웹 구조 마이닝에 초점을 둔 웹 마이닝의 조사†

이석민○ 박대명 유대훈 최웅철
광운대학교 컴퓨터 과학과
{nenunena, eoanddl, yo2dh, wchoi}@kw.ac.kr

A Survey of Web Mining Focused on Web Structure Mining

SeokMin Lee○, DaeMyeong Park, Daehun Yoo, WoongChul Choi
Department of Computer Science, KwangWoon University

요 약

컴퓨터 기술의 발달 및 웹의 확산으로 인해 개인이 얻을 수 있는 정보의 양이 증가되었지만, 이로 인해 필요한 관련 정보를 탐색하는 것과 다량의 정보로부터 지식을 창출한다는 것이 어렵게 되었고, 고객 또는 사용자에 대한 학습 과정 및 정보의 개인화 등의 문제가 대두되게 되었다. 이러한 문제들을 해소하기 위해 웹으로부터 정보를 얻을 수 있는 자동화된 툴이 필요하게 되었고, 얻은 정보를 이용하여 웹 사용자들의 패턴을 식별할 수 있는 방법 또한 필요하게 되었다. 이러한 관심은 데이터 마이닝을 온라인에서 적용하고자 하는 노력으로 이어졌고, 현재 데이터 마이닝 기술을 온라인에 적용한 웹 마이닝 기술을 사용하고 있다. 웹 마이닝은 웹의 방대한 양의 자료 및 구조를 좀 더 유용하고, 효율적인 정보로 가공하여 사용자에게 제공할 수 있도록 도와주는 기술이다. 본 논문에서는 웹 마이닝의 전반적인 개념과 분류를 소개한다. 또한, 웹 마이닝의 분류 중 웹 구조 마이닝에 초점을 맞추어 개념 및 웹 구조 마이닝의 대표적인 알고리듬을 소개한다.

1. 서 론

정보 통신기술의 발달, 데이터의 생성과 저장 능력에 대한 증가 그리고 인터넷 이용자의 확산에 따라 고객 관련 데이터는 급속히 증가하고 있으며 웹 서버에 쌓이는 데이터(log data)의 양 또한 하루에도 수 기가 바이트에 이르고 있다. 이러한 증가는 데이터를 자동적이고 지능적인 방법을 이용하여 유용한 정보나 지식으로 변환해 주는 새로운 기법과 툴들에 대한 개발을 재촉해 왔으며, 이와 같은 환경 하에서 데이터마이닝의 중요성을 점차 더해 가고 있다[1].

웹 사이트의 관리자나 경영자는 어떠한 접속자들이 접속을 하는지, 접속자들의 탐색 형태는 어떤 특성이 있는지 등과 같이 접속자의 접속 패턴과 특성을 이용하여 보다 효율적이고 개선된 웹 사이트 관리와 고객에 대한 서비스를 제공하고자 하는 것에 많은 관심을 갖게 되었으며, 이러한 관심은 데이터마이닝을 온라인에서 적용하고자 하는 노력으로 이어졌다.

웹 서버에는 접속자의 웹 서비스 요청과 이에 대한 응답 기록이 로그파일(log file)이라는 형태로 저장된다. 웹 마이닝이란 웹 서버에 저장되는 로그파일 등과 같이 온라인상에서 기록되고 저장되는 데이터 속에서 의미 있고 유용한 정보를 발견하고 분석하는 일련의 프로세스를 의미한다[2][3]. 로그파†일을 분석하게 되면 페이지별 방문 수 및 체류시간, 접속자들이 가장 선호하는 페이지나 가장 적게 방문 하는 페이지 등의 접속 정보와 서버 응답률, 오류 발생률 등의 기술적인 정보를 얻을 수 있다. 이러한 정보들은 웹 사이트 관리에 있어 유용한 정보임에 틀림없다. 그러나 보다 의미 있고 가치 있는 정보를 발견하기 위해서는 로그 데이터와 함께 고객 데이터 및 웹 사이트의 컨텐츠와 관련한 웹 데이터의 분석이 함께 이루어 져야만 한다. 또한, 웹 사이트의 구조나 웹 페이지 간 링크 구조를 이용한 분석을 통해 정보를 추출하면, 보다 의미 있고 가치 있는 정보를 발견하는데 도움이 된다. 본 논문에서는 웹 사이트의 구조나 웹 페이지 간 링크 구조를 이용하여 유용한 정보를 추출하는 웹 구조 마이닝에 초점을 맞추었다.

