WebPR : 빈발 순회패턴 탐사에 기반한 동적 웹페이지 추천 알고리즘

윤 선 희† · 김 삼 근†† · 이 창 훈††

요 약

월드 와이드 웹(World-Wide Web)은 가장 커다란 분산된 정보저장소로서 계속하여 빠른 속도로 성장해왔다. 그러나 비록 웹이 빠른 속도로 성장하고 있다 할지라도, 웹의 정보를 읽고 이해하는 데는 본질적으로 한계가 있다. 웹 사용자 입장에서 보면 웹의 정보 폭발, 꾸준하게 변화하는 환경, 사용자 요구에 대한 이해 부족 등으로 오히려 혼란을 겪을 수 있다. 웹의 이러한 환경에서 사용자의 순회패턴(traversal patterns)을 탐사하는 것은 시스템 설계나 정보서비스 제공 측면에서 중요한 문제이다. 순회패턴 탐사에 관한 기존의 연구들은 세션(sessions)에 나타나는 페이지들간의 연관성 정보를 충분히 활용하지 못한다. 본 논문에서는 세션에 나타나는 페이지들간의 연관성 정보를 활용하여 빈발 k-페이지집합을 탐사하고, 이를 기반으로 하여 추천 페이지집합을 생성함으로써 효율적인 웹 정보서비스를 제공할 수 있는 Web Page Recommend(WebPR) 알고리즘들을 제안한다. 제안한 WebPR 알고리즘은 웹 사이트를 방문한 사용자에게 추천 페이지집합을 포함하는 새로운 페이지뷰(pageview)를 제공함으로써 궁극적으로 찾고자하는 목표 페이지에 효과적으로 접근할 수 있도록 해준다. 기존 연구들과의 가장 큰 차이점은 페이지들간의 연관성 정보를 활용하는 방법들을 일관성있게 고려하고 있다는 점과 가장 효율적인 트리모델을 제안한다는 점이다. 두개의 실제 웹로그(Weblog) 데이터에 대한 실험은 제안한 방법이 기존의 방법들보다 성능이 우수함을 보여준다.

WebPR : A Dynamic Web Page Recommendation Algorithm Based on Mining Frequent Traversal Patterns

Sun-Hee Yoon† · Samkeun Kim†† · Changhoon Lee††

ABSTRACT

The World-Wide Web is the largest distributed information space and has grown to encompass diverse information resources. However, although Web is growing exponentially, the individual's capacity to read and digest contents is essentially fixed. From the view point of Web users, they can be confused by explosion of Web information, by constantly changing Web environments, and by lack of understanding needs of Web users. In these Web environments, mining traversal patterns is an important problem in Web mining with a host of application domains including system design and information services. Conventional traversal pattern mining systems use the *inter-pages association* in sessions with only a very restricted mechanism (based on vector or matrix) for generating frequent k-Pagesets. We develop a family of novel algorithms (termed WebPR - Web Page Recommend) for mining frequent traversal patterns and then pageset to recommend. Our algorithms provide Web users with new page views, which include pagesets to recommend, so that users can effectively traverse its Web site. The main distinguishing factors are both a point consistently spanning schemes applying *inter-pages association* for mining frequent traversal patterns and a point proposing the most efficient tree model. Our experimentation with two real data sets, including LadyAsiana and KBS media server site, clearly validates that our method outperforms conventional methods.

키워드 : 순회패턴, 빈발 k-페이지 집합, 페이지간 연관관계, 웹추천

1. 서 론

월드 와이드 웹(World-Wide Web)은 뉴스, 광고, 소비자 정보, 재정관리, 교육, 정부, 전자상거래 등의 많은 정보 서비스를 위해 거대하고 널리 분산된 정보 서비스 센터로서의 역할을 한다[1, 2]. 또한 풍부하고 동적인 하이퍼링크(hyperlink) 정보와 웹 페이지 접근 정보 등을 포함하고 있어서 데이터 탐사(data mining)를 수행하는데 요구되는 풍부한 자원을 제공해준다. 그러나 웹 정보의 극히 일부만이 서로 강하게 연관되어 있거나 유용하다.

전형적인 웹 사이트들은 수천 혹은 수백만 명에 의한 페이지 접근정보 시퀀스를 포착할 수 있는 거대한 로그 데이

† 정 회 원 : 미림 전산고등학교 교사
†† 종신회원 : 한경대학교 컴퓨터공학과 교수
논문접수 : 2003년 11월 5일, 심사완료 : 2004년 3월 30일

터를 생성한다. 웹 환경에서 사용자의 접근패턴을 포착하는 것을 순회패턴 탐사(mining traversal patterns)라 한다. 웹 서버로그에서 빈발 순회패턴 탐사는 웹 사이트의 설계에 대한 결정을 도와주거나[3, 4], 적응적인 웹 사이트를 가능하게 하거나[5], 마케팅 결정을 지원하거나 [6], 사용자 인터페이스 테스팅, 보안 목적을 위한 감시 등을 들 수 있다. 특히 더 중요한 응용분야로는 추천시스템[7]이나 목표시장 광고(target advertizing) 등과 같은 웹 개인화 분야가 있다.

한편, 빈발 순차패턴 탐사는 수많은 잠재적인 응용분야에 적용될 수 있다 : 소매점(retailing) (즉, market-basket 데이터), 통신(telecommunications), 의학(medicine), 웹(Web) 등. Market-basket 데이터베이스에서 각 데이터 시퀀스는 일정 기간 동안 개별 고객에 의해 구매된(시간에 따라 순서를 가지는) 항목집합을 말한다. 이때 빈발하게 발생하는 패턴들은 고객의 행위를 예측(predict)하는데 매우 유용하다.

최근, 빈발 순차패턴(혹은 순회패턴) 탐사를 위한 효과적인 알고리즘들이 제안되었다[5, 8]. 이들 알고리즘들은 다양한 휴리스틱(heuristic) 혹은 사이트 정보 등을 이용하여 보다 효과적인 빈발 순회패턴을 탐사하고자 한다. 그러나 이들 방법은 빈발 순회패턴을 탐사함에 있어 순회패턴에 포함되어 있는 정보를 충분히 활용하지 못한다. 예를 들어, [8]에서 제안한 알고리즘은 순회패턴에 포함된 특징을 벡터로 표현하여 빈발 순회패턴을 탐사함으로써 탐사과정에 유용할 수 있는 순회패턴의 순서 특징을 반영하지 못한다. 또한 [5]의 방법은 순회패턴의 특징을 유사도 행렬(similarity matrix)로 표현하고 있다. 그들이 사용한 유사도 행렬은 순회패턴에서 인접한 페이지들간의 특징은 탐사과정에 반영되지만 인접하지 않은 페이지들간의 순서 특징은 표현하지 못한다.

본 논문에서는 효율적으로 빈발 페이지집합을 생성하는 여러 가지 모델들을 제안하고, 이를 웹 사이트 순회에 적용함으로써 보다 개선된 웹 정보 서비스를 제공하는 새로운 Web Page Recommend(WebPR) 알고리즘을 제안한다. WebPR 알고리즘은 생성된 빈발 페이지집합을 이용하여 사용자가 웹 사이트를 순회하는 동안 각 단계에서 추천 페이지집합 R을 사용자의 현재 페이지뷰(pageview)에 포함시킨 새로운 페이지뷰를 제공함으로써 효율적으로 웹 사이트를 순회할 수 있도록 한다.

본 논문의 구성은 다음과 같다. 2장에서는 순회패턴 탐사에 관한 관련연구들을 기술한다. 3장에서는 효율적인 웹 정보서비스 제공을 위해 빈발 k-페이지집합 기반 순회패턴 탐사를 수행하는 Web Page Recommend(WebPR) 알고리즘을 제안하다. 4장에서는 두개의 실제 사이트의 웹로그(Web-log)를 이용하여 제안한 WebPR 알고리즘들을 비교 분석한다. 마지막 5장에서는 결론 및 향후 연구과제에 대해 기술한다.

