Stochastic Process 모델을 이용한 웹 페이지 추천 기법

노 수 호° 박 병 준
광운대학교 컴퓨터과학과
noso77@kw.ac.kr°, bjpark@kw.ac.kr

Web Page Recommendation using Stochastic Process Model

Soo Ho Noh° Byung Joon Park
Department of Computer science, Kangwoon University

요 약

다양하고 많은 양의 정보가 존재하는 웹 환경에서 웹사이트를 방문하는 사용자의 접근패턴도 매우 다양하며, 웹 환경의 변화에 따라서 이러한 접근패턴은 계속 변화한다. 이러한 이유로, 웹 사이트 개발자가 사전에 사용자의 욕구에 완벽하게 부합하는 완벽한 사이트를 개발하기란 사실상 불가능하다. 이에 대한 해결방안으로, 웹사이트에 대한 사용자 접근 패턴을 학습해서 웹사이트의 구조나 외형을 자동적으로 개선시켜 나가는 적응형 웹사이트 (Adaptive Web site)가 제시되었다.

본, 논문에서는 DTMC(descrete-time Markov chain)에 의거한 확률적 모델을 이용하여 적응형 웹사이트 구축에 필요한 사용자 접근패턴을 학습하고 이를 적용하기 위한 효과적인 방법론을 제시한다.

1. 서 론

WWW 환경에서 하나의 웹 사이트는 하이퍼링크로 연결 되어 있는 수많은 HTML 문서들의 집합으로 이루어져 있다. 초기의 웹 사이트는 각 HTML 문서들이 지닌 의미와 문서들 간의 상호관계 등을 고려해 최상의 웹 사이트를 구현하고자 하는 웹 마스터의 의도가 반영된 것이다. 하지만 동적으로 변화해가는 사용자들의 요구를 반영하여 보유정보를 효과적으로 제공할 수 있도록 지속적으로 웹사이트의 설계를 바꾸어 가는 일은 매우 어려운 일이다. 하나의 웹사이트에 대한 사용자들의 접근 로그 데이터는 이 사이트에서 일반적으로 사용자들이 보여주는 정보접근 패턴을 알아낼 수 있는 중요한 자료를 제공한다. 본 논문에서는 이와 같은 웹 로그안의 사용자 접근 패턴을 학습해서 웹사이트의 구조나 외형을 자동적으로 개선시켜 나가는 적응형 웹사이트를 구축하는 한 가지 방안을 제시한다[1].

적응형 웹 사이트를 구축하기 위해서, Apriori 알고리즘과 같은 기존의 웹 로그 패턴 분석 방식들은 사용자의 접근패턴을 분석하여, 현재 사용자가 참조하고 있는 페이지를 기준으로 바로 다음단계에 사용자가 참조할 만한 연관성 있는 페이지를 추천한다[2]. 그러나 이러한 방식은 사용자가 최종 목적 페이지 참조를 위하여 불필요하게 중간 페이지들을 참조하게 되는 가능성을 여전히 가지고 있다.

본 논문에서는 DTMC(descrete-time Markov chain)기반의 확률적 모델을 사용, 웹 사용자의 패턴을 학습하여, 현재 페이지를 기준으로, 바로 다음 단계에서 사용자가 참조할 가능성이 높은 페이지들의 링크를 제시하는 것이 아니라, 임의의 N 단계 이후에 예상되는 참조 웹 페이지를 단계별로 추천함으로써 사용자가 최종적으로 방문할 것으로 예상되는 웹 페이지로 직접 이동할 수 있게 하여, 관련성은 높으나 접근경로가 긴 문서들을 효과적으로 추천할 수 있다는 특징을 지닌다. 본 논문의 구성은 2장에서 기반이 되는 배경지식들을 살펴보고, 3장에서는 본 연구에서 소개하는 웹 페이지 추천 시스템을 설명한 후, 4장에서 실험 결과를 보여준다. 마지막으로 5장에서 결론 및 향후 과제에 대해서 설명한다.

2. 연구배경

2.1 웹 로그 마이닝

웹 서버안의 로그는 사용자들이 그 서버로 접근한 기록들을 가지고 있다. 각각의 기록들은 클라이언트들의 IP 어드레스, 클라이언트의 요청이 받아들여진 날짜와 시간, 요청되어진 페이지의 URL, 요청의 프로토콜 등을 담고 있다. 우리는 이러한 웹서버 로그를 통해서 사용자의 접근 패턴들을 추출할 수 있으며, 이러한 패턴을 통해서 어떻게 웹 사이트를 효과적으로 재구성 할 수 있는지에 대한 정보를 얻어낼 수 있다[3].

