사용자 생성 로그를 이용한 웹 분석시스템 설계 및 구현

고영대*, 이언배*
*한국방송통신대학교 평생대학원 정보과학과
e-mail: goindo@empal.com lub@mail.knou.ac.kr

Design and Implementation of Web Analyzing System based on User Create Log

Young-Dae Go* , Eun-Bae Lee*
Dept of Computer Science, Korea National Open University

요 약

인터넷 사이트가 증가하면서 서비스 제공자는 사용자의 요구나 행동패턴을 파악하기 위하여 웹 마이닝 기법을 활용한다. 하지만 서버에 저장된 웹 로그 정보를 활용한 마이닝 기법은 전처리 과정에 많은 노력이 필요하고 사용자의 행동패턴이나 요구를 정확하게 파악하는데 한계가 있다. 이를 극복하기 위해 본 논문에서는 사용자 생성 로그정보를 이용한 방법을 제안한다. 제안 방법은 기존 서버에 저장되는 로그파일이 아닌 사용자의 행동에 의해 웹 페이지가 로딩될 때 마다 웹 마이닝에 필요한 정보를 수집하여 DB 에 저장하는 방법을 사용하였다. 이때 기존 로그파일에 로딩시간과 조회시간, 파라메타 정보를 추가하여 보다 사실적으로 사용자의 행동패턴을 파악하고자 하였다. 이렇게 생성된 로그파일을 기 등록된 메뉴정보, 쿼리정보와 조합하면 웹 마이닝에 필수적인 데이터정제, 사용자식별, 세션식별, 트랜잭션 식별등 전처리 과정의 효율성을 향상시키고 사용자의 행동패턴파악을 위한 정보 수집을 용이하게 해준다.

1. 서론

인터넷을 이용한 정보제공과 활용이 활성화 되면서 많은 양의 컨텐츠가 웹사이트에서 서비스 되고 있다. 이런 서비스를 제공받는 사용자의 입장에서는 보다 신속하고 정확하게 원하는 정보를 획득하고자 하는 욕구가 강화되고 있고 서비스 제공자는 사용자의 욕구충족을 통해 서비스의 목적을 달성하고자 한다. 웹 마이닝은 이런 문제를 해결하기 위한 방법으로 이용되지만 웹 마이닝에 사용되는 로그분석은 로그파일 자체가 가지고 있는 정보의 제한과 사용자 이용환경 정보의 미흡으로 인한 한계를 가지고 있다. 이런 한계란 사용자 구분의 어려움과 중간 경로 누락으로 인한 전체 경로정보 확인이 어려운 이유로 추정경로를 사용하는 경우, DB 연동인 경우 정확한 페이지 구분이 어려운 경우, 그리고 사용자가 정보를 획득하기 위해 단순히 거쳐가는 페이지와 정보를 확인하는 페이지의 구분이 불명확한 것이다. 따라서 본 논문에서는 서버중심적인 로그파일을 사용하지 않고 사용자의 행동 즉 사용자가 클릭이나 URL 정보 입력에 의해서 웹 페이지를 호출할 경우 호출되는 웹 페이지 로딩 시마다 사용자 생성로그를 DB 에 저장되는 방식으로 웹 분석 시스템을 구현하고자 한다.

본 논문의 구성은 다음과 같다. 먼저 제 2 장에서는 구현하고자 하는 사용자 생성로그 시스템의 기반이 되는 웹 마이닝 연구와 기존 연구의 제한 점에 대해서 알아본다. 제 3 장에서는 사용자 생성로그에 필요한 요구사항을 정의하고 사용자생성 로그를 이용한 시스템 설계 및 구현에 필요한 시스템 구조와 로그파일 생성, 데이터 전처리 과정에 대해서 알아본다. 제 4 장 결론에서는 위 방식을 이용한 응용분야와 추후 연구과제에 대해서 알아봄으로써 웹 분석 시스템의 방향을 제시하고자 한다.