2. 관련 연구

사용자들에게 웹 사이트를 고객화하여 제공하는 것은 모든 웹 사이트가 요구하는 공통적인 사항이다. 최근, 보다 나은 웹 정보 서비스를 제공하기 위한 많은 알고리즘들이 제안되었다. 이러한 알고리즘들의 성공 열쇠는 얼마나 신뢰할만한 지식(knowledge)을 원시 데이터(raw data)로부터 발견할 수 있는가에 달려있다[9]. 이 장에서는 웹 정보 서비스를 개선시키기 위해 연구된 기존의 다양한 접근방법들에 대해 살펴본다.

[8]에서 Yan 등(1996)은 자동 고객화 접근방법(automatic customization approach)을 이용한다. 이 시스템은 사용자들의 카테고리(categories)를 구분하기 위하여 웹 서버 로그상에서 유사한 접근 패턴을 가지는 사용자들의 클러스터를 생성한다. Yan 등[8]은 세션을 벡터(vector)로 표현한다. 따라서 웹 탐사에서 중요한 정보인 세션에서의 페이지들이 접근되는 순서를 반영하지 못한다. 또한 이 방법은 프락시나 웹 브라우저에 의한 페이지들의 캐시(cache) 기능에 의한 효과를 무시한다. 일반적으로 이 문제는 사용자의 순회패턴 탐사의 정확도에 영향을 미친다. [5]에서 Perkowitz와 Etzioni는 웹 서버로부터 획득한 로그 데이터를 탐사함으로써 웹 사이트의 구조와 표현 방법을 자동으로 개선시키는 방법을 제안한다. 그들은 세션을 유사도 행렬(similarity matrix)로 표현한다. 이들의 방법은 3.4절의 WebPR(M)의 방법과 유사하다. 이처럼 추출된 정보는 인덱스(index) 페이지집합을 생성하는데 사용되고, 이러한 인덱스 페이지집합은 사용자의 웹 사이트 순회를 도울 수 있도록 제시된다.

대부분의 기존 방법들은 추천 페이지집합을 생성하는데 있어서 웹로그로부터 획득한 세션에서의 페이지간 순차 빈발성을 극히 제한된 형태로만 이용한다. 예를 들어, 순회패턴 탐사에 관한 초기 모델인 벡터 모델[8]의 경우 오로지 한 페이지의 연관성만을 활용하여 추천 페이지집합을 생성한다. 즉, Yan 등[8]은 세션을 벡터(vector)로 표현하고 있기 때문에 웹 탐사에서 중요한 정보인 세션에서의 페이지들이 접근되는 순서를 반영하지 못한다. 한편, 벡터 모델보다는 한 단계 진보한 방법으로 [5]의 행렬 모델이 있다. 그들은 세션에서 인접한 페이지들의 빈발 확률을 유사도 행렬(similarity matrix)로 표현한다. 즉, 페이지 p_1과 p_2의 각 쌍에 대하여 조건부 확률 $P(p_1 \mid p_2)$와 $P(p_2 \mid p_1)$를 계산하여 둘 중의 최

소값을 선택하여 p_1과 p_2사이의 동시 발생 빈발 수를 표현한다. 이처럼 추출된 정보는 인덱스(index) 페이지집합을 생성하는데 사용되고, 이러한 인덱스 페이지집합은 사용자의 웹 사이트 순회를 도울 수 있도록 제시된다. 이 방법에서는 세션에서의 두 페이지간 연관성을 이용하는 것으로 볼 수 있다. 이에 비해 본 논문에서 제안하는 트리모델은 순회패턴 탐사 모델로서 모든 순차 부분집합의 빈발 집합과 액티브 세션을 비교하여 추천 페이지집합을 생성한다.

〈표 2-1〉 기호 표기법(notation)

기 호	의 미
$s, t, u, \cdots$	입력 데이터베이스에서의 일반적인 세션들
D	세션 데이터베이스
s_i	세션 s의 i-번째 페이지
$\|s\|$	세션 s의 길이(length)
VA	Vector Angle에 기반한 유사도
ED	Euclidean Distance에 기반한 유사도
PED	Projected ED에 기반한 유사도
WebPR	Web Page Recommend 알고리즘
WebPR(V|M|T)	V : Vector, M : Matrix, T : Tree; WebPR의 모델들
s_{min}	최소 지지도(minimum support)
F_k	빈발 k-페이지집합
R	추천 페이지집합
N_{rp}	R의 상위 추천 페이지 개수 제한조건
AM	WebPR(M) 빈발 페이지집합으로 사용되는 인접행렬(Adjacency Matrix)
MT, ST	WebPR(T)의 빈발 k-페이지집합을 표현한 메인 트리와 서브트리

3. 빈발 k-페이지집합 기반 순회패턴 탐사

3.1 개 요

이 장에서는 웹로그 데이터를 이용하여 신뢰할 수 있는 빈발 페이지집합 F_k를 탐사하는 다양한 모델들을 제안하고, 이를 웹 사이트에 적용하여 웹 사용자에게 신뢰할 수 있는 정보 서비스를 제공하는 Web Page Recommend(WebPR) 알고리즘을 제안한다.

추천 페이지집합 탐사를 위한 세 개의 WebPR 알고리즘들은 순회패턴들로부터 빈발 k-페이지집합 F_k를 생성하는 방법들에 대한 전체적인 범위를 고려하는 것으로 볼 수 있다. 여기서 F_k는 모든 빈발 k-페이지집합을 의미한다. 기본적으로 네 개의 알고리즘은 자연스러운 확장 개념을 표현한다. 즉, 각 알고리즘은 빈발 k-페이지들의 집합 F_k를 생성함에 있어서 세션에서의 페이지들간의 연관성을 고려하는 정도를 자연스럽게 확장시킨다. 첫 번째 알고리즘인 WebPR(V)는(여기서 "V"는 Vector를 의미한다) 세션에서의 연관성을 고려하는 정도가 가장 약하다. 이 알고리즘은 단순히 세션 데이터베이스에서 한 페이지의 빈발 횟수만을 고려한다. 따라서 빈발 1-세션들의 집합을 생성한다. 두 번째 알고리즘인 WebPR(M)은(여기서 "M"은 Matrix를 의미한다) 세션에서 인접한 페이지들간의 연관성을 고려한다. 이 알고리즘은 빈발 2-페이지들의 집합을 생성한다. 세 번째 알고리즘인 WebPR(T)는(여기서 "T"는 Tree를 의미한다) 세션의 모든 서브패턴에 대해 연관성을 고려하여 빈발 k-세션들의 집합을 생성한다. 세 개의 알고리즘에 의해 생성된 빈발 페이지집합의 종속관계는 세션에서 페이지들간의 연관성을 얼마나 고려했느냐에 달려있다 ; 즉,

$$|F_k^{WebPR(T)}| \le |F_2^{WebPR(M)}| \le |F_1^{WebPR(V)}|.$$

〈표 3-1〉 세 개의 WebPR 알고리즘 비교

알고리즘	페이지들간의 연관성 활용정도	생성된 빈발 집합
WebPR(V)	1 페이지	빈발 1-페이지집합
WebPR(M)	2 페이지	빈발 2-페이지집합
WebPR(T)	모든 페이지	모든 빈발 k-페이지집합

WebPR 알고리즘들은 다음과 같은 세부 단계들로 이루어져 있다 :

(1) 전처리 : 웹로그에 대하여 데이터 클리닝 등의 전처리를 수행하고, IP 주소, 시간 등을 이용하여 세션 데이터베이스 D를 생성한다.
(2) 빈발 페이지집합 탐사 : 빈발 k-페이지집합을 생성한다.
(3) 추천 페이지집합 생성 : 결속력있는 개념을 표현하는 페이지집합을 생성한다.
(4) 추천 적용 : 웹 사이트를 수정하여 직접 동적 추천을 수행하거나 생성된 추천 페이지집합을 웹 마스터에게 제안한다.