본 논문의 구성은 다음과 같다. 2장에서 웹 마이닝과 웹 구조 마이닝에 대해서 살펴본다. 3장에서는 웹 구조 마이닝의 관련 알고리듬을 살펴보는 것에 이어서4장에서 결론을 내린다.

2. 웹 마이닝과 웹 구조 마이닝

'웹 마이닝'은 트래픽, 등록과 거래 정보, 사용 패턴 등 인터넷에서 벌어지는 모든 웹 데이터를 전통적인 데이터 마이닝 기법에 접목시킨 기술이다. 즉, 웹 마이닝이란 웹 서버 로그(web server log)로부터 웹 사용자의 의미 있는 접속 패턴을 발견하는 과정이라고 할 수 있다. 웹 서버는 웹 사이트 사용자의 행동 패턴에 대한 정보를 웹 서버 로그 파일에 저장하는데 여기에는IP 주소, HTTP 요청 페이지, 요청에 응답한 시간 등 사용자에 관한 귀중한 정보가 들어 있다. 이에 따라 웹 사이트 운영자는 웹 서버 로그 파일을 분석함으로써 사용자의 관심이 어느 곳에 있는지, 자신의 사이트가 사용자의 요구를 잘 반영하고 있는지를 파악할 수 있고, 이를 통해 웹 사이트의 디자인과 성능 등을 개선할 수 있을 뿐 아니라 효과적인 마케팅 전략을 세울 수도 있다. 또한 고객 개개인을 분석하여 고객의 가치를 측정할 수 있고 교차판매를 유도할 수 있으며 캠페인을 효과적으로 수행하기 위

† 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2009-0090108).

한 정보를 얻을 수도 있다.

웹 마이닝의 목적 중 하나는 개인화된 사용 추적으로 개별적인 웹 사용자의 행동 패턴과 성향을 파악하고 이를 통해 웹 사이트를 사용자에 맞게 개인화시키는 것이다. 즉 웹 사용자의 행동 패턴에 근거해서 웹 사이트에 제시되는 정보, 웹 사이트의 구조 등을 개인화하는 것이다. 웹 사이트를 개인화 하려면 웹 서버 로그뿐만 아니라 고객정보, 거래정보를 통합하여 분석해야 하는데 이를 통해서 제품 추천(recommendation)이나 개인화된 광고 등을 제공할 수 있다. 이러한 개인화된 웹 사이트를 적응형 사이트(adaptive site)라고 부르는데, 웹 사용자의 행동 패턴을 학습하고, 행동 패턴에 맞게 스스로 자신을 향상시키는 웹 사이트를 말한다.

2.1 웹 마이닝 분류

웹 마이닝은 크게 웹 컨텐트 마이닝(web content mining)과 웹 이용 마이닝(web usage mining)으로 분류하며, 분석 소스의 유형에 따라 컨텐트 마이닝(content mining)과 구조 마이닝(structure mining)으로 나뉜다[2][3]. 이를 종합해서 웹 마이닝을 재분류하면 그림1과 같다. 다만 각각의 분류에 있어 그 경계가 뚜 렷한 것은 아니며, 서로 혼합하여 사용하는 경우가 많다[4][5].

