Hannu Toivonen, A. Inkeri Verkamo, "Discovering frequent episodes in sequences", *Proc. of 1st Int. Conference on Knowledge Discovery and Data Mining*, Montreal, Canada, Aug. 1995, pp.210-215.

[2] Hisayoshi Kato, Hironori Hiraishi, Fumio Mizoguchi, "Log summarizing agent for web access data using data mining techniques", *IFSA World Congress and 20th NAFIPS International Conference*, July. 2001, pp. 2642-2647 vol.5

[3] Florent Masseglia, Maguelonne Teisseire, Pascal Poncelet, "Real time web usage mining with a distributed navigation analysis", *Twelfth International Workshop,Proceedings*, 2002, pp.169 -174