3.2 전처리

사용자 세션은 한 사용자에 의해 접근된 순서가 있는 페이지들의 집합을 의미한다. 그러나 웹로그 데이터는 웹 서버에 기록된 일련의 페이지 뷰(page views) 혹은 요청들(requests)로 이루어져 있다. 일반적으로 각 요청은 요청 시간, 요청된 URL, 그 요청이 발생한 곳의 IP 주소 등을 포함하고 있다. 또한 원시 웹로그 데이터에는 순회패턴 탐사 목적에는 불필요한 많은 웹로그 항목(Weblog entry)들이

존재한다. 이러한 웹로그 항목들은 HTTP 프로토콜의 정의에 의해 사용자가 명시적으로 요구하지 않았음에도 불구하고 웹로그 항목으로 기록된 것이기 때문에 사용자의 순회 패턴을 탐사하는 데에는 도움이 되지 않는다. 따라서 이러한 불필요한 웹로그 항목들을 제거하는 과정이 요구된다. (이러한 과정을 데이터 클리닝(data cleaning)이라고 한다.)

데이터 클리닝을 통하여 사용자의 요구에 의해 웹로그 항목으로 기록된 웹로그 항목들로만 구성된 웹로그 집합을 얻었다면, 다음은 사용자별로 순회한 페이지들을 시간 순서에 따라 구성된 페이지집합으로 구분하는 작업이 필요하다. 웹로그로부터 세션(session)을 구분하기 위해서는 한 사용자가 웹 사이트의 순회를 종료한 시점을 알아야 한다. 그러나 HTTP가 연결이 지속되지 않는 프로토콜(stateless protocol)이기 때문에 사용자가 현재의 웹 사이트를 언제 떠났는가를 알 수 없다. 일반적으로 30분 시간제한(timeout)을 이용하는 방법을 취하고 있는데, 이것은 [10]의 실험 결과에 따른 것이다. 본 논문에서도 30분 시간제한(timeout)을 이용하여 세션을 구분한다. 일반적으로 세션을 구분하는 방법은 일정 시간 간격(예 : 30분) 내에 있는 동일한 IP 주소에 의한 웹로그 항목들의 집합을 한 명의 사용자에 의한 기록으로 간주하여 하나의 세션으로 구분한다. 본 논문에서도 마찬가지로 위의 일반적인 방법을 이용하여 세션을 구분한다.

3.3 WebPR(V) 알고리즘

WebPR(V)는 순회패턴 탐사와 추천을 위해 Yan 등[8]의 벡터모델을 단순하게 수정한 것이다. WebPR(V)는 후보 세션 s의 모든 페이지들에 대한 빈발 횟수를 계산한다. 이것은 세션에서 페이지들간의 연관성을 전혀 고려하지 않은 방법으로 빈발 1-세션(즉, 한 페이지)들의 집합을 생성한다. WebPR(V)는 빈발 1-페이지집합 F_1의 집합을 생성하기 위해 세션 데이터베이스의 모든 세션들을 클러스터링하는 과정을 요구한다. 생성된 F_1 집합은 최소 지지도($s_{\min}$)에 따라 달라진다. 또한 추천과정에서 추천 페이지집합 R은 F_1 집합으로부터 추천 페이지 개수 제한조건(N_p)에 따라 생성된다.

3.3.1 클러스터링

WebPR(V)에서는 K-means 알고리즘[9]을 이용하여 클러스터링을 수행한다. 먼저 세션을 n차원(n : 전체 웹 페이지 개수) 벡터(vector) 공간으로 매핑시킨다. 일반적으로 클러스터링 알고리즘은 이 공간을 유사도(similarity measure)에 기반하여 서로 가까운 항목들의 그룹으로 분리해준다.

WebPR(V)에서는 사용자 세션 $s \in D$가 주어지면 다음과 같이 사용자 세션을 벡터로 표현한다 :

$$v_s = \langle v_1, v_2, \cdots, v_n \rangle \tag{1}$$

여기서 $v_i = \begin{cases} 1 & \text{if } s_i \in s \\ 0 & \textit{otherwise} \end{cases}$

여기서 주목할 점은 세션 s에서 페이지들의 빈발 횟수는 고려하지 않는다는 것이다. 즉, s_i는 세션에서의 i-번째 페이지의 존재 여부만을 나타내는 이진(binary) 변수이다. WebPR(V)에서는 세션들의 클러스터링을 위해 유사도(similarity measure) Vector Angle(VA)와 Euclidean Distance(ED)를 사용한다. VA는 두 특징 벡터 사이의 각도거리(angular distance)를 사용하여 유사성을 계산한다 :

$$\mathrm{VA}(\mathbf{x},\mathbf{y}) = \cos\phi = \frac{\mathbf{x}\cdot\mathbf{y}}{|\mathbf{x}||\mathbf{y}|} = \frac{\sum_{i=1}^{N} x_i y_i}{\sqrt{\sum_{i=1}^{N} x_i^2}\sqrt{\sum_{i=1}^{N} y_i^2}}$$

여기서 $\phi = \langle x, y \rangle$이고 $\mathrm{VA} \in [0,1] (x_i, y_i \geq 0)$이다. VA는 유사성을 표현해 주는데, VA(x, y)의 값이 클수록 특징 벡터 x와 y는 더 유사하다고 할 수 있다. ED는 특징 벡터들의 비유사성(dissimilarity)을 정량화시켜 준다. 즉, 그 값이 클수록 비교되는 벡터들은 유사하지 않다. ED는 단순히 두 특징 벡터 사이의 유클리드 거리를 계산한다 :

$$\mathrm{ED}(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2} \tag{3}$$

여기서 $\mathrm{ED} \in [0, \infty]$이다. ED의 값이 작을수록 두 벡터 x와 y의 유사도는 높다. *VA*와 비교해 보면, *ED*는 한 벡터와 그 벡터의 크기만을 변형시킨 벡터를 구분할 수 있다 :

$$\mathrm{ED}(\mathbf{y}, k\mathbf{y}) = |(k-1)\ \mathbf{y}| \neq 0 \quad (\text{if } k \neq 1 \text{ and } \mathbf{y} \neq 0).$$

WebPR(V)는 위에서 기술한 유사도 VA와 ED를 이용하여 클러스터링을 수행한다. 클러스터링의 목적은 유사한 순회패턴을 보이는 세션의 클러스터를 발견하는 것이다. WebPR(V)에서는 잘 알려진 K-means 알고리즘[9]을 적용한다. K-means는 처음에 무작위(random)로 클러스터의 중심(center)을 선택한 후, 선택된 중심과의 거리를 평균하여 다시 클러스터의 중심을 옮겨가는 방식으로 클러스터링을 수행하는 알고리즘이다.

3.3.2 빈발 페이지집합 탐사

WebPR(V)는 K-means 알고리즘에 의해 생성된 각 클러스터의 중심 벡터를 이용하여 각 원소 값이 최소 지지도

(s_{min}) 이하의 값을 가지는 원소들을 필터링(filtering)한다. 이와 같이 생성된 중심 벡터들이 빈발 *1*-페이지집합 CV' (즉, F_1)이 된다. 이것은 세션에서 페이지들간의 연관성을 전혀 고려하지 않는다.