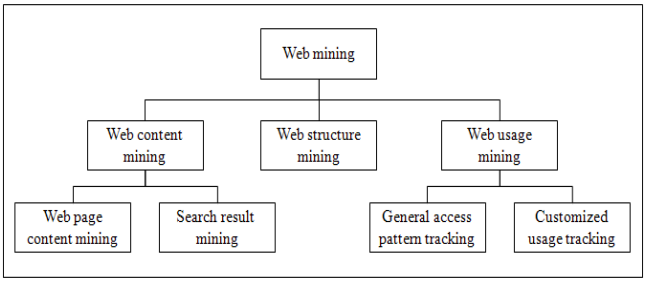

그림 1. 웹 마이닝의 구조

■ 웹 컨텐트 마이닝: 웹에 존재하는 컨텐트를 분석의 대상으로 유용 한 정보나 지식을 추출· 발견하는 프로세스. 텍스트 마이닝(text mining)혹은 텍스트 데이터마이닝(text date mining)이라고 한다.

■ 웹 이용 마이닝: 웹 서버에 저장된 로그 데이터를 이용하여 웹 서버접속자의 접속 패턴을 발견하는 프로세스 이다.

■ 웹 구조 마이닝: 웹 페이지 내 혹은 웹 페이지 간의 하이퍼링크, 그리고 URL을 통해 이들 간의 구조나 링크에 관련한 유용한 지식을 발견하는 프로세스 이다.

표 1. 데이터 유형 및 사용 기법

	웹 컨텐트 마이닝	웹 구조 마이닝	웹 사용 마이닝
데이터 유형	텍스트, 오디오, 비디오 등	웹 문서, 하이퍼링크	사용자 프로파일, 접근 패턴 등
사용 기법	컨텐트 기반 필터, 명성 기반 필터	명성 기반 필터	이벤트 기반 필터

2.2 웹 구조 마이닝

웹 구조 마이닝은 웹 사이트와 웹 페이지의 구조적 요약 정보를 얻는 것을 목표로 한다. 웹 사이트의 구조적 정보란, 웹 페이지 사이의 하이퍼링크(hyperlink)를 통한 그래프(graph) 구조를 뜻한다. 참조(reference) 정 보를 이용하는 경우의 예로서 다음과 같은 표준 로그를 살펴보자.

■ 211.104.136.123 - - [17/Apr/2001:12:00:12 +0900] "GET /index.html HTTP/1.1"

■ 200 16674 "/products/tv.html""Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)"

이 로그를 통해 웹 사이트는 /index.html 에서 /products/tv.html 로의 웹 구조를 추출할 수 있다. 이와 같은 방법은 사이트 내에 페이지가 많거나, 여러 사이트를 통합해 운영하는 대규모 웹 사이트, 또는 페이지를 자주 업데이트하는 사이트에서 구조 정보를 얻을 때나 사이트 관리 등에 응용할 수 있다. 예를 들어, 어떤 페이지는 홈페이지에서 자신을 참조하는 경로가 없을 수 있는데, 이런 페이지는 웹 사이트 사용자가 접근할 수 없는 페이지로 삭제하거나 적당한 링크를 통해 접근할 수 있게 해야 한다.

3. 웹 구조 마이닝의 관련 알고리듬

본 절에서는 웹 구조 마이닝에 사용되는 알고리듬을 소개한다. 웹 구조 마이닝 사용되는 주요 알고리듬은 페이지랭크(PageRank), HITS(Hyperlink-Induced Topic Search), imp 등 이 있다.

3.1 페이지랭크(PageRank) 알고리듬

페이지랭크 알고리듬은 월드 와이드 웹과 같은 하이퍼링크 구조를 가지는 문서에 상대적 중요도에 따라 가중치를 부여하는 방법이다. 이 알고리듬은 서로 간에 인용과 참조로 연결된 임의의 묶음에 적용할 수 있다. 페이지랭크 알고리듬은 "더 중요한 페이지는 더 많은 다른 사이트로부터 링크를 받는다"는 관찰에 기초하고 있다. 예를 들어 페이지 A가 페이지 B,C,D 로 총 3개의 링크를 걸었다면 B는 A의 페이지랭크 값의 1/3 만큼을 가져온다. 또한, 또한 페이지랭크에서는 랜덤 서퍼(Random Sufer)라는 페이지를 임의로 방문하며 탐색하는 모델을 가정한다. 이 모델에서는 위 예의 A페이지를 방문한 서퍼는 A페이지를 보고 만족하여 탐색을 중단하거나, 혹은 A페이지에서 만족하지 못하여 다른 페이지를 방문할 것이다. 이러한 확률(Damping Factor)을 α라 한다면, B페이지는 α * 1/3 만큼 페이지 랭크를 받게 된다. 이와 같은 방법을 통해 페이지 간 페이지랭크 값을 주고 받는 것을 반복하다보면, 전체 웹 페이지가 특정한 페이지랭

크 값에 수렴한다는 사실을 통해 각 페이지의 최종 페이지랭크를 계산한다[6].