2.2 DTMC(Descrete-time Markov chain)의 주변일도 (Marginal Distribution)

아래의 정의를 만족하는 stochastic process$\{X_n, n \square\square 0\}$를 DTMC라고 말한다.

$$P\{X_n+1 = j | X_n = i, X_n-1 = i_n-1, \ldots, X_0 = i_0\} = P\{X_{n+1} = j | X_n = i\}$$

실세계의 많은 시스템들은 DTMC로 모델링 될 수 있으며 DTMC의 초기상태(initial distribution) a(0) = a, 상태전이행렬(state transition matrix) P가 있을때, DTMC를 따르는 시스템에서 n번째 step이후의 시스템의 상태를 아래의 식을 통하여 랜덤변수 X_n의 주변일도를 구함으로써 예측할 수 있다[4].

$$a^{(n)} = a^{(0)}P^{(n)} = aP^n$$

2.3 PageGather 알고리즘

PageGather 알고리즘은 페이지들의 상호참조 빈도수를 기반으로 하여, 유사한 페이지들을 클러스터링 하는

기법이며, 이 알고리즘은 아래의 세 가지 기본 단계를 거친다.

① 방문한 사용자의 접근 로그를 처리한다.
② 페이지들간의 상호참조 빈도수를 계산하고 이를 기반으로 유사행렬을 생성한다.
③ 위의 행렬과 관련 있는 그래프를 생성하고 이 그래프에서 clique(cluster)를 찾아낸다.

PageGather 알고리즘은 웹문서를 클러스터링 할 경우 K-Means 알고리즘이나 HAC알고리즘에 비해서, 매우 빠른 수행속도를 보인다. 따라서, 웹 구조상에 존재하는 많은 수의 문서들을 클러스터링 할 경우 사용할 수 있는 적절한 알고리즘이라고 할 수 있겠다[5].

3. Stochastic Process 모델을 이용한 웹 페이지 추천

본 논문에서는 DTMC에 의거한, 확률적 모델을 적용하여 임의의 n번의 페이지 참조 후에 사용자가 방문할 가능성이 가장 큰 페이지를 추천한다. 그러나, 사이트를 구성하고 있는 웹 페이지의 수가 많아질수록 상태전이행렬(state transition matrix)의 연산은 큰 오버헤드를 동반하게 된다. 이러한 오버헤드를 감소시키기 위하여 PageRank 알고리즘을 사용하여 사이트를 구성하고 있는 웹 페이지 간에 유사성이 큰 것들을 클러스터링(clustering)하고, 상태전이행렬을 구성하는 총 state의 수를 감소시켜 그 오버헤드를 줄인다. 이 시스템의 구조는 아래 그림과 같다.

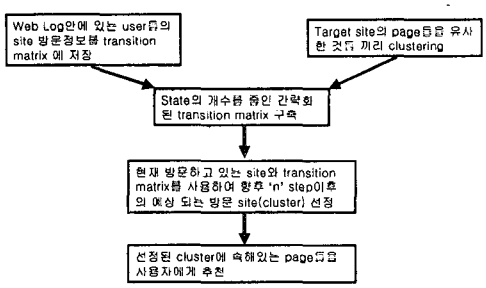
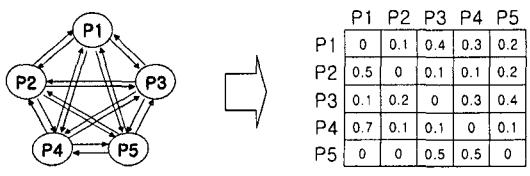


그림 1 시스템 구조

① 웹 로그 안에 있는 사용자들의 사이트 방문정보를 상태전이 행렬에 저장 : 웹 로그에 저장되어있는 클라이언트 IP, 시간, 요구되어진 페이지를 분석하여 각 페이지들간의 전이확률을 구한 후 상태전이행렬을 구축한다. 그림 2는 간단한 웹 사이트에서 페이지들 간의 상태전이행렬을 구축한 예이다.



