링크내역을 이용한 페이지점수법 알고리즘
(PageRank Algorithm Using Link Context)

이 우 기 † 신 광 섭 †† 강 석 호 †††
(Wookey Lee) (Kwangsup Shin) (Sukho Kang)

요 약 웹은 정보의 저장 및 검색에 있어서 보편적인 매체가 되고 있다. 웹에서 정보 검색은 검색엔진을 출발점으로 이용하는 것이 대부분이지만, 그 결과는 사용자의 요구와 늘 일치하는 것은 아니며 때로는 의도적으로 조작된 검색 결과가 제시되기도 한다. 검색엔진의 데이타를 의도적으로 조작하는 것을 스패밍(spamming)이라고 부르며, 다양한 스패밍과 방지기술이 있지만, 최근에 각광을 받고있는 링크기반 검색 방식에는 스패밍이 쉽지 않은 것으로 알려져 있다. 그러나 이러한 방식에서도 구글폭탄(Google Bombing)과 같이 페이지점수법(PageRank)을 조작할 수 있는 약점이 있다. 본 논문에서는 이러한 약점을 방지할 수 있는 알고리즘을 제시한다. 기본적으로 링크 기반 검색 방식을 기초로 하여 웹을 하나의 유향 레이블 그래프로 인식하여 각 웹 페이지들은 하나의 노드로, 하이퍼링크는 에지로 표현함에 있어서 본 연구에서는 링크구조를 기반으로 링크내역(link context)을 부여하고 이를 에지의 레이블로 사용한다. 링크내역과 대상 페이지 사이의 유사도를 구하고, 이것을 이용하여 페이지점수법의 인접행렬을 재구성하는 방법을 취했다. 결과로써 기존의 방법 및 특이값 추출기법(SVD)에 기반한 새로운 기준을 도입해 그 효과를 입증했다.

키워드 : 스패밍, 페이지점수법, 링크내역, 특이값 추출기법

Abstract The World Wide Web has become an entrenched global medium for storing and searching information. Most people begin at a Web search engine to find information, but the user's pertinent search results are often greatly diluted by irrelevant data or sometimes appear on target but still mislead the user in an unwanted direction. One of the intentional, sometimes vicious manipulations of Web databases is Web spamming as Google bombing that is based on the PageRank algorithm, one of the most famous Web structuring techniques. In this paper, we regard the Web as a directed labeled graph that Web pages represent nodes and the corresponding hyperlinks edges. In the present work, we define the label of an edge as having a link context and a similarity measure between link context and the target page. With this similarity, we can modify the transition matrix of the PageRank algorithm. A motivating example is investigated in terms of the Singular Value Decomposition with which our algorithm can outperform to filter the Web spamming pages effectively.

Key words : Web Spamming, PageRank, Link context, Singular Value Decomposition

1. 서 론

웹(WWW)은 정보저장 및 검색에 있어서 보편적인 매체가 되고 있다. 웹 정보검색은 대다수 검색엔진에 질의어를 입력하는 것으로부터 시작되며, 모든 검색엔진들은 입력된 사용자의 질의와 가장 관련성이 높은 웹 페이지를 찾아내기 위한 나름대로의 검색 알고리즘을 가지고 있다. 이러한 알고리즘은 크게 내용기반검색, 구조기반검색, 그리고 사용행태 마이닝 등으로 분류된다[1]. 내용기반검색은 전통적인 정보검색 방법에 기초하여 웹 문서의 내용으로부터 정보검색을 하는 방법으로 Yahoo, AltaVista, DMOZ, Naver등 대다수 이 방법을 사용하고 있다. 이러한 기법들의 문제는 웹 페이지의 키워드를 조작에 대해 취약하다는 점이다[4,5]. 이에 반해 구조기반 검색방법은 내용과는 무관하게 하이퍼링크 구조에 기반한 방식으로서 본 연구는 이 분류에 속한다. 사용행태 마이닝은 웹로그 기반 데이타마이닝 기법으로 웹 사용내역을 분석하며, 정보검색이 초점이 아니며 일반적으

· This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Advanced Information Technology Research Center (AITrc).
† 종신회원 : 인하대학교 산업공학부 부교수
wookeylee@gmail.com
†† 비 회 원 : LG CNS
Kwangsup@gmail.com
††† 비 회 원 : 서울대학교 산업공학과 교수
shkang@cybernet.snu.ac.kr
논문접수 : 2004년 5월 27일
심사완료 : 2006년 7월 13일

로 적용 도메인이 다르다[2,3].

