

# 사용자 접근 패턴 분석을 이용한 적응형 웹 사이트 구축에 관한 연구

고경자, 김인철

경기대학교 전자계산학과

e-mail: rain825@kuic.kyonggi.ac.kr

## A Study on Adaptive Web Site Construction by Analyzing User Access Patterns

Kyong-Ja Ko, In-Cheol Kim

Dept. of Computer Science, Kyonggi University

### 요 약

본 논문에서는 웹 사이트에 접근하는 사용자 접근 패턴을 학습하여 정보 제공이 보다 용이한 구조로 자동 개선시켜 나가는 적응형 웹 사이트를 구축하고자 한다. 즉, 기존 웹 사이트의 구조를 가능한 한 파괴하지 않는 범위 내에서 웹 사이트를 변경하고자 관련성은 높으나 접근 경로가 긴 문서들의 클러스터를 찾아내고, 이들에 대한 별도의 색인 페이지를 생성하여 웹 사이트 내에 위치시킨다. 이를 위하여, 먼저 대용량의 웹 서버 로그 데이터들을 대상으로 순차 패턴 탐색 방법인 AprioriAll 알고리즘을 적용함으로써 웹 문서간의 충분한 연관성 지지도를 갖는 사용자 순차 접근 패턴을 분석해낸다. 사용자 순차 접근 패턴 분석을 통해 관련성 있는 문서들의 집합을 알아낸 후, 웹 사이트의 하이퍼 링크 구조 정보를 고려하여 접근 경로가 긴 문서들만을 골라 웹 문서 클러스터를 생성시킨다. 이러한 웹 문서 클러스터들에 대한 색인 페이지를 추가 생성하여 제공함으로써 사용자들의 보다 효과적인 정보 접근을 지원할 수 있는 웹 사이트로의 변경이 가능하다.

### 1. 서론

WWW(World Wide Web) 환경에서 하나의 웹 사이트(web site)는 하이퍼 링크(hyperlink)로 연결되어 있는 수많은 웹 문서들의 집합으로 이루어져 있다. 초기의 웹 사이트는 각 웹 문서들이 지닌 의미와 문서들 간의 상호 관계 등을 고려해 최상의 사이트를 구현하고자 하는 웹 마스터의 의도가 반영

된 것이다. 그러나 동적인 사용자들의 요구를 반영하여 보유 정보를 효과적으로 제공할 수 있도록 하기 위해서는 웹 마이닝(web mining) 과정을 통한 지속적인 사이트의 변경과 갱신 작업이 요구된다. 일반적으로 웹 마이닝은 대용량의 문서들이나 웹 사이트에 관련된 데이터들을 분석함으로써 유용한 정보나 지식을 발견해 내는 기술을 말한다. 특히, 웹 사이트에 대한 사용자들의 접근 로그 데이터는 사용자들의 일반적

인 정보 접근 패턴을 알아낼 수 있는 중요한 자료를 제공한다. 이러한 자료를 통해 사용자 접근 패턴을 분석해냄으로써 웹 사이트의 구조와 표현 방식을 개선시킬 수 있다. 본 논문에서는 사용자 접근 패턴에 대한 학습 과정을 통해 웹 사이트의 구조나 외형을 자동적으로 개선시켜 나가는 적응형 웹 사이트의 구축 방안을 제시하고자 한다. 이를 위해 웹 서버의 대용량 로그 데이터를 대상으로 하여 순차 패턴 탐색 방법인 AprioriAll 알고리즘을 적용함으로써 웹 문서간의 충분한 연관성 지지도(support of association)를 갖는 사용자 순차 접근 패턴(sequential access pattern)을 분석한다. 관련성 있는 문서들의 집합을 사용자 순차 접근 패턴을 통해 알아낸 후, 웹 사이트의 하이퍼 링크 구조 정보를 고려하여 접근 경로(access path)가 긴 문서들만을 골라 웹 문서 클러스터(cluster)를 생성시킨다. 생성된 클러스터를 구성요소로 한 색인 페이지들(index pages)을 기존 웹 사이트에 자동 생성시킴으로써 웹 사이트의 구조를 가능한 한 파괴하지 않는다는 특성을 지니게 된다. 이와 같이, 색인 페이지의 자동 생성은 사용자들을 대상으로 원하는 웹 문서를 빠르게 접근 할 수 있는 서비스를 제공한다.