2. 관련연구

2.1 웹 마이닝의 정의와 분류

웹 마이닝은 웹 로그 파일에서 정보를 추출하기 위하여 데이터 마이닝 기법을 적용하여 유용한 새로운 정보를 찾으려는 시도라고 정의할 수 있다. [1] 이런 웹 마이닝은 크게 콘텐츠 마이닝(Contents Mining), 구조 마이닝(Structure Mining), 사용 마이닝(Usage Mining)으로 나누어진다. [2] [3] 웹 콘텐츠 마이닝은 온라인상에서 이용 가능한 정보를 자동적으로 찾아주는 마이닝 기법을 말하고 웹 구조 마이닝은 콘텐츠 구성에 관한 자료를 마이닝하는 것으로 웹 환경에서 참조한 페이지와 참조된 페이지 사이의 관계와 구조에 대한 정보를 찾거나 웹사이트나 웹 페이지에 대한 요약된 구조를 생성시키는 마이닝 기법이다. 웹 사용 마이닝은 웹 서버로부터 사용자들의 접속 유형

을 자동적으로 찾아주는 마이닝 기법으로 Web Log Mining 이라고도 한다. [4] 즉, 사용자들이 브라우징 했던 기록을 남기는 웹 서버 로그로부터 사용자들의 접속 유형을 발견하는 것을 목적으로 하는 마이닝 기법이다. 사용되는 데이터는 로그파일, 고객등록정보, 설문데이터, 거래정보 등이 있다.

2.2 웹 마이닝과정

웹 마이닝 과정은 전처리 과정, 패턴발견, 패턴분석으로 이루어 지는데 전처리 과정은 로그정보를 분석이 용이한 상태로 정제 가공하는 단계로 데이터정제, 방문자식별, 세션식별, 경로완성, 트랜잭션식별의 단계로 구성된다. [5] 전처리 과정은 웹 접근 로그에서 필요한 항목을 추출하여 불필요한 노이즈들을 제거하는 데이터 정제 과정과 각 방문자 별로 로그에 있는 정보를 구별해 주는 방문자 식별과정, 각 페이지 간의 시간의 차이를 이용해서 각 방문자들 간의 로그 정보를 논리적인 작업단위인 세션으로 나누는 과정, 캐시로 인하여 로그에 남지 않는 페이지를 유추해서 방문자가 이동한 페이지들의 경로를 완성하는 과정, 그리고 데이터 마이닝을 하기 위해 페이지들을 그룹화 시켜서 분석 목적에 의미 있는 단위인 트랜잭션 데이터로 세션을 나누거나 병합하는 단계로 구분되어진다. 트랜잭션 식별에는 참조 길이, 최대 전진 길이, 시간간격과 같은 요소를 사용한다. [6] 패턴발견 단계에서는 정제된 데이터를 가지고 패턴기법을 이용하여 사용자들의 패턴을 찾아내는 과정으로 찾아낸 패턴을 이용하여 사용자들의 성향을 파악하고 행동을 예측하는 등의 대응전략을 세울 수 있다. [7] 패턴분석은 찾아낸 규칙이나 패턴 중에서 유용하고 의미 있는 규칙과 패턴을 찾아내서 그것을 이해하고 해석하는 단계로서 발견된 패턴을 분석하는 기법으로는 시각화와 OLAP 가 있다.

2.3 기존연구 및 로그분석의 제한 점

웹마이닝과 관련된 기존 연구는 대부분 로그분석 데이터에 대한 정확도향상이나 전처리 과정에 대한 효율성 향상 또는 분석데이터를 활용한 웹사이트 설계방안을 제시하는데 중점을 두고 이루어 졌는데 이런 연구의 대부분은 웹 서버에 저장된 로그파일을 이용하는 방법으로 진행된다.