	P1	P2	P3	P4	P5
P1	0	0.1	0.4	0.3	0.2
P2	0.5	0	0.1	0.1	0.2
P3	0.1	0.2	0	0.3	0.4
P4	0.7	0.1	0.1	0	0.1
P5	0	0	0.5	0.5	0

그림 2 페이지간의 상태전이행렬 구축

② 사이트의 페이지들을 유사한 것끼리 클러스터링 : PageGather 알고리즘을 사용하여 상호참조 확률이 임계치를 초과하는 페이지들끼리 클러스터링 한다. 아래 그림은 PageGather 알고리즘의 적용 예이다.

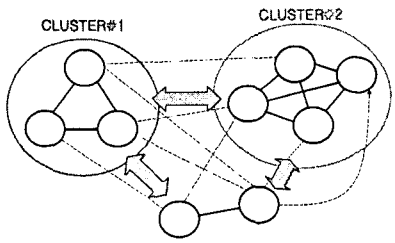


그림 3 PageGather 알고리즘을 사용한 클러스터링

③ State의 개수를 줄인 간략화 된 상태전이행렬 구축 : ②에서 생성한 페이지들의 클러스터들을 적용한 상태전이행렬을 생성한다. 그림 4는 state의 개수가 감소된 상태전이행렬을 생성한 예를 보여준다.

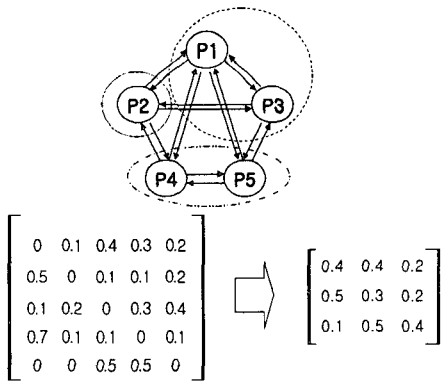


그림 4 State의 개수가 감소된 상태전이행렬

④ 현재 방문하고 있는 사이트와 상태전이행렬을 사용하여 향후 'n' 단계 이후 방문이 예상 되어지는 클러스터를 선정 : 2.2절에서 언급한 식($a^{(n)} = a^{(0)}P^{(n)} = aP^{n)}$)을 사용하여 현재 사용자가 참조하는 페이지가 속해있는 클러스터 a(initial distribution)를 기준으로 n-step이후 참조할 가능성이 가장 큰 페이지의 클러스터(marginal distribution of $X_{n)}$를 예측한다. 현재 3번 페이지를 참조할 경우 initial distribution 'a'는 벡터 (0, 0, 1) 이며, n step이후의 사용자가 방문할 것으로 예상되는 페이지의 클러스터는 아래 식의 결과로 나온 벡터의 element중에서 가장 큰 값을 가진 것의 index이다.

$$(0,\ 0,\ 1) \quad \times \quad \begin{bmatrix} 0.4 & 0.4 & 0.2 \\ 0.5 & 0.3 & 0.2 \\ 0.1 & 0.5 & 0.4 \end{bmatrix}^{n}$$

⑤ 선정된 클러스터에 속해있는 페이지들을 사용자에게 추천 : ④에서 선정한 클러스터에 속해있는 페이지들의 링크를 사용자가 현재 참조하고 있는 페이지에 동적으로 첨부함으로써 본 시스템이 추천하는 사용자의 최종 목표 페이지들을 보여준다.

4. 실험결과

본 논문의 실험에서 사용한 웹 로그는 가격비교사이트(www.bb.co.kr)에서 추출한 것이다. 이 사이트는 시간당 약 2만8천건의 페이지 요청을 받는다. 처음 5시간동안의 웹 로그를 training 데이터로 사용하였으며 이후 5시간동안의 웹 로그를 test데이터로 사용하였다. 이 웹 로그 분석시 총 2032개의 페이지가 발견되었다. 실험에서는, 2032개의 페이지를 사용해서 상태전이행렬을 구성하였으며, 임의로 선택된 200개의 페이지가 사용자가 현재 방문하고 있는 페이지 a로써 사용되었다. 사용자가 페이지 a를 참조하고 있을 때 본 논문에서 제시한 시스템이 n -step이후에 사용자가 참조할 클러스터로써 c를 추천하였을 경우, test데이터의 로그들 중에서 페이지 a를 참조한 세션이 최종 목표 페이지로 클러스터 c안에 있는 페이지를 참조할 확률을 accuracy로써 측정하였다. 아래의 표는 본 실험에서 측정된 각 step별 accuracy를 측정한 결과이다.