본 논문의 2장에서 적응형 웹 사이트에 대한 관련 연구를 살펴보고, 3장에서 순차 접근 패턴 탐색을 위한 분석 데이터의 전처리 과정을 살펴본다. 이어지는 4장에서 AprioriAll 알고리즘을 이용한 사용자 순차 접근 패턴 탐색하는 과정을 설명하고, 5장에서 하이퍼 링크 정보를 적용한 웹 문서 클러스터 생성을 설명한다. 6장에서 색인 페이지의 생성에 대해 살펴보고, 마지막으로 7장에서 결론을 맺는다.

## 2. 관련 연구

적응형 웹 사이트를 구현하는 방법으로 맞춤화(customization)는 사용자 개인의 접근 패턴을 학습한 후 개인의 요구와 기호를 고려해서 웹 사이트를 개선시켜 나가는 방식이다. 구현 시스템의 예로 WebWatcher[6]는 사용자의 접근 경로를 추적해 학습한 후, 사용자의 긍정적인 반응을 근거로 성공적인 항해(navigation)가 된 패스(path)들만을 이후 접근한 사용자들에게 제시한다. AVANTI 프로젝트[10]에서는 사용자가 항해 초기에 기입한 개인 정보와 더불어 지금까지 위해 온 행동을 고려해서 다음 이동 경로를 추적하면서 최종 목표점을 예측한다. 이와 같은 작업을 수행하는 데는 웹 서버가 제공하는 기본적인 정보들 이외에도, WebThread 와 같은 서비스를 추가로 이용해 사용자가 한 세션 동안 거처간 페이지들을 추출해 내기도 한다. 적응형 웹 사이트를 구현하

는 또 다른 방법인 최적화(optimization)는 특정 개인이 아니라 접근할 가능성을 지닌 모든 사용자들을 위해 웹 사이트를 개선하는 방식으로, 이미 접근한 사용자들을 대상으로 접근 패턴을 학습해 나간다. 예로, Perkowiz 와 Etzioni 가 구현한 시스템에서는 과제를 제출 마감 기간 이후의 해 집합(solution set)에 접근하는 학생들의 패턴을 파악한 후, 빈번히 접근된 해 집합을 웹 사이트 구조상 식별이 잘 되는 곳으로 이동시키는 작업을 수행한다.

웹 마이닝을 위한 분석 자료에는 웹 자체가 지니고 있는 자원들(resources)을 분석하는 웹 콘텐츠 마이닝(web content mining)과 사용자 접근 패턴을 발견하는 웹 유지지 마이닝(web usage mining)으로 나눌 수 있다. 전자의 분석 대상이 되는 메타 정보(meta-information)는 웹 사이트의 내용(content), 구조(structure), 조직(organization)에 대한 정보를 일컫는 것으로, 대표적인 구현 시스템으로 STRUDEL[11]이 있다. 후자의 분석 대상이 되는 서버 로그 파일은 접근 빈도수 뿐만 아니라 시간 의존적인 경향들을 분석할 수 있는 자료를 제공한다. 사용자 접근 패턴을 알아내는데 적용되는 알고리즘에는 조건부 확률 알고리즘과 대표적인 연관규칙(association rule)탐사 방법인 Apriori[1] 알고리즘, 그리고 Apriori 알고리즘을 기반으로 순차 패턴까지 고려한 AprioriAll[12] 알고리즘이 있다. Perkowiz 와 Etzioni 가 제안한 조건부 확률 알고리즘에서는 문서 P1 과 P2 가 있다고 가정 했을 때 P1 을 방문한 사용자가 P2 을 방문할 확률과 P2 을 방문한 사용자가 P1 을 방문할 확률 중 최솟치를 선택해 동시 발생(co-occurrence)빈도수로 정하고 이를 행렬과 그래프로 도식화해서 클러스터를 생성한다. 반면, Apriori 알고리즘에서는 최소 지지도(support degree)와 신뢰도(confidence degree)를 만족하는 문서들의 집합으로 클러스터를 생성하고, AprioriAll 알고리즘에서는 이를 기반으로 문서의 운행 순서까지 고려해서 클러스터를 생성한다. Apriori 알고리즘에서 파생된 알고리즘에는 AprioriTID, AprioriHybrid, DHP[14] 등이 있고, AprioriAll 알고리즘에서 파생된 알고리즘으로는 AprioriSome, AprioriAll, GSP[13], SPADE 등이 있다.