3.3.3 추천 페이지집합 생성

(그림 3-1)은 WebPR(V)의 추천 페이지집합 생성 알고리즘을 보여준다. WebPR(V)의 추천 페이지집합 생성과정은 간단하다. 첫 번째 단계는 새로운 사용자의 카테고리(category)를 얻기 위해 3.3.2절에서 생성한 중심 벡터집합 CV와 액티브 세션 벡터와의 유사도 계산을 통하여 가장 유사한 빈발 1-페이지집합 CV'을 결정하는 것이다(Step 1). 다음은 CV'의 각 페이지 p에 대한 가중치를 가져와서 현재 액티브 세션 s에 대한 추천 페이지집합 R을 생성한다(Step 3-7). 여기서 추천 페이지집합의 크기가 매우 클 수 있다. 따라서 WebPR(V)에서는 한번에 추천되는 페이지집합의 크기에 대해 개수 제한을 둔다. 먼저 R을 가중치의 크기 순으로 정렬시킨다(Step 8). 다음은 사용자에 의해 미리 정해진 추천 페이지집합 크기 제한조건에 따라서 상위 N_{rp}만큼의 페이지들이 추천된다(Step 9).

```
Procedure VectorRecommendation( CV, CV', s, N_rp )
// CV : 클러스터 중심 벡터 집합,  CV' : 빈발 페이지집합
   s : 액티브 세션
// N_rp : 각 추천단계에서의 추천페이지 개수 제약조건
1. determine cluster by computing match(s, CV)
2. R := ∅
3. for each page in the frequent 1-pageset CV'(c) do {
4.     Rec(s, p) := weight(p, CV'(c))
5.     p.rec_score := Rec(s, p)
6.     R := R∪{p}
7. }
8. R' := sort(R)
9. select top N_rp pages from R'
end
```

(그림 3-1) WebPR(V)에 의한 추천 페이지집합 생성

한편, (그림 3-1)의 $match(s, CV)$은 3.3.1절에서 기술한 유사도(VA, ED)에 기반하여 계산된다. 그러나 유사도 VA와 ED는 유사한 길이의 세션들을 비교하는 경우에는 잘 적용될 수 있지만, 액티브 세션의 부분 정보와 한 클러스터의 전체 정보를 표현하는 클러스터 모델을 비교하는 경우에 과대평가(overestimation)될 수 있다. 따라서 본 논문에서는 이러한 과대평가 문제를 완화시킬 수 있는 유사도 PED(Projected ED)[11]를 적용한다. PED는 ED에서의 과대평가 문제를 해결하기 위한 변형된 유사도 측정 방법이다. 예를 들어, 두 벡터 s와 c를 각각 세션 벡터와 클러스터 벡터라고 하자. 각 벡터는 n개의 구성원소를 가진다. 벡터 s와 c의 비유사성을 계산하기 위하여 벡터 s가 영(zero)이 아닌 구성원소를 가지는 좌표평면 상의 c의 사상(projection)을 이용한다 :

$$PED(s,c)=\sqrt{\sum_{i=1, s_i\neq 0}^{n}(s_i-c_i)^2} \quad (4)$$

여기서 PED $\in [0,\infty)$이고, 교환법칙은 성립되지 않는다. 따라서 PED가 여러 가지 길이로 세션들을 비교할 수 있기 때문에 주어진 시간에 세션의 일부만이 이용 가능한 실시간 클러스터링에 유용하게 사용될 수 있다.

3.4 WebPR(M) 알고리즘

WebPR(M)은 빈발 페이지집합을 생성하기 위해 세션 데이터베이스로부터 빈발 2-페이지 집합을 생성한다. 이것은 세션의 인접한 페이지들간의 연관성을 고려한 것이다[5, 9]. WebPR(M)은 F_2 집합을 생성하기 위해 세션 데이터베이스의 모든 세션들을 각 클러스터별로 인접 행렬(adjacency matrix)을 생성한다. 이러한 인접 행렬을 이용하여 액티브 세션의 현재 페이지와 연관성이 높은 페이지집합을 생성할 수 있다. 즉, WebPR(V)에서 생성한 CV를 이용하여 현재 액티브 세션의 클러스터를 찾은 다음 액티브 세션의 현재 페이지와 연관성이 높은(즉, 평균 빈발횟수가 높은) 페이지들의 집합을 생성한다. WebPR(M)의 F_2집합은 최소 지지도(s_{min})에 따라 달라진다. 또한 추천과정에서 추천 페이지집합 R은 F_2집합으로부터 추천 페이지 개수 제한조건(N_{rp})에 따라 생성된다.

3.4.1 클러스터링

WebPR(M)에서의 클러스터링은 3.3.1절의 WebPR(V)의 클러스터링을 그대로 적용한다.

3.4.2 빈발 페이지집합 탐사

WebPR(M)은 먼저 세션 데이터베이스에서의 모든 세션들에 대하여 클러스터별로 인접 행렬(adjacency matrix) AM(즉, 빈발 2-페이지집합 F_2)을 생성한다. 여기서 클러스터별로 AM을 생성할 수 있는 것은 3.3.1절의 클러스터링의 결과로 얻은 각 세션에 대한 *cluster_id*를 세션 데이터베이스에 저장해 두었기 때문이다. WebPR(M)에서는 이러한 cluster_id를 이용하여 클러스터별로 인접 행렬 AM을 생성한다. 예를 들어, 웹 사이트의 페이지집합 $\{1, 2, \cdots, n\}$이 주어지면, 인접 행렬 AM을 $n \times n$ 행렬로 정의한다. 여기서

$AM(i,j)$는 세션에서 페이지 i로부터 j로 가는 링크가 존재하면 1의 값을 추가하고, 그렇지 않으면 무시한다. 이때 한 세션에서 페이지 i로부터 j로 가는 링크가 2회 이상 발생해도 1의 값만 추가된다. 이와 같이 클러스터별로 모든 세션들에 대해 $AM(i,j)$를 계산하였다면, 이제 각 클러스터별 $AM(i,j)$를 해당 클러스터의 크기로 나눈다. 따라서 모든 $AM(i,j) \in [0,1]$이 된다. 마지막으로 모든 클러스터별 $AM(i,j)$에 대하여 최소 지지도($s_{\min}$) 이하의 값을 가지는 원소들을 0의 값으로 한다. 이러한 인접 행렬은 세션에서의 인접한 두 페이지 간의 연관성을 고려한 빈발 2-페이지집합 F_2를 의미한다. 이것은 인접한 페이지 간의 연관성을 전혀 고려하지 않은 WebPR(V)에서 보다 한 단계 확장된 방법으로 볼 수 있다.

3.4.3 추천 페이지집합 생성

(그림 3-2)는 WebPR(M)의 추천 페이지집합 생성 알고리즘을 보여준다. WebPR(M)의 추천 페이지집합 생성을 위한 첫 번째 단계는 새로운 사용자의 카테고리(category)를 얻기 위해 3.4.2절에서 생성한 클러스터별 인접행렬 AM (즉, 빈발 2-페이지집합 F_2)과 액티브 세션 벡터 s와의 유사도 계산을 통하여 가장 유사한 빈발 2-페이지집합 $AM(c)$ (즉, $F_2(c)$)을 결정하는 것이다(Step 1). 다음은 $AM(c)$로부터 액티브 세션 s의 현재 페이지에 대응되는 행 r을 추출한다(Step 2). 이러한 행벡터 $\mathrm{r} = \langle r_1, r_2, \cdots, r_n \rangle$의 각 페이지들이 후보 추천 페이지집합이 된다. 즉, $AM(c)$의 행벡터 r의 각 페이지에 대한 가중치를 가져와서 현재 액티브 세션 s에 대한 추천 페이지집합 R을 생성한다(Step 4-8). 여기서 추천 페이지집합의 크기가 매우 클 수 있다. 따라서 WebPR(M)에서도 마찬가지로 한번에 추천되는 페이지 집합의 크기에 대해 개수 제한을 둔다. 먼저 R을 가중치의 크기 순으로 정렬시킨다(Step 9). 다음은 사용자에 의해 미리 정해진 추천 페이지집합 크기 제한조건에 따라서 상위 N_p 만큼의 페이지들이 추천된다(Step 10).

```
Procedure MatrixRecommendation( CV, AM, s, N_p )
// CV : 클러스터 중심 벡터집합, AM : 인접 행렬, s : 액티브 세션
// N_p : 각 추천단계에서의 추천페이지 개수 제약조건
1. determine cluster AM(c) by computing match(s, CV) ;
2. AM(c)로부터 s의 현재 페이지에 대응되는 행 r 추출 ;
3. R := ∅ ;
4. for each page p of selected row vector r of AM(c) do {
5.     Rec(s, p) := weight(p, r) ;
6.     p.rec_score := Rec(s, p) ;
7.     R := R ∪ {p} ;
8. }
9. R' := sort( R)
10. select top N_p pages from R'
end
```

(그림 3-2) WebPR(M)에 의한 추천 페이지집합 생성

3.5 WebPR(T) 알고리즘

WebPR(T)는 추천 페이지집합 R을 생성하기 위해 세션의 페이지들 사이에 존재하는 모든 연관성을 이용하는 알고리즘이다. WebPR(T)는 먼저 세션 데이터베이스로부터 메인트리와 서브트리를 생성한다. 즉, 빈발 k-페이지집합 F_k의 집합이 이러한 트리들로 표현된다. 일단 트리들이 생성되면 이러한 트리들을 이용하여 N_p에 따라 추천 페이지집합 R을 생성할 수 있다.