하지만 이런 로그파일 이용방법은 다음과 같은 한계를 가지고 있다. 첫째 그림에서와 같이 사용자에 대한 구분이 접속 IP 단위로 되기 때문에 내부 IP 를 사용하거나 유동 IP 를 사용할 경우 IP 정보가 사용자를 구분하기에 부정확하다. 둘째 웹 서버 로그 정보는 웹 서버에 요청된 트래픽만 측정하므로 중간의 프록시 서버나 웹 브라우저의 캐시에 의해 처리되는 페이지는 측정이 불가함으로 전처리 과정에서 누락페이지에 대한 추가작업이 필요하다. [8] 셋째 웹 환경에서 중요한 페이지 로딩시간, 정보조회시간 등 사용자 이용환경을 고려한 분석자료가 미흡하여 사실적인 사용자 행동패턴 분석이 미흡하다. 넷째 다량의 로그파일로 인해 서버에 많은 공간이 필요하며 다량의 파일 중 의미 있는 정보 산출을 위해 전처리 과정에서 많은 부하를 발생시킨다. 다섯째 DB 연동페이지인 경우 파라메타 값에 의해 페이지가 구분되는데 get 방식으로 처리되는 파라메타의 경우 파일명으로 구분이 가능하지만 post 방식으로 처리되는 페이지인경우 정확한 파악이 불가능하다. 이런 로그분석의 한계로 인해 항상 오차범위가 존재하고 이런 오차범위로 인해 실제로 우리가 알고자 하는 사용자의 기록과 실제 사용자의 행동에 차이가 있을 수 있다.

따라서 본 논문에서는 사용자 행동중심의 접근 로그를 생성해서 활용하는 웹 사용 마이닝 기법에 대해서 연구하고자 한다.

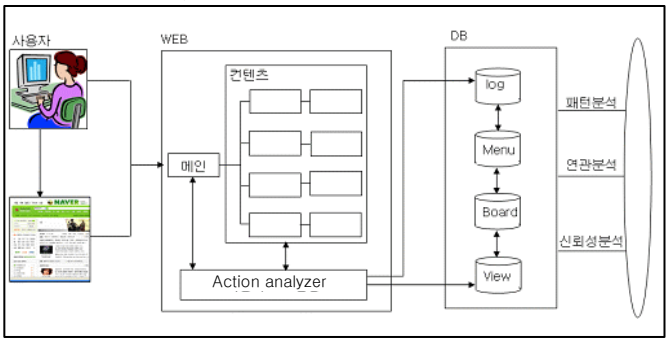
3. 시스템 설계 및 구현

3.1 요구사항 정의

시스템 설계 및 구현에 있어서 기존 로그저장 방법에서 정확도향상을 위해서 다음과 같은 요구사항을 전제로 한다. 첫째 로그정보에 불필요한 정보가 포함되어서 이를 제거하기 위한 데이터 정제과정이 필요하지 않도록 기존 로그파일에서 기록되는 이미지 파일이나 css, js 파일 등의 파일들은 기록하지 않고 HTML 파일 또는 DB 연동 파일만 기록한다. 둘째 사용자 구분에서는 공유 IP 나 가변 IP 를 구분하기 위해 IP 정보 외 추가적인 정보를 이용하여 사용자 구분이 용이하도록 한다. 셋째 프록시 서버나 캐시에 의한 정보 누수가 없도록 사용자가 조회한 모든 웹 페이지 정보를 저장한다. 넷째 DB 연동정보인 경우 정보자원의 정확한 분류를 위해 post 방식으로 처리되는 파라메타 값을 저장한다. 다섯째 사용자 행동에 맞는 세션구분 및 트랜잭션 구분을 위한 로그정보가 필요하다.

3.2 시스템 구조

그림 1 은 본 논문에서 사용된 사용자 행동중심 로그에 대한 전체적인 구조를 나타낸다.



(그림 1) 사용자로그 수집구조

사용자가 웹 서버에 요청을 하면 웹 페이지 내에 포함된 Action analyzer 에 의해 파악된 사용자의 정보가 DB 내 로그정보에 저장된다. 저장된 로그정보는 미리 정의된 메뉴정보를 이용하여 내부링크정보는 코드화 처리된다. Action analyzer 는 사용자의 정보를 수집하기 위한 파일로 모든 화면에 추가하는데 그 방법으로는 서버 측 부하 감소와 브라우저 종속적으로

작동하기 위해 JavaScript 형식으로 제작되어 공통파일로 호출하게 된다.