운행 패턴을 지니는 웹 상에서 관련성 있는 웹 문서 집합을 추출하기 위해서는 사용자가 전진한 경우에 거처간 웹 문서 집합들(maximal forward references)을 구한 후, 이 가운데 각종 알고리즘의 적용 결과로 빈번히 발생한 웹 문서 집합들(large reference sequences)을 구한다. 마지막 단계에서 빈번히 발생한 웹 문서 집합을 모두 포함할 수 있는 웹 문서 집합들(maximal reference sequences)을 구해내는 과정을 거친다.

### 3. 전처리 과정

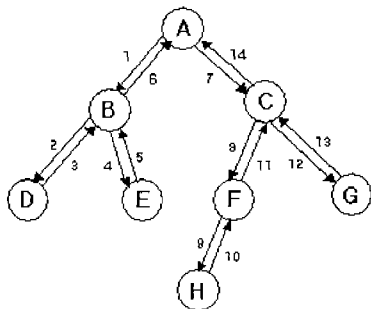
[그림 1]과 같이 로그 파일 안에는 사용자의 IP address, 사용자의 ID, 웹 문서에 접근한 날짜와 시간, 요청 방법, 접근한 문서의 URL, 데이터 전송에 사용된 프로토콜, 에러 코드, 전송 바이트 수 등에 대한 정보가 들어 있다. 서버 로그 파일은 웹 마이닝 작업에 불필요한 항목(entry)들을 포함하고 있어서 웹 로그 파일을 대상으로 순차 접근 패턴을 분석하기 위한 선행작업이 요구된다. 단계별로 살펴보면 다음과 같다.

```

203.249.22.75 -- [25/Feb/2000:15:57:46 +0900] "GET / HTTP/1.1" 304 -
203.249.22.75 -- [25/Feb/2000:15:57:46 +0900] "GET /last1.jpg HTTP/1.1" 304 -
203.249.22.75 -- [25/Feb/2000:15:57:47 +0900] "GET /main.html HTTP/1.1" 304 -
203.249.22.75 -- [25/Feb/2000:15:57:47 +0900] "GET /menu.htm HTTP/1.1" 304 -
    
```

[그림 1] 웹 로그 데이터

1 단계: 서버 안의 정제되지 않은 데이터를 어구 해부 (parsing)하는 과정과 URL의 접미사(suffix)와 같은 불필요한 로그 속성(entry)들 즉, gif, jpeg, GIF, JPEG, jpg, JPG, map 등을 제거하는 정화(cleansing)과정을 거친다.



[그림 2] 웹 문서 운행 경로

2 단계: 사용자 별로 한 세션 동안 거쳐간 웹 문서 운행 경로(traversal path)를 알아낸다. [그림 2]를 예로 들면, {ABDBEBACFHFCGCA}가 된다.

3 단계: 후진한 경우를 제외하고 전진한 경우에 거쳐간 웹 문서(maximal forward reference)들을 알아낸다. [그림 2]를 예로 들면, {ABD}, {ABE}, {ACFH}, {ACG}가 된다.

4 단계: 3 단계에서 얻은 문서 집합의 모임을 하나의 트랜잭션(T)으로 간주하고 트랜잭션에 한번 등장한 문서는 먼저 등장한 것을 우선해서 다시 등장하지 않는 것으로 한다.

$$T = \{ABDABEACFHACG\}$$

$$\downarrow$$

$$T = \{ABDECDFHG\}$$

5 단계: 하이퍼 링크 구조상 시작점이 되는 문서(entrance page)는 내용에 충실한 문서이기 보다는 다른 문서를 접근하기 위해 거쳐가는 문서의 성격이 강하므로 트랜잭션에서 제외시킨다.