표1. Step별 추천한 페이지의 accuracy

Step	1-step	5-step	10-step
Accuracy	35.1%	42.5%	24.5%

위의 표를 살펴보면, 단순히 바로 다음 페이지를 볼 확률(1-step)만 적용한 페이지 추천의 accuracy 보다는 일정 step 특히, 5-step 이후에 참조할 페이지를 추천한 accuracy가 더 높은 것을 알 수 있다. 또한, 불필요하게 많은 step 이후에 참조할 페이지를 추천한 결과는 오히려 accuracy가 감소하는 것을 확인할 수 있다.

다음, 앞의 실험에서 사용한 200개의 페이지 중에서 accuracy가 높은 상위 10%의 페이지들과 accuracy가 낮은 하위 10%의 페이지들을 대상으로, 원래의 상태전이행렬과 클러스터링을 하여 state의 개수를 감소시킨 상태전이행렬로 5-step이후의 페이지를 예측하였을 경우, 걸리는 시간 및 accuracy의 차이를 측정하였다. 아래의 표에서 이 실험의 결과를 보여준다.

표2. 클러스터링 이전/이후에 추천한 페이지에 대한 성능 비교

	클러스터링이전 (2032 state)	클러스터링이후 (619 state)
RunTime	15분32초	6초
Accuracy	48.9%	43.2%

측정결과, 하나의 페이지를 추천하는데 걸리는 시간은 클러스터링 이후 비약적으로 감소하였으며, Accuracy의 감소는 상대적으로 적었던 것을 알 수 있다.

5. 결론 및 향후연구

본 논문에서는, DTMC에 의거한 확률적 모델을 적용하여 임의의 n번째 페이지 참조 후에 사용자가 방문할 가능성이 가장 큰 페이지들을 추천하는 형태의 적응형 웹사이트 구축을 위한 방법론을 제시하였다. 실험에서, 바로 다음 페이지를 추천하는 것보다는 본 논문에서 제안한 일정 step이후에 예상되어지는 페이지를 추천하는 방식이 사용자가 방문할 최종페이지를 예측하는데 있어서 더 향상된 결과를 가져왔다. 또한, 상태전이행렬의 연산에 의한 오버헤드를 줄이기 위하여 PageGather 알고리즘을 사용하여 사이트를 구성하고 있는 웹 페이지간의 유사성이 큰 것들을 클러스터링(clustering)하였고 실제로 이를 통해 비약적인 실행시간 감소의 효과를 얻을 수 있었다. 그러나, 링크구조가 단순하거나, 계층적 웹 구조 상에서 다수의 terminal 노드에 해당하는 페이지들을 가지고 있는 사이트에서는 이러한 PageGather 알고리즘을 적용했을 경우, 클러스터의 크기가 작아져서, state의 수를 효과적으로 감소시키지 못하였다. 다른 기존의 클러스터링 기법을 사용할 경우, 다수의 페이지들을 클러스터링 하는데 있어서 적지 않은 오버헤드가 예상된다. 따라서, 다양한 구조의 웹사이트에 적용 가능하면서도, 오버헤드가 적은 효과적인 클러스터링 기법의 개발은 본 논문이 제시하는 시스템을 광범위한 사이트에 적용하기 위한 우선적인 필요조건이 될 것이다.

참 고 문 헌

[1] Mike Perkowitz and Oren Etzioni, "Adaptive Web Sites: an AI Challenge", In Proc of the 15th International Joint Conference on Artificial Intelligence, pp. 16-21, 1997

[2] R.Agrawal and R.srikant, "Fast algorithms for mining association rules", In Proc of the 20th International Conference on Very Large DataBases (VLDB94), pp. 487-499, 1994

[3] R.Cooley, B.Mobasher and J.Srivastava, "Web Mining: Information and Pattern Discovery on the World Wide Web", In Proc of the 9th IEEE International Conference, pp. 558-567, 1997

[4] Vidyadhar G. Kulkarni, Modeling and Analysis of Stochastic Systems, Chapman & Hall, London, UK 1995

[5] Mike Perkowitz, Oren Etzioni, "Adaptive Web Sites: Automatically Synthesizing Web Pages", In Proc of the 15th national/10th conference on Artificial intelligence/Innovative applications of artificial intelligence, pp.727-732, 1998