웹 정보검색에서 기본적인 문제는 검색결과가 사용자의 요구와 늘 일치하지는 않으며, 심각한 경우 악의적으로 혹은 상업적으로 조작된 검색결과가 제시되기도 한다는 점이다[3]. 검색엔진에서 높은 순위를 얻기 위한 모든 조작을 스패밍(Spamming)이라 하며, 내용기반검색 기법은 내용의 스패밍에 대해 근본적으로 자유롭지 못하다. 그러므로 내용 대신 링크를 활용한 구조기반 검색기법이 대안으로 떠오르면서 구글이 현재 가장 효과적인 검색기법으로 인식되고 있다. 웹의 링크구조는 그 자체로서 중요한 정보를 포함하고 있을 뿐 아니라, 웹 페이지의 순위 평가에 효과적이다[4]. 특히 링크구조는 웹 페이지의 내용과는 무관하기 때문에, 순위 조작이 곤란하고 사용자 질의를 비교적 정확히 반영한다고 평가되고 있다[2]. 그러나 이 역시 새로운 방식의 순위 조작의 가능성이 있다. 예컨대, 2001년 처음 발생하였으며, 최근 웹 블로그의 확산으로 다시금 문제시 되고 있는 구글폭탄(Google bombing)은 페이지점수법 알고리즘의 약점을 이용한 것이다[5]. 이는 몇몇 키워드에 대해 특정한 웹페이지가 과다 링크되게 함으로써 구글검색을 조작하는 방식이다.

본 논문에서는 페이지점수법이 가진 이러한 문제점을 인식하고 링크내역을 활용한 페이지점수법 알고리즘을 제안하면서 구조기반 검색기법의 새로운 대안을 제시하고자 하는 것이다. 구체적으로 질의어와 검색엔진의 결과와의 유사성 문제에 대하여 링크기반 접근법으로 스패밍페이지의 중요도의 변화를 확인한다. 또한 구조기반 접근법이 스패밍 페이지를 여과하는데 효과적이라는 점을 입증하기 위해 기존의 페이지점수법 알고리즘을 링크내역과 대상 페이지 간의 유사도를 이용하여 제시한 것이다.

논문의 구성은 다음과 같다. 2장에서는 구조기반 검색기법과 링크정보에 대해 살펴보며, 3장에서는 본 논문에서 중점적으로 다루고 해결하고자 하는 웹의 구조화 문제점에 대해 설명하고 본 연구의 목적을 제시한다. 4장에서는 문제해결을 위한 알고리즘을 제안하고, 5장에서는 모델링과 예제를 통해 알고리즘을 설명하고, 대안들을 비교한다. 마지막으로 6장에서 결론과 향후 연구 방향에 대해 언급한다.

2. 구조기반 검색기법

구조기반 검색기법들은 웹의 링크구조에 기반하여 동적체계론 등 이론적으로 매우 안정적인 모델들이 준용되고 있다. 이러한 모델들은 하이퍼링크를 기반으로 웹 페이지들을 분석 및 분류하며, 웹 사이트 간 유사도와 같은 정보를 생성하는 것이다. 여기에는 키워드 검색뿐만 아니라 계층구조 생성, 군집화 등의 목적으로 구조를 이용하는 알고리즘이 있다[3,4]. 이러한 방식들은 페이지의 내용만을 분석하는 것보다 더 효과적인 평가기준을 제공한다. 최근 등장한 웹 구조화 기법의 경우 링크에 의미적 가중치를 부여하여 계층적 구조를 추출하는 알고리즘도 있다[4]. 그러나 이 경우 가중치 자체가 각 웹 페이지로부터 유도된 값이므로 스패밍 페이지의 값이 링크에 그대로 전달된다는 약점이 개선되지 못했다. 이와는 달리 웹의 전역적 수준에서 링크구조를 분석하는 알고리즘은 스패밍페이지에 강한 면모를 보인다[5,6].

링크 만을 기반으로 하는 검색 방식의 대표적인 예로 HITS(hypertext induced topic selection)와 페이지점수법이 있다[7]. HITS의 기본 개념은 웹의 하위 그래프를 정의하고, 이를 대상으로 주어진 질의어에 대해 중요 페이지(authority page)와 연결 페이지(hub page)로 나누는 과정에서 링크분석을 수행한다. 한 하위 그래프는 매우 많은 웹 페이지를 가지며 웹그래프에서 링크분석에 사용될 뿐만 아니라 다음 단계에서의 계산량을 줄여주는 역할을 한다[4,7]. 그러나 HITS알고리즘은 다음과 같은 두 가지 약점이 알려져 있는데, 알고리즘 진행방향의 유일성이 보장되지 못하는 성질(non-uniqueness)과 가중치 부여에서의 문제점(nil-weighting) 등이 그것이다[8,9].