로그정보는 CLF(Common Log Format)정보 중 접근시간, 호스트 명, 파일명, 파일크기에 추가적으로 사용자 ID, 레퍼파일명, 로딩시간, 조회시간, 파라메타로 구성된다. 접근시간은 페이지를 호출한 시간을 나타내고 호스트 명은 사용자의 접근 IP 를 표시한다. 파일명은 호출된 페이지의 URL 정보를 나타낸다. 파일크기는 호출된 페이지 내에 포함된 이미지와 텍스트 정보 등의 전체용량을 KB 단위로 나타내고 사용자 ID 는 사용자 구분을 위해 추가적으로 사용하는 부분이다. 레퍼파일명은 이전경로를 나타내는 것으로 세션 구분을 위해 사용된다. 로딩시간은 사이트 내 각 페이지의 속도 체크를 위해 사용되는 요소로 사이트의 안정적인 정보제공을 위해 필요하다. 조회시간은 사용자가 정보조회를 위해 해당 페이지가 잔류한 시간으로 세션구분을 위해 필요하다. 파라메타는 DB 연동 시 파라메타 정보에 의해 구분되는 페이지를 위해 post 방식으로 넘어 오는 변수들을 저장한다.

로그정보는 그림 2 와 같이 최초호출 시 산출된 기본정보인 접근시간, 호스트 명, 파일명, 파일크기, 사용자 ID, 레퍼파일명, 파라메타변수등을 등록하고 로딩 완료 시에 로딩시간을 계산하고 페이지 브라우저 종료, 새로 고침, 페이지이동 등의 정보조회 종료 시에는 조회시간을 산출해서 초기 등록된 로그정보에 로딩시간, 조회시간을 수정하는 방식으로 처리된다.

```
//최초호출시 기본정보 산출 및 입력
Start <- print(Date);
hostname <- print(IP)   ;
filename <- print(filename);
filesize  <- print(filesize);
user <- print(user);
ref <- print(ref);
par <- printpar);

call  TimeInsert('1',Start,hostname,filename,filesize,user,ref,par,0,0) ;


//페이지 종료여부 확인
procedure checkExit()
    if self.screen> 9000 then
        call TimeUpdate('종료');
    else
        call TimeUpdate('이동');
    end if
end checkExit


//페이지 조회시간 산출
procedure TimeUpdate()
    End <- print(Date);
    View <-  End- Start ;
    call TimeInsert("2",Start,hostname,filename,filesize,user,ref,par,Load,View) ;
end TimeUpdate


//로그정보 DB반영
procedure TimeInsert(div,Start,hostname,filename,filesize,user,ref,par,Load,View)
    if div = "1" then
        read "로그정보 입력";
    else
        read "로그정보수정";
    end if
end TimeInsert
```

(그림 3) 로그정보 저장 알고리즘

3.3 데이터 정제 및 사용자 구분

사용자 생성 로그 파일에서는 웹 페이지 정보 이외에는 저장되지 않으므로 별도의 데이터 정제과정은 필요하지 않다. 사용자 구분을 위해 사이트 최초 접속 시 시간정보를 쿠키 값으로 부여하여 이 값을 사용자 ID 필드에 저장하여 이용한다. 사용자가 경로이동 중에 로그 인을 할 경우에는 사용자 ID 필드에 등록된 쿠키 값을 수정하여 기존 시간정보 사용자와 로그인 사용자가 동일한 사용자로 구분되도록 처리한다. 이렇게 저장된 로그 값에서 사용자는 IP+사용자 ID(시간정보)의 결합 값으로 구분 가능하다.