페이지랭크 알고리듬은 다음과 같다.

$$r(P_i) = \sum_{P_j \in B_{P_i}} \frac{r(P_j)}{|P_j|} \quad \cdots\cdots(1)$$

$r(P_i)$: 페이지 Pi 를 가르키는 모든 페이지의 페이지랭크 값의 합

B_{P_i} : Pi를 가르키는 페이지의 집합

$|P_j|$: Pj로부터의 Outlink의 개수

$r(P_j)$: 페이지 Pj를 가르키는 모든 페이지의 페이지랭크 값의 합

다음과 같은 페이지랭크 알고리듬은 최종 페이지랭크 값을 구하기 위해서 페이지 간 페이지랭크 값을 주고 받는 것을 반복한다. 다음 알고리듬은 알고리듬 (1)을 반복적으로 표현한 것 이다.

$$r_{k+1}(P_i) = \sum_{P_j \in B_{P_i}} \frac{r_k(P_j)}{|P_j|} \quad \cdots\cdots(2)$$

$r_{k+1}(P_i)$: Pi의 K+1 번째 반복

페이지랭크 값을 계산하는 예를 보면 다음과 같다. WWW(World Wide Web)의 웹 페이지가 총 6개 이고, 그림 2와 같은 링크 관계를 갖는다고 가정한다. 또한, 모든 페이지의 초기 페이지랭크 값은 1/n 이라고 가정한다. n 은 총 페이지의 개수 이다. 알고리듬 (2)를 이용하여 2번 반복 했을 때의 페이지랭크 값 및 순위를 구해보면 그림 3과 같다.

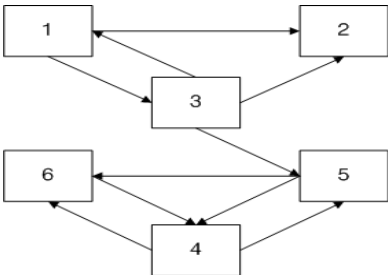

그림 2. 웹 페이지 구조

Iteration 0	Iteration 1	Iteration 2	Rank at Iter 2.
$r_0(P_1) = 1/6$	$r_1(P_1) = 1/18$	$r_2(P_1) = 1/36$	5
$r_0(P_2) = 1/6$	$r_1(P_2) = 5/36$	$r_2(P_2) = 1/18$	4
$r_0(P_3) = 1/6$	$r_1(P_3) = 1/12$	$r_2(P_3) = 1/36$	5
$r_0(P_4) = 1/6$	$r_1(P_4) = 1/4$	$r_2(P_4) = 17/72$	1
$r_0(P_5) = 1/6$	$r_1(P_5) = 5/36$	$r_2(P_5) = 11/72$	3
$r_0(P_6) = 1/6$	$r_1(P_6) = 1/6$	$r_2(P_6) = 14/72$	2

그림 3. 반복적인 페이지랭크 알고리듬을 적용 결과

3.2 HITS 알고리듬

HITS 알고리듬은 크게 2단계로 나뉜다. 첫 번째 단계에서는 질의어와 관계있는 페이지들의 부분집합서브 그래프(subgraph))을 말들고, 두 번째 단계에서는 만들어진 서브 그래프를 이용해서 협과 오쏘리티(hubs & authorities)를 계산한다[7].

3.2.1 서브 그래프(subgraph) 만들기

먼저, 하이퍼링크로 연결된 페이지들의 컬렉션V를 G = (V,E)라는 directed graph로 표현한다. 이때 각각의 노드는 각 페이지에 해당하고 (p,q) ∈ E는 p에서 q로의 링크가 있다는 것을 뜻한다.