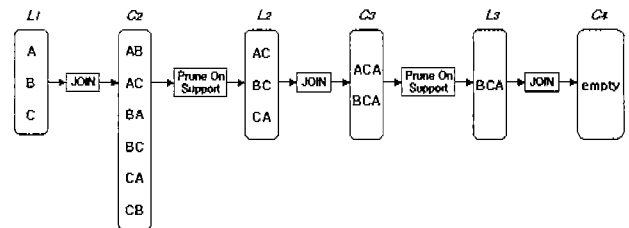
$$T = \{ABDECDFHG\}$$

$$\downarrow$$

$$T = \{BDECDFHG\}$$

### 4. 순차 패턴 접근 탐색

일반적으로, AprioriAll 알고리즘을 포함한 순차 패턴 알고리즘의 결합 형태는 [그림 3]과 같다. 우선, 데이터 베이스



[그림 3] 순차 패턴에 대한 결합 알고리즘의 형태

스를 스캔(scan) 하여 문서 각각으로 구성되는 순차 후보 1-문서 집합들의 모임  $C1$  을 생성한다. 만약, 최소 지지도를  $\alpha$  라고 가정하면 순차 후보 1-문서 집합들의 모임  $C1$  에서  $Sup \geq \alpha$  를 만족하는 순차 빈발 1-문서 집합들의 모임  $L1$  을 구할 수 있다. 다음 패스에서, 순차 후보 문서 집합들의 모임  $C2$  를 생성하기 위해 결합 연산자를 이용해 순서까지 고려한  $L1 * L1$  구한다. 두 번째 패스에서 다시 데이터 베이스를 스캔 한 후, 순차 후보 2-문서 집합들의 모임  $C2$  에서  $Sup \geq \alpha$  를 만족하는 순차 빈발 2-문서 집합들의 모임인  $L2$  을 구한다. 다음 패스에서, 순차 후보 문서 집합들의 모임  $C3$  를 생성하기 위해  $L2$  를 기반으로 순서까지 고려된 등장 가능성이 있는 순차 후보 3-문서 집합들의 모임  $C3$  구한다. 세 번째 패스에서 데이터 베이스를 스캔 한 후, 순차 후보 3-문서 집합들의 모임  $C3$  에서  $Sup \geq \alpha$  를 만족하는 순차 빈발 3-문서 모임인  $L3$  을 구한다. 이러한 방식으로 여러 패스를 거치면서 순차 후보 문서 집합들의 모임이 생기기 않을 때 까지 반복한다.

3장에서 본 바와 같이 다수의 사용자를 대상으로 이루어지는 선행 작업을 통해 T를 구하고 이를 순차 접근 문서(document sequences)항목으로 묶어 데이터 베이스를 구축한다. 여기서 최소 지지도를 2 이상이라 가정하고, AprioriAll 알고리즘

집합을 적용한다. 적용 결과는 [그림 4] 와 같이 매 패스마다  
순차 접근 문서 (*ds*)

<{B} {D} {E} {C} {F} {H} {G} >  
<{B} {D} {H} {G}>  
<{B} {D} >  
<{C} {F} {H} {B} {E} >  
<{B} {D} {G} >  
<{C} {F} {H} {G} >  
<{B} {D} {H} >  
<{B} {D} {G} >  
<{C} {F} >  
<{B} {C} {G} >

데이터 베이스  
순차 빈발 1-문서 집합  
지지도

B  
8  
C  
5  
D  
6  
E  
2  
F  
4  
G  
6  
H  
5

패스 1을 거친 후( *L1* )

순차 빈발 2-문서 집합  
지지도

BC  
2  
BD  
5  
BG  
5  
BH  
3  
CF  
4  
CG  
3  
CH  
3  
DG  
4

DH  
2  
FG  
2  
FH  
3  
HG  
2

패스 2를 거친 후( *L2* )  
순차 빈발 3-문서 집합  
지지도

BDG  
4  
CFH  
3

패스 3를 거친 후( *L3* )

[그림 4] AprioriAll 알고리즘의 적용 결과

순차 빈발 문서 집합의 모임이 구해진다. 이 과정이 끝나면

순차 빈발 아이템 집합의 모임에서 최대 순차 빈발 문서 집합들 (maximal reference sequences)의 모임을 구해내는 단계를 거친다. 그 결과 모든 패스의 순차 빈발 문서를 포함할 수 있는 최대 빈발 문서 집합의 모임들만을 구하면 <BC>, <BH>, <CG>, <DH>, <FG>, <HG>, <BDG>, <CFH>가 된다.