페이지점수법의 기본 개념은 다음과 같다. 웹그래프 $G(N,A)$에서 N은 페이지 집합, A는 이들간의 링크 집합이다. 이때 웹페이지 u에서 v로의 링크$(u,v)\in A$는 중요도(R)를 전달하는 역할을 한다. 식(1)의 전개를 통해 전체 웹페이지에 대한 중요도 벡터의 생성으로 확장할 수 있다. 만약 페이지의 수는 n이고, N_u는 웹페이지 u에 존재하는 출력링크수, 모든 페이지의 초기 값을 $1/n$로 설정한다. 이때 B_v는 v를 가리키고 있는 페이지의 집합을 의미한다.

$$\forall v, R^{(i+1)}(v) = \sum_{w \in B_v} R^{(i)}(u)/N_u \tag{1}$$

위 식은 중요도 벡터가 역치값(threshold) 내에 수렴될 때까지 반복 연산 된다[9]. 그 수렴을 보장하기 위해 다음 식(2)처럼 감쇠요소(damping factor) d를 사용한다(이때, $0 \le d \le 1$).

$$R^{(i+1)}(u) = (1-d)\times[1/n]_{n\times 1} + d \times \sum_{\substack{(i,j)\in A \\ j\in B_v}} R^{(i)}(v)/N_v \tag{2}$$

구글에서는 일반적으로 감쇠요소의 값으로 0.85를 사용하며[10], 다른 연구[9,11]에서는 페이지점수법 벡터가 실수 값으로 수렴한다는 점과 또한 감쇠요소 크기를 조정하면 페이지점수법 값이 수렴하는 속도를 통제할 수 있다는 점을 증명하였다.

3. 링크정보를 활용한 웹의 구조화

3.1 문제의 정의와 목표

스패밍은 검색엔진에서 높은 순위를 얻기 위한 모든 조작행위로 정의된다. 스패밍에는 웹페이지에 인위적인 조작을 가하는데 다음과 같은 방식이 있다. 키워드 반복, 관련없는 용어삽입, 작거나 보이지 않는 글자, cloaking, 거짓 제목 및 태그, 또한 over-submitting, Swamping, Meta-spamming, Spoofing, Squatting, Slandering, Scamming 그리고 도메인 스패밍 등이 있다[13,14].

이러한 스패밍 전략은 크게 두 가지로 분류할 수 있다. 하나는 웹 페이지의 내용을 조작하는 방법이고, 다른 하나는 링크정보를 조작하는 방법이다. 1장에서 이미 구조기반 기법이 내용기반기법에 비해 스패밍 페이지를 여과하는데 더욱 효과적임을 언급했다. 페이지점수법 알고리즘은 각 링크들이 원천 페이지로부터 대상 페이지에 동일한 비율로 중요도를 전달한다는 것이 기본 개념이다. 그러나 이와 같이 링크만을 수치화하게 되면 링크와 대상 페이지간의 관련성이 희박해지는 문제가 발생한다. 이것은 하이퍼링크 자체가 가지는 의미를 무시한 것으로 구글폭탄은 이를 역이용한 대표적 방식이다.

본 연구의 목표는 스패밍 페이지 및 링크를 다수 생성하여 순위를 조작하는 웹 구조화 스패밍을 효과적으로 검출하는 것이며, 다른 말로는 정확한 검색이 가능케 되는 구조화 웹 검색 방식을 제시하는 것이다. 다음 절에서 설명하는 링크내역에 기초하여 스패밍을 효과적으로 검출하되, 알고리즘을 내역기반으로 수정하여 스패밍 페이지가 검색엔진에서 높은 순위를 갖지 못하도록 하는 것이며, 이를 검증하는 것이다.

3.2 링크내역 방식

웹을 일종의 유향 레이블 그래프로 인식하면 각 웹 페이지를 하나의 노드로, 웹 페이지 사이의 하이퍼텍스트 링크가 에지를 나타내는 것으로 볼 수 있다. 이때 각 노드는 웹 페이지가 가지는 URL로 정의되며, 링크는 방향을 가진다. 본 논문에서는 링크내역을 정의하고 이를 에지 레이블로 사용한다. 지금까지 정보검색에 있어서 내용기반 검색은 물론 구조기반 검색에서도 링크내역은 대다수 무시되어 왔다[4].

링크내역이란 다음과 같이 정의된다: Link keyword . 이때 링크키워드(Link keyword)는 대상 페이지에 대한 설명을 담고 있는데, 대부분 충분한 설명이라기 보다는 제목 정도에 그치고 있기 때문에 이것만을 링크내역으로 이용하기에는 충분치 않다. 따라서 본 연구에서는 강건한 링크(robust hyperlink)의 사전적 날인방식(lexical signature)개념을 도입하여 이러한 링크키워드의 의미를 확장한다[12]. 그 초점은 링크의 역할이 웹 페이지들 간의 단순한 연결이라는 점 외에, 링크 자체에도 의미를 부여할 수 있으며, 이러한 링크에서 키워드를 추출하는 방식을 준용하자는 것이다[3,4,11]. 강건한 링크[12]란 대상 페이지를 효과적으로 표현하는 정보를 담고 있는 링크로서 그 정보는 다섯 개 이내의 단어로 축약될 수 있다는 것이며, 해당 연구에서는 이를 실증적으로 입증했다. 물론 이러한 숫자에 본 연구가 제한을 받지는 않지만 제5절 예제에서 이를 적용하여 보았다. 해당 논문에서 표현한 사전적 날인방식에서 링크는 다음과 같은 주요어의 집합 형식으로 표현될 수 있다.