- 노드 p의 out-degree: 노드 p에서 밖으로 나가는 링크의 개수
- 노드 p의 in-degree : 노드 p를 가리키는 링크의 개수
- G[W] : V에 속하는 부분집합 W로부터 만든 그래프
- σ : 질의어

질의어 σ를 담고 있는 페이지 전체 집합을 Qσ라고 가정 했을 때. 알고리듬은 이 Qσ를 대상으로 하면 안 된다. 그 이유는 첫째, 질의어를 담고 있는 페이지는 아마도 수백만 페이지 이상일 것이기 때문에 'computationally expensive'하다. 둘째, 앞에서 살펴 본 대로 중요한 오쏘리티는 질의어 자체를 담고 있지 않은 경우도 매우 많다.

사용자가 원하는 집합을 Sσ라 하면, 알고리듬의 첫 단계인 서브 그래프를 구하는 것은 이러한 집합Sσ를 구하기 위한 것 이다. Sσ는 다음과 같은 특징을 가져야 한다.

1. 상대적으로 작아야 한다.
2. 관계되는 페이지가 많아야 한다.
3. 대부분의 오쏘리티들을 담고 있어야 한다.

일반적인 텍스트 기반 검색엔진에 질의어를 넣었을 때의 결과 중 상위 t개의 페이지를 루트셋(root set) Rσ라 가정한다. 이 Rσ는 아마도 위의 조건 중 1,2번은 만족하지만 3번은 만족하지 못 할 가능성이 높다. 그러므로 Rσ를 이용해서 강한 오쏘리티(strong authorities)를 찾아낼 수 있다면 우리가 원하는 Sσ에 가까운 서브 그래프를 만들어 낼 수 있다.

오쏘리티는 Rσ에 있는 페이지들이 가리키는 페이지들 중에 많이 존재할 가능성이 높다. 관계되는 페이지들이 가리키고 있는 페이지는 '권위 있는' 페이지일 가능성이 크기 때문이다. 따라서 Rσ를 링크를 이용해서 확장하면 우리가 원하는 강력한 오쏘리티가 들어있으면서 작고, 관계성 높은 그림 4와 같은 서브 그래프를 만들어 낼 수 있다.

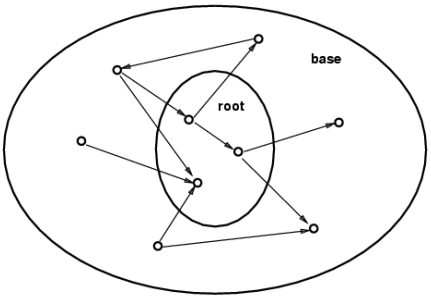

그림 4. 루트 셋을 확장한 그래프

3.2.2 협과 오쏘리티 계산

앞 단계에서 만든 서브그래프를 이용하여 협(hubs)과 오쏘리티(authorities)를 찾는다. 일단 Sσ를 그래프화 한 것을 Gσ라 가정한다. 이제 링크 구조를 이용해서 이Gσ 속에 존재하는 협과 오쏘리티를 찾아내는데 제일 쉽게 생각해 볼 수 있는 것이 Gσ내의 페이지들을 in-degree 순으로 순위를 매기는 것이다. 이것은 직관적으로 좋은 방법일 것 같은 느낌이 든다. 일단 Gσ라는 특정 검색어와 관계성이 높은 페이지들 집합 속에서 다른 페이지로부터의 링크가 많다는 것은 그 만큼 '좋은' 페이지일 가능성이 높아지기 때문이다. 하지만 실제로 그렇게 해보면 강력한 오쏘리티와 보편적으로 인기가 높은 페이지(universally popular pages) 사이의 긴장이라는 문제가 생긴다. 예를 들어 "java"라는 질의어의 경우 www.gamelan.com과 java.sun.com이라는 결과와 함께 카리비안 휴양 관련 싸이트와 아마존 홈페이지가 함께 순위권에 든다. 이것은 특정 주제에 관한 오쏘리티와 함께 '주제에 관계없이 링크가 많이 된(universally popular) 것이 존재하기 때문이다. 이 문제의 해결책은 페이지의 텍스트 내용도 같이 감안해서 검색어와 매칭되는 단어가 등장하는 것을 골라내는 정도가 될 것 이다. 이것도 실제로 적용해 보면 별 효과가 없다.