## 5. 웹 문서 클러스터 생성

웹 문서 클러스터를 생성하기 위해서 4장에서 구한 순차 빈발 아이템 집합의 모임을 바탕으로 웹 사이트의 하이퍼링크 구조 정보를 적용하는 과정을 거친다. [그림 2]가 초기에 웹 마스터가 만든 가상 사이트 'S-site'의 하이퍼링크 구조라면, 노드(node)는 웹 문서를 대표하고 아크(arc)는 문서들 간의 연결 상태를 의미한다. 노드와 노드 사이의 길이(length)를 1이라고 가정하고, 매트릭스(matrix)의 행과 열에 각각 웹 문서들을 나열한 후, 해당 행과 열이 만나는 지점에 두 문서 간의 최단 접근 경로 길이를 표현하면 [그림 5]과 같다. 이를 바탕으로 클러스터 생성 단계에서  $Lenth \geq \gamma$ 를 만족하는 문서만을 추출하게 되는데, 여기서는  $\gamma$ 를 4라고 가정한다.

A	E
B	2
C	1
D	3
E	2
F	0
G	4
H	4
	5
A	F
0	2
1	3
1	1
2	4
2	4
2	0
3	2
	1
B	G
1	2
0	3
2	
1	

1  
4  
4  
2  
0  
3

H  
3  
4  
2  
5  
5  
1  
3  
0

D  
H  
G

[그림 5] 두 문서간 최단 경로

접근 경로가 긴 문서들만을 구성요소로 한 웹 문서 클러스터를 생성시키기 위한 알고리즘은 다음과 같다.

- 웹 문서 클러스터 결정 알고리즘**
1. 최대 순차 빈발 문서 집합 개수만큼 반복  
 최대 순차 빈발 문서 집합 길이만큼 반복  
 두 문서의 최단 경로 길이가  $\gamma$  이상이면  
 두 문서를 클러스터에 추가
  2. 클러스터 개수 만큼 반복  
 클러스터 길이-1 만큼 반복  
 (두문서 비교) 전문서가 첫번째 열에 없다면  
 행 방향으로 추가해 전문서 기입  
 추가된 행의 열 방향으로 후문서 기입  
 있다면  
 전문서가 있는 행의 열 방향으로 후문서 추가

1. 알고리즘 적용 결과
- 클러스터 1 <BH>
  - 클러스터 2 <DH>
  - 클러스터 3 <DG>

2. 알고리즘 적용 결과
- B  
H

알고리즘 1을 통해 세 개의 웹 문서 클러스터인 <BH>, <DH>, <DG>를 생성한다. 다음 단계인 알고리즘 2에서는 첫 번째 열 방향이 색인 페이지 생성 위치 문서들이고 두 번째 열 방향 부터는 행 단위로 해서 색인 페이지를 구성하는 웹 문서 클러스터가 된다. 따라서 B 문서의 위치에 H를 구성요소로 한 웹 문서 클러스터가 생성되고 D 문서의 위치에 H,G를 구성요소로한 웹 문서 클러스터 생성된다.

### 6. 색인 페이지 생성

색인 페이지는 기존 웹 상에서 하이퍼 링크로 연결 되어 있지는 않지만 사용자 접근 패턴을 분석한 결과를 토대로 관련성이 있다고 추정되는 페이지들을 그 구성요소로 한다. 색인 페이지의 구성 요소인 클러스터를 생성하기 위해 관련성 있는 문서들의 집합을 사용자 순차 접근 패턴을 통해 알아낸 후, 웹 사이트의 하이퍼 링크 구조 정보를 이용하는 과정을 거쳐 접근 경로가 긴 문서들만을 구성요소로 한 클러스터를 생성시킨다. 색인 페이지 생성 과정에서 고려해야 할 문제로, 색인 페이지의 구성요소가 되는 문서를 정렬하는 방법과 색인 페이지의 제목을 정하는 방법, 색인 페이지를 기존 사이트의 디자인과 일치시키는 방법 등이 있다. 색인 페이지 제목과 구성 문서들의 정렬 방법은 문서간 내용 유사성을 측정된 결과로 얻을 수 있다. 색인 페이지의 제목인 경우 클러스터로 묶인 문서들을 대상으로 자주 나타나는 공통 단어를 추출해 이를 제목으로 사용할 수 있고, 구성 문서들의 정렬에서는 생성 시키는 위치의 문서의 내용과 유사성 정도를 측정해 높은 유사성을 갖는 문서 순으로 정렬할 수가 있다. 이와 같은 문서간 내용 유사성을 측정하기 위한 가장 간단한 방법은 문서를 대표할 만한 특정 단어를 추출해 단어의 출현 여부, 혹은 빈도수를 벡터로 정형화해 비교하는 방법이다. 만약, TFIDF 알고리즘을 적용한다면 문서에 나타나는 단어들의 중요도에 따라 가중치가 부가됨으로써 문서간 좀 더 정확한 내용 비교가 가능하다.