3.5.1 빈발 페이지집합 탐사

(1) 트리 생성

(그림 3-3)은 WebPR(T) 트리 생성 알고리즘을 보여준다. WebPR(T)는 세션 데이터베이스로부터 세션들을 가져와서 하나의 메인 트리(main tree)와 하나의 서브트리(sub-tree)를 생성한다. 여기서 트리의 각 노드는 세션의 페이지를 표현한 것으로 빈발횟수, 페이지 제목, URL, 날짜(date) 필드들로 구성된다.

WebPR(T)에서는 메인트리와 서브트리를 동시에 생성한다. 첫 번째 단계는 MT의 루트(인덱스) 노드와 ST의 더미(루트) 노드를 생성하는 것이다(Step 1). 두 번째 단계에서는 세션 데이터베이스의 각 세션을 가져와서 세션 s의 첫 번째 페이지(인덱스 페이지)를 제거한다(Step 3). 다음은 s의 리스트를 $[p \mid P]$로 분리하는데, 여기서 p는 첫 번째 원소이고 P는 나머지 원소들을 의미한다(Step 4). 메인트리 MT를 생성하기 위한 마지막 단계로 insert_tree($[p \mid P]$, MT)를 호출하여(Step 5), s의 모든 페이지를 빈발횟수 $freq$와 함께 MT에 표현한다. insert_tree($[p \mid P]$, MT)는 s의 각 페이지를 트리로 표현해주는 재귀적(recursive) 함수이다((그림 3-6)의 하단).

```
Procedure TreeGeneration ( D)
// D : 세션 데이터베이스, 모든 세션의 첫 번째 페이지는 index 페이지
// MT : WebPR(T)의 메인트리, ST : WebPR(T)의 서브트리,
1. create the root(index) node of a MTand the dummy(root)
   node of a ST ;
2. for each session s in D do {
3.   remove the first(index) page in
4.   let the list in s be [p | P],
       where p is the first element and P is the remaining list.
5.   call insert_tree( [p | P], MT) // 메인트리 생성
```

```
6.   repeat { // 서브트리 생성
7.   for each subsession s' in s do {
8.     remove the first page in s'
9.   let the list in s' be [p | P],
         where p' is the first element in s'.
10.    call insert_tree( [p | P], ST)
11.  }
12. } until ( P is empty)
13. }
end

Function insert_tree ( [p | P], T)
1. if ( T has a child N such that L_last.pname = p.pname) then {
2.    increment N's freq by 1
3. } else {
4.    create a new node N and let its freq be 1
5. }
6. if ( P is nonempty) then {call insert_tree( P, N) }
end
```

(그림 3-3) WebPR(T) 트리 생성 알고리즘

한편, 서브트리 ST를 생성하기 위한 첫 번째 단계로 서브세션 s'의 첫 번째 페이지를 제거한다(Step 9). 다음은 s'의 리스트를 $[p' | P]$로 분리하는데, 여기서 p'은 첫 번째 원소이고 P는 나머지 원소들을 의미한다(Step 9). 서브트리 ST를 생성하기 위한 마지막 단계로 insert_tree($[p' | P]$, ST)를 호출하여(Step 10), s'의 모든 페이지를 빈발횟수 $freq$와 함께 MT에 표현한다. 이와 같은 과정을 $P=\emptyset$이 될 때까지 반복한다(Step 12).

(2) 트리 유지관리

(그림 3-4)는 트리의 추천 성능을 개선하기 위한 WebPR(T) 트리 유지관리 알고리즘을 보여준다. WebPR(T)의 메인트리와 서브트리는 주기적으로 유지관리(maintenance)를 수행함으로써 추천 성능을 개선시킬 수 있다. 예를 들어, 주기적으로(예 : 한 달간) 새롭게 획득한 세션 정보를 이용하여 트리를 재구성할 수 있다. 이것은 사용자들의 최신 경향을 반영하는 것이 된다. 다음 단계는 트리의 각 노드 정보를 이용하여(예 : 참조 횟수, 날짜 등) 추천을 위한 후보 노드들을 가지치기(candidate pruning) 하는 것이다. 트리의 가지치기 과정에서는 노드의 참조 횟수 필드가 임계값보다 적고, 노드의 생성된 날짜가 임계값으로 지정된 기간을 초과한 경우라면 그 노드를 제거하는 것이다.

```
Procedure TreeMaintenance ( MT, ST, D')
// MT : WebPR(T)의 메인트리, ST : WebPR(T)의 서브트리,
// D' : 새로운 세션 데이터베이스
1. // 새로운 페이지집합 추가
2. call TreeGeneration using D'
3. // 후보 가지치기(candidate pruning)
4. for each node a in Tree do
5.    // s_min : 최소 지지도, τ : 최소 기간 임계값
6.    if ( a.freq < s_min and (curr_date − a.data) < τ) then
7.       remove that node from Tree
8.    }
9. }
end
```

(그림 3-4) 트리 유지관리 모듈

3.5.2 추천 페이지집합 생성

WebPR(T)는 추천 페이지집합 R을 생성하기 위해 세션의 페이지들 사이에 존재하는 모든 연관성을 이용한다. 즉, WebPR(T) 메인트리 MT와 서브트리 ST에 표현된 빈발 페이지집합 $\{F_1, \cdots, F_k\}$를 이용하여 R을 생성한다. WebPR(A)가 빈발 페이지집합 F_{l_c} ($l_c \geq 2$) (l_c : 액티브 세션의 현재 길이)만을 이용하여 R을 생성하는 반면에, WebPR(T)에서는 $\{F_1, \cdots, F_k\}$ 모두를 이용하여 R을 생성한다. 이것은 세션의 페이지간 연관성을 최대한 이용하는 방법이라고 할 수 있다. 예를 들어, 현재 액티브 세션의 길이가 3이라고 하면 ($l_c=3$), $\{F_1, F_2, F_3\}$을 이용하여 R을 생성한다.

```
Procedure TreeRecommendation( MT, ST, s_a, N_rp)
// MT : WebPR(T) 메인트리, ST : WebPR(T) 서브트리
// s_a : 액티브 세션, N_rp : 각 추천단계에서의 추천페이지 개수 제약
      조건
1. R : = ∅
2. for each active session s_a do {
3.    if ( s_a={index}) then {
4.       R: = R∪{children of foot node in MT}
5.       break
6.    }
7.    // MT에서 추천
8.    remove the first(index) page in s_a
9.    let the list in s_a be [P|p],
         where p is the last element and P is the remaining list.
10.   if ( MT has a L path such that L_last.pname = p · pname) then {
11.      R: = R∪{children of p}
12.   }
13.   if (|s_a| < 2) then { break }
14.   repeat { // ST에서 추천
15.      for each subsession s_a' in s_a do {
16.         remove the first page in s_a'
          let the list in s_a' be [P'|p] ,
17.         where P' is the remaining list(the first page removed).
18.      call recommend( [P'|p], ST)
19.      }
20.   } until ( P' is empty)
21. }
22. R' = sort(R)
23. select top N_rp pages from R'
end
```

```
Function recommend ( [P|p], T)
1.   if ( T has a path L such that L_last.pname = p.pname) then {
2.:        R := R ∪ {children of p}
3. }
end
```

(그림 3-5) WebPR(T)의 추천 페이지집합 생성 알고리즘

(그림 3-5)는 WebPR(T)의 추천 페이지집합 생성 알고리즘을 보여준다. R을 생성하기 위한 첫 번째 단계는 액티브 세션 s_a가 *index* 페이지이면(사용자가 웹 사이트의 홈에 접근한 상태), 의 *root* 노드의 자식 노드들을 R에 포함시키고 루프를 빠져나간다(Step 3-6). 다음 액티브 세션이 한 단계 더 진행하면 MT 먼저 MT에서 추천 페이지집합 R을 생성한다. 먼저 액티브 세션 s_a로부터 첫 페이지인 인덱스 페이지를 제거하고(Step 8), s_a의 리스트를 $[P \mid p]$로 구분한다(Step 9). 여기서 p는 마지막 원소를 의미하고 P는 나머지 리스트를 의미한다. 만일 MT가 $L_{last}.pname = p.pname$을 만족하는 경로 L을 가진다면 p의 자식노드들을 추천 페이지집합에 포함시킨다(Step 10-12). 또한 현재 액티브 세션이 인덱스 페이지를 지나 첫 번째 페이지에 있을 때의 추천은 메인트리에서만 이루어지기 때문에 서브트리를 탐색할 필요가 없다(Step 13).