어떤 주제에 관한 오쏘리티들을 많이 링크하고 있는 페이지를 중심축 역할을 한다는 의미에서 협이라고 명명한다. 그러면 "모든 오쏘리티들은 큰 in-degree를 갖는다는 점과 함께 그 오쏘리티들을 가르키고 있는 페이지들 역시 중복된다는 공통점을 갖는다" 라는 것은, "좋은 오쏘리티들은 in-degree가 높다는 점과 함께 많은 협들로 부터 링크되어 있다는 공통점이 있다"는 얘기가 된다. 여러 협들로부터 링크되어 있을수록 좋은 오쏘리티가 되며, 여러 오쏘리티를 링크하고 있을수록 좋은 협이 되는 것이다. 즉, 협과 오쏘리티는 상호강화적인 관계(mutually reinforcing relationship)이다.

그러므로 좋은 오쏘리티는 좋은 협을 찾는 것을 통해서 찾아낼 수 있고, 좋은 협은 좋은 오쏘리티를 통해서 찾아낼 수 있다. 그리고 이들 협과 오쏘리티는 광범위적 질의어의 검색 결과 중에서도 특별히'좋은' 페이지로 생각해 볼 수 있는 것이다. 다음 그림 5는 협과 오쏘리티의 관계의 2가지 예를 보여준다.

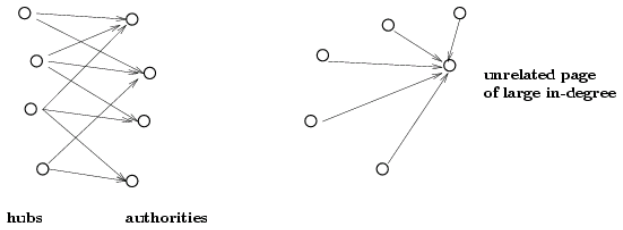

그림 5. 협과 오쏘리티와의 관계

위에서 살펴 본 내용을 수식을 이용하여 기술하면 다음과 같다.

"authority weight", 즉 오쏘리티 가중치를(높을수록 좋은 오쏘리티) x(p)라 하고, "hub weight", 협 가중치를(높을수록 좋은 협) y(p)라 하면 다음과 같은 두 가지의 수식을 생각해 볼 수 있다.

$$x^{<P>} \leftarrow \sum_{q:(q,p)\in E} y^{<q>} \cdots\cdots(3)$$

$$y^{<P>} \leftarrow \sum_{q:(q,p)\in E} x^{<q>} \cdots\cdots(4)$$

(3) : q 에서 p로의 링크가 있을 때 p 페이지의 오쏘리티 가중치는 q 페이지들의 협 가중치 y(p)를 모두 합한 것이다.

(4) : p 에서 q로의 링크가 있을 때 p 페이지의 협 가중치는 q 페이지의 오쏘리티 가중치 x(q)를 모두 합한 것이다.

그런데 이것은 반복적 알고리듬(iterative algorithm)이기 때문에 이터레이션을 시켜서 계산해야 한다. 그렇게 반복 연산을 했을 때 어떤 일정한 값으로 수렴하면서 평형을 이루는지 알아보는 것이다. 그리고 수렴해서 일정한 값들을 갖는다면, 우리는 각 페이지들의 최종 협 가중치와 오쏘리티 가중치를 알게 될 것이다. 어떤 페이지가 얼마나 좋은 협인지, 얼마나 좋은 오쏘리티인지를 알 수 있게 되는 것이다. x(p)의 집합 {x(p)}를 벡터 x라 하고, y(p)의 집합 {y(p)}를 벡터 y라 가정하고, 반복적인 알고리듬을 적용한다[7]. 그 결과는 x, y의 벡터 값으로 나타나고, 그 결과 값의 나열로 상위 오쏘리티 및 협 값을 갖는 페이지를 결정할 수 있다.