3.4 웹 페이지 조회시간

로딩속도는 네트워크 속도나 웹 페이지 크기, DB 연동여부에 따라 같은 사이트 내에서도 많은 차이가 발생하고 일반적으로 사용자가 감내할 수 있는 3~5 초 정도의 시간을 초과하는 경우에는 전체적으로 사이트의 신뢰도를 저하 시키는 요인이 된다. 정보 조회 시간은 로딩 후에 실제 사용자가 해당 페이지에서 정보를 조회한 시간으로 기존 time stamp 를 이용(n+1 time stamp - n time stamp) 해서 소요 시간을 산출할 수 있지만 이 방식은 네트워크 환경이나 로컬캐시 사용여부, 로딩시간과 정보 조회 시간의 혼합으로 인해 정확한 웹 페이지 조회시간 산출하기 힘든 단점이 있다. 하지만 이 두 가지 시간은 웹 마이닝에 있어서 중요한 정보가 된다. 따라서 다음과 같은 방법으로 페이지 로딩시간과 웹이지 조회시간을 산출하고자 한다.

로딩시간 = 페이지 호출완료시간-페이지 호출시작시간

조회시간 = 페이지 이탈시간 - 페이지 호출완료시간

3.5 세션구분

세션은 사용자가 웹사이트에 목적을 가지고 방문했을 때 최초 방문시점에서부터 목적을 달성하고 연결을 종료한 순간을 하나의 세션으로 정의한다. 본 논문에서는 3.3 에서 정의된 사용자 구분에서 나온 결과값을 시간 순으로 정렬하여 세션을 구분하는데 다양한 사용자의 행동을 고려하여 다음과 같은 경우에 새로운 세션으로 간주한다. 레퍼경로가 현 사이트 도메인과 다른 경우 다른 사이트에서 링크정보를 이용해 들어온 경우이므로 새로운 세션으로 간주한다. 레퍼경로가 없는 경우는 브라우저를 생성해서 처음 경로이동을 한경우 이므로 이때도 새로운 세션으로 간주한다. 사용자가 정보조회 중이거나 정보조회 완료 후 해당 브라우저를 닫지 않으면 조회시간이 계속 증가함으로 이때는 30 분 이상 된 조회시간이 발생한 경로 이후를 새로운 세션으로 간주한다.

3.6 트랜잭션구분

트랜잭션은 사용자의 웹 방문 경로의 정보를 군집화에 사용할 수 있도록 군집화 목적에 따라 경로의 길이를 정하는 것으로써 군집화를 위한 패턴의 기본 단위가 된다. 기존 트랜잭션 구분법인 참조 길이, 최대 전진길이, 시간간격과 같은 요소대신에 평균조회시간을 이용한 구분법을 이용하고자 한다. 평균조회시간이란 전체조회시간에서 세션 내 페이지 수를 나눈 것으로 이 평균 조회시간보다 큰 페이지 별로 트랜잭션을 군집화 한다.

예를 들어 표 1 과 같이 경로정보가 있다면 평균조

회시간은 (3+30+4+3+50+2+4+35)*8 = 16.375 초이다. 이때 16.4 초 보다 작은 1,3,4,6,7 번은 평균조회시간보다 큰 2,5,8 번으로 이동하기 위한 이동경로이므로 목적페이지인 2,5,8 을 기준으로 트랜잭션을 구분한다.