### 7. 결론

본 논문에서는 웹 사이트에 접근하는 사용자 접근 패턴을 학습하여 사이트의 구조나 외형을 자동 개선시켜 나가는 적응형 웹 사이트를 구축 방안으로 색인 페이지의 자동 생성을 제시했다. 생성 과정에서는 대용량의 웹 서버 로그 데이터들을 대상으로 하여 AprioriAll 알고리즘을 적용함으로써 웹 문서간의 충분한 순차 연관성 지도도를 갖는 사용자 순차 접근 패턴을 분석한다. 분석 결과 구해낸 관련성 있는 문서들의 집합을 바탕으로 웹 사이트의 하이퍼 링크 구조 정보를 추가로 이용하는 과정을 거쳐 접근 경로가 긴 문서들만을 구성요소로 한 클러스터들을 생성시켰다. 이와 같이, 클러스터를 이용한 색인 페이지를 생성함으로써 보다 효과적인 사용자들의 정보 접근을 지원할 것이다.

#### 참고문헌

- [1] R. Agrawal and R. Srikant, Fast Algorithms for Mining Association Rules, In *Proc. of the 20<sup>th</sup> VLDB Conference*, pp.487-499, 1994.
- [2] M. S. Chen, J. S. Park, and P. S. Yu, Data Mining for Path Traversal Patterns in a Web Environment, In *Proc. 16<sup>th</sup> Int. Conf. Distributed Computing Systems*, pp. 385-392, 1996.
- [3] D.W. Chung, Ben Kao, and Joseph Lee, Discovering User Access Patterns on the World-Wide Web, In *Proc. of PAKDD-97*, 1997.
- [4] R. Cooley, B. Mobasher, and J. Srivastava, Web Mining: Information and Pattern Discovery on the World Wide Web, In *Proc. of TAI-97*, 1997.
- [5] Fernandez M. Florescu D., Kang J., Levy A., and Suciu D., System Demonstration –Strudel: A Web-site Management System, In *ACM SIGMOD Conference on Management of Data*, 1997.
- [6] Joachims, T.; Freitag, D. and Mitchell, T.. Webwatcher : A tour guide for the World Wide Web. In *Proc. IJCAI-97*, 1997.
- [7] Osmar R. Zaiane, Man Xin, and Jiawei Han, Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs, In *Proc. of ADL-98*, 1998.
- [8] Perkowski M. and Etzioni O., Adaptive Web Sites: an AI Challenge, In *Proc 15<sup>th</sup> Int. Joint Conf. AI*, 1997.
- [9] P. Pirolli, J. Pitkow, and R. Rao, Silk from a Sow's Ear: Extracting Usable Structures from the Web, In *Proc. of CHI-96*, pp.118-125, 1996.
- [10] Fink J., Kobsa A. and Nill A., User-oriented Adaptivity and Adaptivity in the AVANTI Project, In *Designing for the Web: Empirical Studies*
- [11] M. Fernandez, D. Florescu, A. Levy, and D. Suciu. Reasoning about Web-site structure, 1998.
- [12] Srikant, R. & Agrawal, R. (1996). Mining sequential patterns.

Research Report RJ 9910, IBM Almaden Research Center, San Jose, California, October 1994.

- [13] Srikant, R. & Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements. In *EDBT*, Avignon, France, March 1996.
- [14] Park, J. S., Chen, M-S, and Yu, P. S. *An Effective Hash Based Algorithm for Mining Association Rules*, Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD-95), 1995.