 click here

3.3 특이값 추출기법(Singular Value Decomposition)

내역기반 페이지점수법을 이용하여 중요도가 낮은 페이지를 스패밍페이지로 정의하고자 한다면 그 기준을 어느 수준에서 결정할 것인가가 중요한 문제로 부각된다. 본 연구에서는 이러한 문제에 대하여 특이값 추출기법(SVD: Singular Value Decomposition)을 적용하여 평가하고자 한다. SVD는 주어진 하나의 행렬을 분해하여 수리적 해답을 얻는 과정인데, 인접행렬M을 동적시스템의 전이행렬(transition matrix)로 인식하여 식(3)과 같이 U, V 그리고 Σ행렬로 각각 분해한 다음 얻어지는 고유치를 이용하여 스패밍페이지 여부를 결정할 수 있다[16]. 웹에서의 인접행렬은 특정 페이지에서 다른 페이지로의 링크를 나타내는 것이므로 시간에 따른 전개를 전이행렬로 인식할 수 있다. 이때 주목할 부분은 부행렬 Σ의 구성 요소인 σ_i가 고유치(eigenvalue)의 정렬된 값으로 표현된다는 점이다. 이 방법은 고유치의 숫자를 임의로 결정하지 않고 다음과 같이 일정한 오차 범위를 가지고 분석할 수 있다는 장점이 있으며, 물론 인접행렬이 가역적(nonsingular)이지 않으면 적용할 수 없다[15,17].

$$M \approx U \Sigma V^T \tag{3}$$

이때 U와 V는 각각 $m \times m$ 및 $n \times n$ 직교행렬이며, Σ은 $m \times m$ 특이값(σ_i)의 대각행렬이다. 이때 오차의 범위는 낮은 값을 근사화시키면서 확인할 수 있는데, Σ행렬에서 (행렬의 구성요소의 수치 m보다 작고 0보다 큰) k개의 최대값만을 취하고, 그것을 제외한 나머지 요소를 모두 0으로 대체한 근사행렬(M_k)과의 차이를 다음 식(4)와 같이 Frobenius norm을 통해 구한다[15].

$$\|M - M_k\|_F = \min_{\text{rank}(B) \le k} \|M - B\|_F = \sqrt{\sigma_{k+1}^2 + \sigma_{k+2}^2 + \cdots + \sigma_{rA}^2} \tag{4}$$

3.4 스패밍 페이지의 형식적 정의(Formal definition)

일반적인 Frobenius norm값은 선택되지 않은 고유치의 합으로 계산되며, 낮은순위근사(low rank approximation)라는 것은 행렬에서 고유치가 큰 값부터 정렬되

므로 낮은 특이값 중에서 용납할 수 있는 오차만큼을 안고 근사를 수행한다는 의미이다. 이 요소가 웹에 적용될 때에는 대상 웹의 인접행렬 구조에서 해당 노드를 제외할 경우에 발행하는 오차를 의미하는 것이다. 본 연구에서는 이 요소를 스패밍페이지를 나타낸다고 정의하고, 다음 식(5)로써 표현한다. 다시 말해서 이 요소는 해당 노드의 특이값 크기라 볼 수 있으며, 즉, 전체 노드를 표현한 행렬에서 특정 노드(k)를 제외할 때의 특이값의 차이란 결국 제외한 특정 노드의 특이값이 된다는 것이다.

$$\xi_k = 1 - \| M_k \|_F / \| M \|_F \quad (5)$$

이것은 또한 근사 오차의 의미로 확장하여 사용될 수 있으며, 그것은 전체 오차 합계에서 정규화한 상대 오차로 인식될 수 있다.

4. 링크내역기반 페이지점수법 알고리즘

4.1 알고리즘

알고리즘은 전체적으로 링크정보의 입력을 받고(1~5행), 링크의 가중치를 정규화하고(6행), 수렴 조건을 설정하고(8행), 페이지 점수를 계산함에 있어서(9~12행) 종료 조건에 도달할 때까지 반복하는 절차이다. 가중치의 정규화는 Rank벡터 성분의 합이 일정한 값으로 수렴되도록 하는 것으로, 한 행의 합이 1이 되도록 하기 위해서 각 가중치의 합을 나눈다.