다음은 ST에서 추천하는 단계로 s_a의 서브세션 s_a'에 대해 첫 번째 페이지를 제거한 s_a'의 리스트를 $[P' \mid p]$로 구분한다(Step 16-17). 여기서 P'는 첫 번째와 마지막 원소를 제외한 나머지 리스트를 의미한다. 이제 recommend($[P' \mid p]$, ST)를 호출한다((그림 3-10)의 하단)(Step 18). recommend($[P \mid p]$, T)는 ST에서만 이용되는 것으로 s_a가 포함하는 모든 서브세션에 대해 서브트리로부터 추천 페이지집합을 생성할 때 이용되는 재귀적 함수이다. 이러한 과정을 $P=\varnothing$이 될 때까지 계속한다.

추천 과정의 마지막 단계로 MT와 ST에서 획득한 추천 페이지집합 R을 *freq*의 크기 순으로 정렬시킨다(Step 22). 다음은 사용자에 의해 미리 정해진 추천 페이지집합 크기 제한조건에 따라서 상위 N_p만큼의 페이지들을 최종적으로 추천한다(Step 23).

4. 실험 및 평가

4.1 실험 방법

WebPR 알고리즘들의 평가는 다음과 같은 척도에 기반하여 평가한다[5] : Impact, Benefit. **Impact**는 얼마나 많은 사용자가 얼마나 자주 추천한 페이지를 이용하였나를 평가하고, **Benefit**은 웹 사이트를 방문한 사용자들이 얼마나 많은 노력을 절약하였나를 평가한다. 본 논문에서는 각 사이트의 웹 서버 로그 데이터를 두개의 그룹으로 나누어 실험을 수행한다; 훈련 데이터와 테스트 데이터. 실험에 사용하는 척도는 위에서 언급한 Impact/Benefit을 기본적으로 사용한다. 3장에서 기술한 WebPR 알고리즘들을 먼저 훈련 데이터에 적용하여 추천 페이지집합을 생성한 후 테스트 데이터를 이용하여 Impact/Benefit을 비교함으로써 평가한다.

본 논문에서는 제안한 알고리즘과 다른 탐사 알고리즘과의 비교를 위해 **impact**와 **benefit** 척도를 이용한다. 각 클러스터에 대하여 각 사용자에 의해 순회된 추천 페이지집합 내의 페이지 개수를 카운트하고, 적어도 한 페이지를 방문한 사용자들의 수, 적어도 두 페이지를 방문한 사용자들의 수 등을 계산한다. 다음은 특정 알고리즘에 의해 생성된 모든 클러스터에 대해 평균을 계산한다. 각 알고리즘에 대해 **benefit**(추천 페이지집합에서 선택된 페이지 수) 대 **impact**(페이지 개수별로 순회한 사용자들의 수)를 그래프로 표현한다. 만일 특정 알고리즘에서 특정한 개수 이상의 페이지들(예 : 일곱 개 이상의 페이지들)을 순회한 사용자들이 없다면 해당 알고리즘의 그래프는 거기서 종료한다. 모든 실험에서 각 알고리즘은 조정할 수 있는 파라메타(parameters)를 갖는다. 모든 경우에 최적의 결과가 나오도록 파라메타를 조정한다.

4.2 실제 데이터

본 논문에서의 실험은 두 개의 웹 사이트로부터 얻은 로그 데이터를 이용하여 수행한다. 첫 번째 사이트는 아시아나 항공의 LadyAsiana 사이트이다. 이 사이트는 211개의 서로 다른 웹 페이지들로 구성되어 있으며, 그 외 많은 이미지와 텍스트 등이 포함되어 있다. LadyAsiana 사이트로부터 총 84일간의 로그 데이터를 얻었다. 두 번째 사이트는 KBS 방송국의 미디어서버 사이트로 이 사이트는 62개의 서로 다른 웹 페이지들로 구성되어 있으며, 총 16일간의 로그 데이터를 얻었다.

실험 환경으로 OS는 Windows 98, CPU는 Pentium Ⅲ, RAM 512MB, 프로그래밍 언어는 JAVA를 사용하였다. <표 4-1>은 두개의 웹로그에 대해 훈련 데이터와 테스트 데이터의 실험조건들을 보여준다.

본 논문에서는 세션의 길이가 너무 짧거나 혹은 너무 긴 경우는 관심이 없다. 즉, 많은 사용자들의 순회패턴에 적용하기 위해 최소 페이지 개수(즉, 3페이지) 미만인 세션들과 최대 페이지 개수(즉, 20페이지 혹은 25페이지)를 초과하는 세션들을 제거한다. 또한 최소 페이지 개수는 두개의 사이트 모두 3페이지 이상인 세션들로 제한하고, 최대 페이지 개수는 LadyAsiana 사이트의 경우 20페이지, 그리고 KBS-Media 사이트의 경우는 25페이지로 각각 다르게 제한을 두었다. 이것은 두개의 웹로그에 대한 세션길이의 분포도에

〈표 4-1〉 실험 환경

구 분		Train Data			Test Data	
		Size	세션개수	수행시간	Size	세션개수
LadyAsiana	WebPR(V)	78M (761678건)	117,893	11시간 10분	17M (161346건)	24,330
	WebPR(M)			13시간 05분		
	WebPR(T)			7시간 25분		
KBSMedia	WebPR(V)	42M (425725건)	34,576	4시간 20분	13M (126616건)	10,120
	WebPR(M)			5시간 40분		
	WebPR(T)			8시간 40분		

따른 것이다.

4.3 실제 데이터를 이용한 WebPR 알고리즘들의 성능 평가

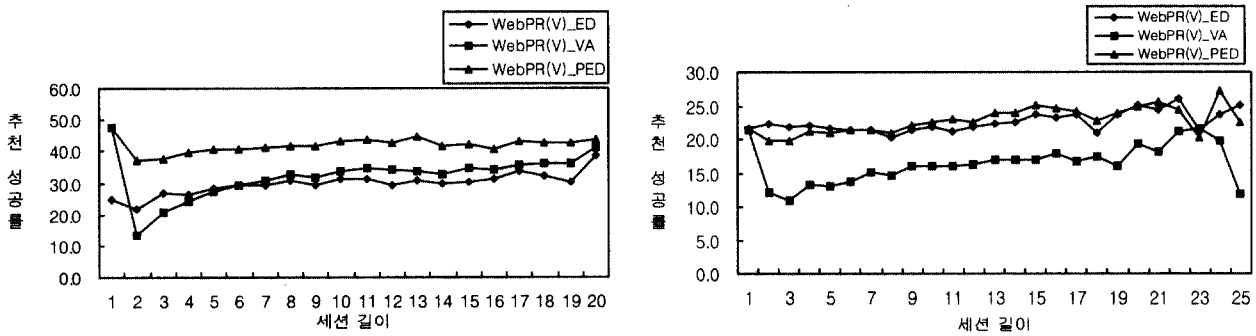
이 절에서는 제안한 WebPR 알고리즘들을 4.2절에서 기술한 두개의 실제 데이터에 적용한 실험을 수행한다. 실험 방법은 4.1절에서 기술한 방법과 동일하다.