3.3 imp 알고리듬

imp 는 HITS 를 개선한 알고리듬 이다. 3.2 절에서 언급한 수학적 표현식 (3), (4)에서 각각에 오쏘리티 가중치와 협 가중치를 곱한 것 이다. 이를 수식으로 표현하면 다음과 같다[8].

$$x^{<P>} \leftarrow \sum_{q:(q,p)\in E} y^{<q>} \times authority\_weight(q,p) \cdots\cdots(3)$$

$$y^{<P>} \leftarrow \sum_{q:(q,p)\in E} x^{<q>} \times hub\_weight(p,q) \cdots\cdots(4)$$

각각에 오쏘리티 가중치 및 협 가중치를 곱한 이유는 다음과 같다. 좋은 오쏘리티를 갖는 페이지는 좋은 협들로부터 링크 되어있다. 하지만, 하나 또는 몇몇의 좋은

협에 의해서 좋은 오쏘리티 페이지 결정에 큰 영향을 미칠 수 있다. 이러한 편향된 결과의 발생을 방지하기 위해 하나의 협 페이지가 가르키는 링크의 수로 협 가중치를 정한다. 예를 들어 10개의 링크가 있다고 한다면 협 가중치는 1/10 이다. 이와 마찬가지로 오쏘리티 가중치도 정해지게 된다. 이 부분 외의 알고리듬은 HITS 알고리듬과 동일하다.

4. 결론

정보 통신기술의 발달, 데이터의 생성과 저장 능력에 대한 증가 그리고 인터넷 이용자의 확산에 따라 고객 관련 데이터는 급속히 증가하고 있으며 웹 서버에 쌓이는 데이터(log data)의 양 또한 하루에도 수 기가 바이트에 이르고 있다[1]. 웹의 구조 또한 복잡하게 얽히게 되었고, 서로의 연관 관계가 더욱 긴밀해 지고 있는 상태이다.

현재, 우리나라 뿐만 아니라 세계적으로, 인터넷 사용자 행태를 종합분석해서 마케팅 정보로 가공해주는'웹 마이닝' 시장이 정보통신산업에서 급부상하고 있다.이러한 웹 마이닝 기술을 이용하여 향후에는 어떤 접속자들이 접속을 하는지, 접속자들의 탐색 형태는 어떤 특성이 있는지, 웹 사이트의 구조는 어떠한지, 웹 페이지들은 어떤 링크 구조를 갖는지 등과 같은 접속자의 접속 패턴과 특성 및 웹 사이트의 구조와 링크 정보를 추출할 수 있어야한다. 이는 보다 효율적이고 개선된 웹 사이트 관리와 고객에 대한 서비스를 제공하는데 큰 도움을 줄 것이다.

참 고 문 헌

[1] Ming-Syan Chen, et al., "Data Mining- An Overview from a Database Perspective", IEEE Transactions on knowledge and data engineering, vol. 8, No. 6, December, 1996

[2] R. Cooley et al, "Web Mining: Information and Pattern Discovery on the World Wide Web", IEEE, 1997, p.558

[3] R. Cooley, et al., "Grouping Web Page References into Transactions for Mining World Wide Web Browsing Patterns", IEEE, 1997

[4] Wang Jicheng, et al., 1999, "Web Mining: Knowledge Discovery on the Web", IEEE, 1999, p11- 137

[5] Raymond Kosala, et al., "Web Mining Research: A Survey", ACM SIGKDD, Volume2, 2000, p1-3

[6] Haveliwala, T. Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search, IEEE TKDE.15(4), pp.784-796, 2003

[7] J. Kleinberg. Authoritative sources in a hyperlinked environment. In Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms, pp.668-677, January 1998

[8] K. Bharat and M. Henzinger. Improved algorithms for topic distillation in hyperlinked environments. In proceedings of the 21st International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR'98), pages 104 111, 1998