수렴 조건은 인접행렬 M이 기약적(irreducible) 및 비주기적이어야 한다[17]. 기약적이란 그래프가 강력 연결되어 하나의 노드에서 모든 노드로 링크를 따라 도달할 수 있어야 한다는 의미이며, 비주기적이라는 것은 실제 웹이 가지고 있는 중요한 특징의 하나이다. 만일 이를 만족시키지 못하는 특수한 경우에 대비하기 위해서 다음 2가지 조치를 취한다. 첫째, 출력 링크 수가 0인 에지를 가진 노드에는 출력 링크의 완전 집합을 포함시킨다. 둘째, 감쇠지수를 사용하여 1/n의 중요도를 전체 노드로 확대시켜도 인접행렬의 수렴하는 성질을 만족시킬 수 있다[9]. 따라서 식(6)을 이용해 만들어진 행렬 D를 전이 행렬 M에 더한다.

원천페이지와 대상페이지와의 유사도는 대상페이지 내의 키워드 집합과 링크내역과의 유사도로써 다음 식을 이용하여 계산하며, 그 중 최대값을 이용한다. 링크내역과 문장간의 유사도는 쟈카드계수(J: Jaccard)를 사용하며, 다음 식(6)으로 계산한다. 이때 $Keyword_{ij}$는 링크내역(즉, i번째 원천 페이지로부터 j번째 대상 페이지로의 링크), 그리고 S_{tl}은 대상 페이지 내의 l번째 문장을 의미한다.

$$J(Keyword_{ij}, St_k) = \frac{| Keyword_{ij} \cap St_k |}{| Keyword_{ij} \cup St_k |} \quad (6)$$

대상페이지와 링크내역의 유사도는 다음 식(7)처럼 최대값을 사용한다.

$$Sim(Keyword_{ij}, Page_j) = \max_k \{J(Keyword_{ij}, St_k)\} \quad (7)$$

최근 여러 웹 검색엔진에서 흔히 사용되는 *tf-idf*(term frequency and inverse document frequency) 방식처럼 대상 페이지 내의 전체 내용을 대상으로 적용한다면, 대상 페이지의 길이나 페이지 내의 용어들에 의해 큰 영향을 받게 된다[1,14]. 이러한 영향을 최소화하기 위해 문장 단위로 유사도를 측정한다. 질의어와 문장 단위의 유사도 측정이 검색 성능 향상에 입증되기도 하였다[17]. 그러나 링크내역과는 관련 정도가 높지만 스패밍 페이지로 인식될 가능성은 여전히 존재한다.

마지막으로 알고리즘의 종료 조건을 설정하는 데는 크게 두 가지 방법이 사용되는데, 하나는 사전에 결정된 횟수만큼 반복하는 것이고, 다른 하나는 일종의 역치값을 설정하고 i번째 벡터와 (i+1)번째 벡터의 차이가 그 역치값 이하일 때 반복을 종료하는 것이다. 본 논문에서는 후자를 사용한다.

```
1.  for i {
2.      for j {
3.          if   w_ij ← sim(Keyword_ij, Page_j)
4.          else  w_ij ← 0
5.      }
6.      w'_ij  ← normalize(w_ij)
7.  }
8.      M ←[w'_ij]+D  /* transition matrix */
9.  while (Rank vector converges){
10.         for (i=1 to n){
11.  R(i) = (1-d)×[1/n]_n×1 + d×Σ_{j∈B_i} m_ji * R(i)
12.     }
13. }
```

용어 설명

$Page_j$: 웹페이지j

$Keyword_{ij}$ = 링크 키워드

w_{ij} : 가중치 값

d: 감쇠요소

M: 전이행렬

$R(i)$: 웹페이지i 의 가중치

그림 1 내역기반 페이지점수법 알고리즘

5. 예제 및 분석 결과

5.1 예제

그림2는 제안된 알고리즘이 어떻게 스패밍 링크를 찾아내고, 스패밍 페이지의 중요도를 어떻게 줄여나가는가를 확인하기 위한 예제이다. 그래프는 다음과 같은 4개의 웹 페이지와 7개의 링크로 이루어져 있다.

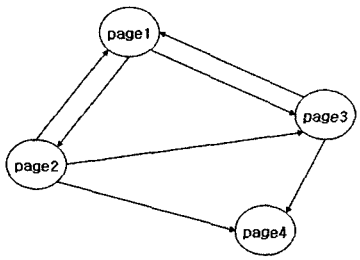

그림 2 예제 그래프

표1은 예제의 각 웹페이지를 설명하는 것으로, 페이지의 제목, 링크 그리고 초기 Rank값을 포함한다. 특히, 페이지4는 다른 페이지로의 링크가 없는 반면, 페이지2는 그래프 내의 모든 페이지를 가리키는 3개의 링크를 가지고 있다. 초기 Rank값은 모두 동일하게 1로 두어, 초기값에 의해 영향을 받는 것을 방지하였다.