4.3.1 클러스터링

3.3절에서 언급한 바와 같이 WebPR(V)와 WebPR(M)에서는 클러스터링 과정이 요구된다. 본 논문에서는 잘 알려진 K-means 알고리즘을 이용하여 세션 데이터베이스의 세션들을 클러스터링한다. 유사도 ED를 적용한 경우 LadyAsiana는 K = 42개, 그리고 KBSMedia는 K = 37개의 클러스터들을 생성하였다. 유사도 VA를 적용한 경우 LadyAsiana는 K = 66개, 그리고 KBSMedia는 K = 39개의 클러스터들을 생성하였다. K-means 알고리즘을 수행할 때 초기 K의 개수는 각 사이트의 전체 페이지 개수로 정하여 시작하였다. 종료조건은 각 클러스터의 세션들이 40개 이하로 변동될 때까지 수행하는 것으로 주었다. 대부분의 경우 6-9 사이클 내에서 수렴하였다.

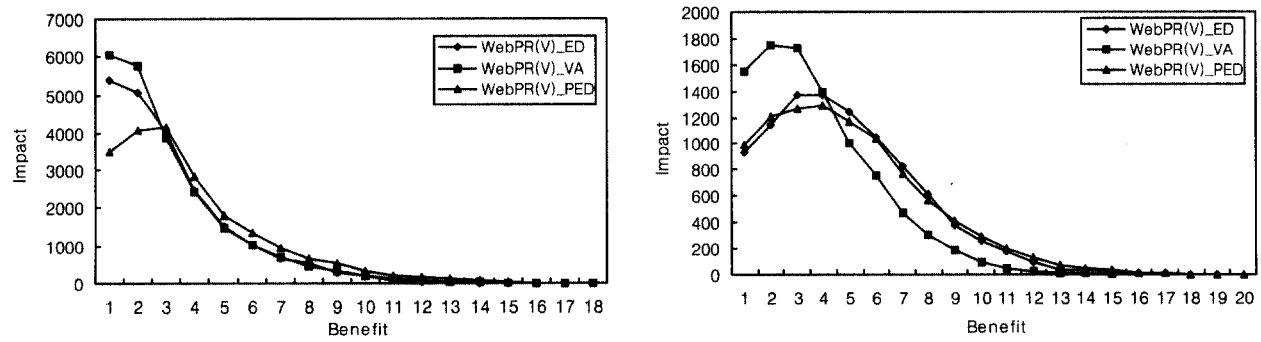
4.3.2 WebPR(V)_VA, WebPR(V)_ED, WebPR(V)_PED의 성능 평가

WebPR(V)_VA, WebPR(V)_ED, WebPR(V)_PED 중에서



(a) Lady Asiana (b) KBS Media

(그림 4-1) WebPR(V)_ED, WebPR(V)_VA, WebPR(V)_PED의 세션 길이별 추천 성공률



(a) Lady Asiana (b) KBS Media

(그림 4-2) WebPR(V)_ED, WebPR(V)_VA, WebPR(V)_PED의 성능 평가

가장 성능이 좋은 모델을 선택하여 WebPR(V)로 표기한다. 가장 좋은 모델 선택을 위해 먼저 각 모델들의 세션 길이별 추천 성공률을 조사하였다. (그림 4-1)은 $N_{rp}=10$인 경우의 WebPR(V)_PED 모델이 가장 우수함을 보여준다. $N_{rp}=10$인 경우의 LadyAsiana에 대하여 평균 추천 성공률은 WebPR(V)_ED가 41.7%, WebPR(V)_VA가 43.3%, 그리고 WebPR(V)_PED가 49.9%를 보여주었다. 반면에 $N_{rp}=5$인 경우의 평균 추천 성공률은 WebPR(V)_ED가 26.9%, WebPR(V)_VA가 28.5%, 그리고 WebPR(V)_PED가 41.0%를 보여주었다. KBSMedia의 경우에도 이와 유사한 결과를 보여주었다. 따라서 이후부터 $N_{rp}=10$인 경우의 WebPR(V)_PED 모델을 WebPR(V)의 대표 모델로 간주하여 WebPR(V)로 표기한다. 또한 모든 실험에서 $N_{rp}=10$인 경우와 $N_{rp}=5$인 경우에 대해 실험하였으나 모든 실험 방법에서 $N_{rp}=10$인 경우에 더 좋은 성능을 보였다. 따라서 이후 모든 실험 결과는 $N_{rp}=10$인 경우만을 보여준다. (그림 4-2)는 WebPR(V)_VA, WebPR(V)_ED, WebPR(V)_PED의 Impact/Benefit 성능 평가를 보여준다. 그림에서 WebPR(V)_PED의 결과가 가장 우수함을 보여준다.

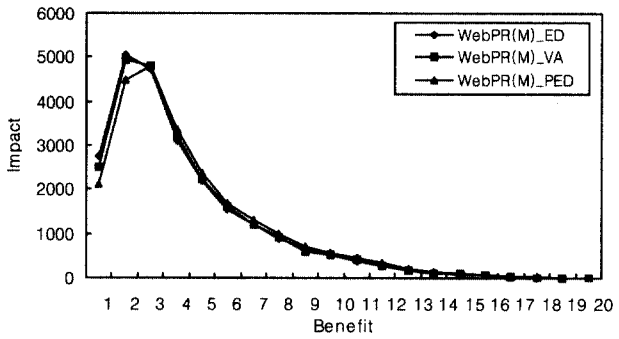
4.3.3 WebPR(M)_VA, WebPR(M)_ED, WebPR(M)_PED의 성능 평가

WebPR(M)_VA, WebPR(M)_ED, WebPR(M)_PED 중에서 가장 성능이 좋은 모델을 선택하여 WebPR(M)으로 표기한다.

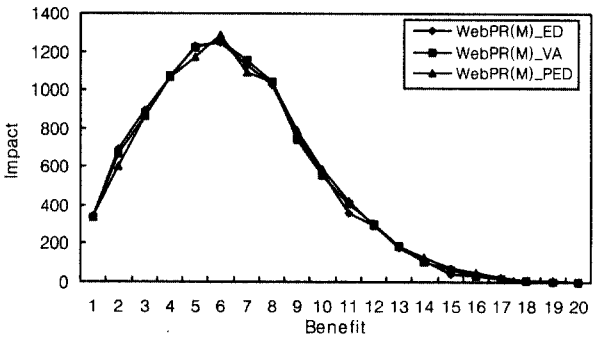
가장 좋은 모델 선택을 위해 먼저 각 모델들의 세션 길이별 추천 성공률을 조사하였다. 실제 $N_{rp}=10$인 경우의 LadyAsiana에 대하여 평균 추천 성공률은 WebPR(M)_ ED가 62.2%, WebPR(M)_VA가 63.2%, 그리고 WebPR(M)_ PED가 67.2%를 보여주었다. 반면에 $N_{rp}=5$인 경우의 평균 추천 성공률은 WebPR(M)_ED가 51.0%, WebPR(M)_VA가 53.5%, 그리고 WebPR(M)_PED가 57.0%를 보여주었다. KBSMedia의 경우에도 이와 유사한 결과를 보여주었다. 따라서 이후부터 $N_{rp}=10$인 경우의 WebPR(M)_PED 모델을 WebPR(M)의 대표 모델로 간주하여 WebPR(M)으로 표기한다. (그림 4-3)은 WebPR(M)_VA, WebPR(M)_ED, WebPR(M)_PED의 Impact/Benefit 성능평가를 보여준다. 그림에서 WebPR(M)_PED의 결과가 가장 우수함을 보여준다.