<표 1> 경로정보 예

번호	1	2	3	4	5	6	7	8
시간	3	30	4	3	50	2	4	35

3.7 서버로그와 사용자로그 비교분석

사용자 생성 로그 파일은 표 2 에 나와 있는 것처럼 사용자구분, 프록시 서버/캐시영향, 이용환경고려, 파라메타 확인 등 기존 서버로그 방식에서의 제한 점들을 개선할 수 있다. IP 외에 추가적으로 사용자 ID 를 사용함으로써 사용자 구분을 좀더 명확하게 할 수 있고, 프록시 서버나 캐시의 영향을 받지 않고 사용자가 이동하는 모든 페이지에 대한 로그파일을 생성할 수 있다. 또한 웹사이트 관리에서 중요한 요소인 로딩시간, 조회시간을 이용한 사용자 이용환경분석 데이터를 생성 할 수 있고 get 방식에 추가하여 post 방식으로 처리되는 파라메타값도 저장함으로써 페이지를 좀더 세부적으로 구분할 수 있다. 표 3 에 나와 있는 것처럼 사용자 구분 면에서 약 19%의 성능향상과 일일 생성 로그 량에서도 기존 26.7M 에 비해 약 80%이상 감소한 3.4M 정도로 저장공간을 절약할 수 있고 전처리 과정에서 데이터정제과정이나 경로완성과정을 생략할 수 있는 장점이 있다.

<표 2> 사용자 로그방식 개선점

구분	서버로그	사용자로그
사용자구분	IP	IP + 사용자 ID
프록시/캐시영향	있음(추가작업필요)	없음
이용환경 고려 값	없음	로딩시간, 조회시간
파라메타 확인	GET	GET/POST

<표 3> 로그방식 비교분석

구분		9.3	9.4	9.5	9.6
서버로그	사용자	3739	3605	3394	3287
	사용량(M)	29.2	27.8	25.3	24.6
사용자로그	사용자	4426	4306	4092	3904
	사용량(M)	3.8	3.6	3.3	3.2

4. 결론

이상에서 살펴본 바와 같이 DB 에 저장되는 사용자 생성로그를 이용한 웹사이트 분석방법은 기존 서버저장 방식에 비해서 사용자 구분, 저장공간절약, 전처리 과정에서 데이터정제과정이나 경로완성과정 생략 등의 많은 장점을 가지고 있다. 또한 패턴발견과 분석에 사용되는 기초 값이 사용자의 행동패턴을 좀더 현실적으로 표현함으로써 분석자료의 신뢰도를 향상시킨다. 이런 분석방법을 활용하면 사용자가 정보에 보다 쉽게 접근하도록 사이트 구조를 개선할 수 있고 로딩속도 문제페이지 발견을 용이하게 하여 효율적인 사이트 관리가 가능해 지고 DB 쿼리 정보와 DB 서버 내 실행뷰를 비교하여 DB 마이닝을 위한 기초자료로 활용할 수 있다. 앞으로 연구에서는 패턴발견/분석기법을 적용한 웹사이트 분석사례에 대해서 연구하고자 한다.

참고문헌

[1] Gordon S.Linoff, Michael J.A.Berry : “Mining the Web” wiley, 2001

[2] M. Kitsuregawa, T. shintani and pramudiono, “Web Mining and its SQL Based Parallel Execution, “ in Proc. ITVE. Queensland, Australia, p.128-134, Jan. 2001.

[3] R Cooley, B. Mobasher, and J. Sricasrava. “Web Mining: Information and Pattern Discovery on the World Wide Web,“ in Proc. IEEE 9th ICTAI., pp.558-567, CA, Nov 1997

[4] J. Strivastava, R Cooley and M. Deshpand, "Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data," in SIGKDD. Explorations, Vol. 1, Issue 2, pp. 12-23, Jan. 2000.

[5] R Cooley, B. Mobasher, and J. Sricasrava. “Data Preparation for Mining World Wide Web Browsing Patterns, “ Journal of Knowledge and Information Systems, Vol. 1, No1 1999.

[6] 김우주, “웹 마이닝의 사용자 패턴 분석 정확도 향상에 관한 연구”, 서강대학교 석사학위 논문, pp. 24-25, 2002

[7] B. Mobasher, N Jain, E. H. Han, and J.serivastava, “Web Mining : Pattern Discovery from World Wide Web Transaction, “ Technical Report TR-96050, Department of Computer Science, University of Minnesota, Minnieapolis, 1996

[8] I.Y. Lin, X.M. Hung, and M.S. Chen, "Capturing user access patterns in the web for data mining" Proceedings of the 11th IEEE International Conference Tools with Artificial Intelligence, Chicago, IL, pp.22-29, 1999