표 1 예제 웹 페이지의 세부 사항과 초기값

Page	Title	Link to	Initial Rank
1	Introduction to DB	Page2 and Page3	1
2	Java	All	1
3	JSP	Page1 and Page4	1
4	Bibliography of Bush	None	1

5.2 페이지점수법과 내역기반 페이지점수법

표2는 7개의 링크정보를 나타내는 것으로 링크키워드와 사전적 날인방식, 그리고 이 두 가지로 만들어진 링크내역과 대상 페이지간의 유사도를 측정한 가중치 값이 주어져 있다.

가중치 값을 식(6) 및 정규화함으로써 다음과 같은 전이 행렬을 얻을 수 있다.

$$M^T = \begin{bmatrix} 0 & 0.60274 & 0.971831 & 0.25 \\ 0.474453 & 0 & 0 & 0.25 \\ 0.525547 & 0.383562 & 0 & 0.25 \\ 0 & 0.012699 & 0.028169 & 0.25 \end{bmatrix}$$

본 알고리즘 및 페이지점수법과 SVD의 결과를 Matlab6.1을 이용하여 얻었다. 그림3은 본 알고리즘의 수행 결과를 나타낸다. 그림에서와 같이 연산을 진행함에 따라 개별 노드는 특정 값으로 수렴함을 확인할 수 있다. 표3에서는 기존 방식과 비교하였다. 각 페이지의 중요도는 알고리즘이 수렴할 때의 값이며, 비율은 전체 중요도에 대한 상대적 크기이다.

기존의 페이지점수법 알고리즘에서 페이지4는 전체

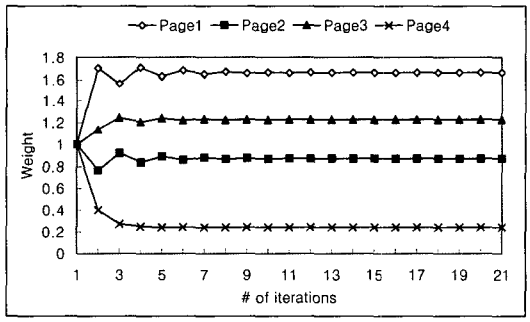

그림 3 내역기반 페이지점수법에 따른 수렴 노드값

표 2 링크 상세 사항

(u,v)	Link keyword	Lexical signature	Weight
(1,2)	Java example	Java, class, sun, interpreter, API	0.65
(1,3)	JSP example	Servlet, Web, DB, server, compile	0.72
(2,1)	JDBC	DB, table, driver, query, result	0.88
(2,3)	Java Server Page	class, HTML, useBean, server, Servlet	0.56
(2,4)	Miserable failure	Bush, war, defeat, Iraq, invasion	0.02
(3,1)	query example	SQL, query, select, where, result	0.69
(3,4)	Miserable failure	Bush, war, defeat, Iraq, invasion	0.02

표 3 내역기반 페이지점수법과 페이지점수법 방법의 결과 비교

	Page 1	Page 2	Page 3	Page 4	Sum
페이지점수법	1.260893	0.953819	1.34754	1.260893	4.823146
	26.1%	19.8%	28%	26.1%	100%
내역기반 페이지점수법	1.661116	0.871044	1.227172	0.240667	4
	41.5%	21.8%	30.7%	6%	100%

Rank값 중 페이지1과 같은 26.1%를 차지하고 있었으나, 수정된 알고리즘에서는 스패밍페이지인 페이지4의 Rank값을 감소시켜 결국 전체 Rank 중 6%만을 차지하게 되었다. 반면 페이지1의 중요도는 41.5%로 증가하였으며, 나머지 두 페이지 역시 비중이 페이지점수법에 비해 각각 2%정도 증가하였다. 앞서 스패밍 페이지로 정의했던 페이지4의 비중이 크게 감소하였기 때문에 웹 페이지 검색 결과에서 페이지4가 높은 순위를 나타낼 가능성이 크게 감소하게 된다.

5.3 내역기반 페이지점수법과 SVD

다음으로 SVD와 내역기반 페이지점수법과의 결과를 비교하고자 한다. SVD에 이용할 전이 행렬은 앞 절에서 계산한 결과를 그대로 이용한다. Matlab의 SVD 함수를 이용하여 다음과 같은 결과를 얻었다.

$$U = \begin{bmatrix} -0.9332 & 0.3415 & -0.0361 & 0.1058 \\ -0.1363 & -0.6119 & -0.4918 & 0.6043 \\ -0.3212 & -0.7072 & 0.5025 & -0.3796 \\ 0.0855 & -0.0941 & -0.7102 & -0.6925 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 1.2204 & 0 & 0 & 0 \\ 0 & 0.7892 & 0 & 0 \\ 0 & 0 & 0.2526 & 0 \\ 0 & 0 & 0 & 0.1763 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.1913 & -0.8388 & 0.1217 & 0.4950 \\ -0.5628 & -0.0846 & 0.6386 & -0.5180 \\ -0.7451 & 0.4171 & -0.2181 & 0.4725 \\ -0.3024 & -0.3395 & -0.7279 & -0.5133 \end{bmatrix}$$