4.3.4 WebPR(V), WebPR(M), WebPR(T)의 성능 평가

LadyAsiana의 경우 평균 추천 성공률은 WebPR(V)가

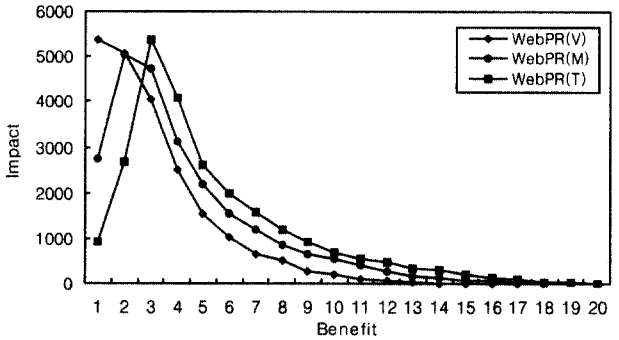


(a) Lady Asiana

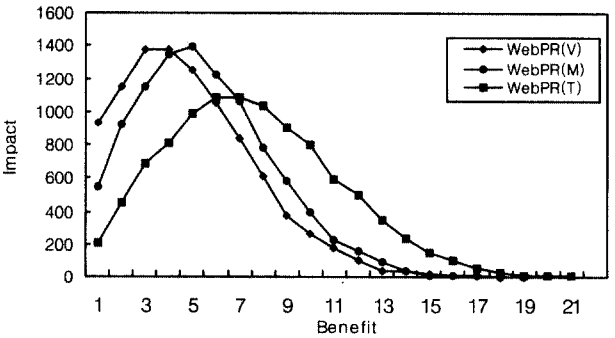


(b) KBS Media

(그림 4-3) WebPR(M)_ED, WebPR(M)_VA, WebPR(M)_PED의 성능 평가



(a) Lady Asiana



(b) KBS Media

(그림 4-4) WebPR(V), WebPR(M), WebPR(T)의 성능 평가

49.9%, WebPR(M)이 67.2%, 그리고 WebPR(T)가 79.3%로 WebPR(T)의 결과가 다른 세 개의 모델에 비해 월등한 추천 성능을 보여주었다. KBSMedia의 경우 LadyAsiana의 경우보다는 추천 성능이 약간 떨어진다. WebPR(V)가 37.5%, WebPR(M)이 52.0%, 그리고 WebPR(T)가 59.9%를 보여준다.

(그림 4-4)는 WebPR(V), WebPR(M), WebPR(T)를 두개의 실제 사이트에 적용한 Impact/Benefit 성능평가를 보여준다. 그림에서 알 수 있듯이 두 사이트 모두에서 WebPR(T)의 추천 성능이 가장 우수한 것으로 나타났다. 또한 세션 길이별 분포가 짧은 쪽으로 치우친 LadyAsiana의 경우에는 세 개의 모델들의 추천 성능 그래프가 매우 유사한 모양을 가진다. 그러나 세션 길이별 분포가 골고루 분포된 KBSMedia의 경우는 WebPR(T)의 그래프가 다른 두 개의 그래프와 많은 차이를 보여준다. 이와 같은 결과는 WebPR(T)의 추천 방식이 현재 액티브 세션의 성향을 최대한 활용하여 추천을 수행하기 때문인 것으로 판단된다.

5. 결 론

본 논문에서는 세션에 나타나는 페이지들간의 연관성 정보를 활용하여 빈발 k-페이지집합을 생성하고, 이를 기반으로 하여 추천 페이지집합을 탐사함으로써 효율적인 웹 정보서비스를 제공할 수 있는 PageRecommend(WebPR) 알고리즘을 제안하였다. 제안한 WebPR 알고리즘은 빈발 k-페이지집합을 탐사하기 위해 페이지들간의 연관성 정보를 이용한다. 기존 연구들과의 가장 큰 차이점은 페이지들간의 연관성 정보를 활용하는 방법들을 전체 범주에 걸쳐서 일관성 있게 고려하고 있다는 점과 가장 효율적인 트리 모델을 제안하고 있다는 점을 들 수 있다. WebPR 알고리즘들은 웹로그로부터 빈발 k-페이지집합을 탐사하고, 이를 기반으로 하여 추천 페이지집합을 생성한다. 특히 제안한 트리 모델은 세션에 나타나는 페이지들간의 모든 연관성 정보를 활용함으로써 가장 우수한 성능을 보임을 실험결과를 통하여 알 수 있다. 이와 같이 추천 페이지집합에 기반하여 웹 사이트를 방문한 사용자에게 추천 페이지집합을 포함하는 새로운 페이지뷰(page view)를 제공함으로써 궁극적으로 찾고자하는 목표 페이지에 효과적으로 접근할 수 있도록 한다. 두 개의 실제 웹 사이트로부터 얻은 웹로그 데이터에 적용한 실험 결과에서 알 수 있듯이 페이지간의 연관성 정보를 활용하는 정도가 높을수록 좋은 추천 성능을 보인다.

향후 연구과제로는 페이지간의 연관성 정보를 활용하는 정도를 좀 더 체계적으로 기술하는 것과 다양한 실험을 통하여 WebPR 알고리즘의 특성을 파악하는 것이다. 마지막으로 더 많은 실제 사이트에 적용하는 것과 데모 사이트를 구축하여 WebPR 알고리즘을 적용하는 것이다.

참 고 문 헌

[1] W3C Web Characterization Activity, http://www.w3.org/WCA/, 2003.

[2] J. E. Pitkow, "Summary of WWW characterizations," Web Journal, 2, pp.3-13, 1998.

[3] M. Spiliopoulou, "Web usage mining for site evaluation : making a site better fit its users," Communications of ACM, 43, pp.127-134, 2000.

[4] M. C. Drott, "Using web server logs to improve site design," Proceedings on the Sixteenth Annual International Conference on Computer Documentation, Quebec, Canada, pp.43-50, 1998.

[5] M. Perkowitz and O. Etzioni, "Towards adaptive Web sites : Conceptual framework and case study," Artificial Intelligence, Vol.118, pp.245-275, 2000.

[6] A. Buchner and M. D. Mulvenna, "Discovering internet marketing intelligence through online analytical Web usage mining," SIGMOD Record, 27(4), 1999.

[7] B. M. Sarwar, G. Karypis, J. A. Konstan and J. Riedl, "Analysis of recommender algorithms for e-commerce," ACM E-Commerce'00 Conference, Mineapolis, MN, pp. 158-167, 2000.

[8] T. W. Yan, M. Jacobsen, H. G. Molina and U. Dayal, "From User Access Patterns to Dynamic Hypertext Linking," The 5th Int'l World Wide Web Conf., Paris, France, May, 1996.

[9] J. Han and M. Kamber, "Data Mining : Concepts and Techniques," Morgan Kaufmann publishers, pp. 349-351, 2001.

[10] L. Catledge and J. Pitkow, "Characterizing browsing behaviors on the world wide web," Computer Networks and ISDN Systems, 27(6), 1995.

[11] C. Shahabi, F. Banaei-Kashani, J. Faruque and A. Faisal, "Feature Matrices : A Model for Efficient and Anonymous Web Usage Mining," EC-Web 2001, Germany, September, 2001.

윤 선 희

e-mail : sunniyoon@hanmail.net

1986년 숭실대학교 전자계산학과(공학사)

1988년 숭실대학교 전자계산학과(공학석사)

2003년 숭실대학교 전자계산학과(공학박사)

1992년~현재 미림전산고등학교 교사

관심분야 : 데이터마이닝, 웹컴퓨팅, 멀티미디어 통신, 멀티미디어 응용 등

김 삼 근

e-mail : skim@ce.hankyong.ac.kr
1985년 부산대학교 계산통계학과(이학사)
1988년 숭실대학교 전자계산학과(공학석사)
1998년 숭실대학교 전자계산학과(공학박사)
1992년~현재 한경대학교 컴퓨터공학과 부교수
관심분야 : 신경망, 데이터마이닝, 웹컴퓨팅, 멀티미디어 통신 등

이 창 훈

e-mail : be4u@hnu.hankyong.ac.kr
1987년 광운대학교 전자계산학과 이학사
1989년 중앙대학교 전자계산학과(이학석사)
1998년 중앙대학교 컴퓨터공학과(공학박사)
1999~2002년 중앙대학교 정보통신연구소 연구전담교수
2002년~현재 한경대학교 컴퓨터공학과 조교수
관심분야 : 소프트웨어공학, 형식명세기법, 컴포넌트 기반 방법론, 품질 및 프로세스개선 등