Σ행렬의 대각 성분은 특이값을 의미한다. 전이 행렬의 각 행을 하나씩 제거한 후, Frobenius norm을 계산해보면 각 페이지의 중요도를 상대적으로 계산할 수 있으며, 이를 통해 근사오차(ξ_k) 즉, 앞서 언급한 식(5)에서 언급한 스패밍 값을 알 수 있으며 전체오차에서 차지하는 상대적인 비율 즉, 상대오차를 유도하여, 다음 표4에서는 그 결과를 보여준다.

Σ행렬의 마지막 특이값을 제거하였을 때의 오차 값은 0.1696341이며, 전체에 대한 상대적 오차는 약 9%이다. 위의 결과로부터, 페이지4가 가지는 특이값의 전체 값에 대한 비중이 극히 작다는 것을 알 수 있다. 이는 앞서 언급한 내역기반 페이지점수법 알고리즘의 결과와 유사한 내용이다. 다른 의미로는 페이지 4를 제거하였을 경우, 약 9%의 정보 손실이 발생한다는 것을 의미한다. 그러나 본 알고리즘의 결과에서 페이지4는 전체의 약 6%정도의 중요도를 가지기 때문에 정보 손실에 있어서 SVD의 결과에 비해 본 연구에서 제안하는 방법이 더욱 효과적임을 알 수 있다. 결국 본 연구에서 제시된 내역기반 구조적 페이지점수법 알고리즘이 스패밍페이지의 중요도를 명확히 감소시킴으로써 기존의 페이지점수법 알고리즘의 성능을 현저하게 개선할 수 있음을 알 수 있다.

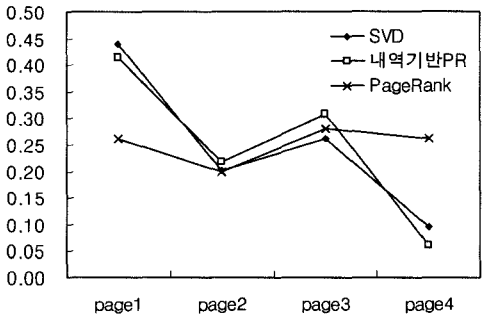

그림 4 내역기반 페이지점수법 및 기타 방식과의 비교

이를 SVD방식 및 PageRank방법과 함께 본 연구의 링크 내역기반PR을 비교한 것이 그림4에 제시되어있다. 그림에서 기존의 구글 페이지점수법은 모든 페이지에 대해 비슷한 값을 주기 때문에 스패밍페이지에 대해 분별력이 현저히 떨어지는 반면, 내역기반 페이지점수법 및 이의 검증을 위해 도입한 SVD방식은 검출력이 분명하다는 것을 확인할 수 있다. 그러므로 기존의 페이지점수법에 기반한 검색엔진에서는 스패밍페이지에 의해 결과가 오도되기 쉬운 취약성이 있지만, 본 연구에서 제시한 방식은 검색결과에서 상대적으로 매우 검출력이 우월하다는 것을 알 수 있다.

6. 결론 및 향후 연구

본 연구에서는 우선, 웹 정보검색의 내용에 기반한 스패밍은 근본적으로 그 내용의 조작으로부터 자유로울

표 4 특이행렬분석 및 근사화 오차, 상대오차

	None	1	2	3	4
$\sum\sigma_i^2$	2.2071013	0.8369691	1.9194069	1.7212030	2.1435836
Frobenius norm		1.170526	0.5363715	0.6970641	0.2520272
ξ_k		0.7878979	0.3610394	0.4692039	0.1696431
상대오차		0.440712	0.201948	0.26245	0.09489

수 없다는 점과 구조기반 접근법이 스패밍 페이지를 여과하는데 효과적이라는 점을 밝혔다. 또한 여기서 제안한 새로운 내역기반 페이지점수법 알고리즘은 검색엔진에 있어 가장 핵심적인 사안의 하나인 사용자 질의어와 검색 엔진의 결과와의 유사성 문제를 링크기반으로 해결하고자 하였다. 본 연구의 초점은 검색엔진이 가진 약점을 극명하게 드러낼 수 있는 스패밍 페이지의 인식과 중요도의 변화를 통한 그 해결 여부에 두었다. 따라서 우리는 기존의 페이지점수법 알고리즘을 링크내역과 대상 페이지 간의 유사도를 이용하여 새로이 내역기반 페이지점수법 알고리즘으로 확장한 것이다.

예제를 통해 내역기반 구조적 페이지점수법이 기존의 구조기반 알고리즘의 약점을 보완하고 성능을 개선할 수 있음을 보여주었으며, SVD의 결과와의 비교를 통해 스패밍 페이지 탐색 효용성을 확인할 수 있었다. 결론적으로 본 연구에서 제안하는 내역기반 구조적 페이지점수법 알고리즘을 사용할 경우, 구글폭탄과 같은 스패밍 페이지가 높은 순위를 차지하게 될 확률이 줄어들 뿐 아니라, 각 페이지와 링크가 가지는 의미를 충분히 반영한 새로운 검색엔진이 가능하다는 전망을 할 수 있게 된다.

참 고 문 헌

[1] Kowalski, G. and Maybury, M. Information Storage and Retrieval Systems, Kluwer Pub. 2000.

[2] Kosala, R. and Blockeel, H., "Web mining Research: A Survey," ACM SIGKDD, Vol.2, pp.1-15, 2000.

[3] Arasu, A., Cho, J., Garcia-Molina, H., Paepcke, A. and Rachavan, S., "Searching the Web," ACM Trans. Internet Technology, Vol.1, No.1, pp.2-43, 2001.

[4] Halkida, M., Nguyen, B., Varlamis, I. and Vazirgiannis, M., "THESUS: Organizing Web document collections based on link semantics," The VLDB Journal, Vol. 12, pp.320-332, 2003.

[5] Gyöngyi, Z., Garcia-Molina, H. and Pedersen, J., "Combating Web Spam with TrustRank," VLDB, pp.576-587, 2004.

[6] Wookey, L. and Geller, J., "Semantic Hierarchical Abstraction of Web Site Structures for Web Searchers," Journal of Research and Practice in Information Technology, Vol. 36, No. 1, pp.71-82, 2004.

[7] Gibson, D. and Kleinberg, J., Raghavan, P., "Clustering Categorical Data: An Approach Based on Dynamical Systems," The VLDB Journal, Vol.8, No.3-4, pp.222-236, 2000.

[8] Miller, J., Rae, G. and Schaefer, F., "Modifications of Kleinberg's HITS Algorithm Using Matrix Exponentiation and Web Log Records," ACM SIGIR, pp.444-445, 2001.

[9] Taher, H. and Haveliwala, T., "Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search," IEEE TKDE, Vol. 15, No. 4, pp.784-796, 2003.

[10] Eiron, N., McCurley, K. and Tomlin, J., "Ranking the Web Frontier," WWW, pp. 309-318, 2004.

[11] Novak, J., Raghavan, P. and Tomkins, A., "Anti-aliasing on the web," In Proc. WWW, pp. 30-39, 2004.

[12] Phelps, T. and Wilensky, R., "Robust Hyperlinks: Cheap, Everywhere, Now," Digital Documents and Electronic Publishing, LNCS 2023, pp. 28-43, 2000.

[13] Hinde, S., "Smurfing, Swamping, Spamming, Spoofing, Squatting, Slandering, Surfing, Scamming and Other Mischiefs of the World Wide Web," Computers & Security, Vol. 19, No. 4, pp.312-320, 2000.

[14] Goth, G., "Much Ado About Spamming," IEEE Internet Computing, Vol. 7, No. 4, pp.7-9, 2003.

[15] Papadopoulo, T. and Lourakis, M., "Estimating the Jacobian of the Singular Value Decomposition: Theory and Applications," ECCV (1), pp.554-570, 2000.

[16] Castelli, V., Thomasian, A. and Li, C., "CSVD: Clustering and Singular Value Decomposition for Approximate Similarity Search in High-Dimensional Spaces," IEEE TKDE, Vol. 15, No. 3, pp. 671-685, 2003.

이 우 기

1996년 서울대학교 산업공학과 학사, 석사, 박사. 현재 인하대학교 공과대학 교수. 2000년 카네기멜런대 MSE, 2002년 UBC 방문교수. 2004년 한국경영과학회 최우수논문상. 현재 한국경영정보학회 및 정보과학회 이사. 현재 한국ITA학회 논문지편집위원장. PC멤버: WAIM06, iiWAS06, IEEE DEST07등. 관심분야는 IR, Web Mining, DW 등

신 광 섭

2003년 서울대학교 산업공학과 학사 2006년 서울대학교 산업공학과 석사. 현재 LG CNS Entrue Consuting 전문컨설턴트. 관심분야는 IR, BPM, SOA, Web Service, SCM

강 석 호

서울대학교(학사), U of Washington(석사), Texas A&M(박사, 76'). 현재 서울대학교 공과대학 교수. 서울대학교 AIP 주임교수, 한국 O.R.학회 회장. 아시아 태평양 O.R.학회 부회장(현). 관심분야는 MIS, IR, BPM, Medical Informatics 등