링크 중요도에 기반한 웹사이트의 계층 구조화

임태수[1*], 박범환[2], 이우기[1]
1. 성결대학교 컴퓨터공학부 {tshou*, wook}@sungkyul.edu
2. 한국철도기술연구원 bhpark@krri.re.kr

Link ranking-based hierarchical structuring of web site

Taesoo Lim[1*], Bum Hwan Park[2], Wookey Lee[1]
1. Dept. of Computer Engineering, Sungkyul Univ., Anyang-8-dong, Manan-gu, Anyang, Korea
2. Dept. of Railway Policy&Logistics Research, Korea Railroad Research Ins., Uiwang-City, Korea

요 약

수많은 웹페이지들이 하이퍼링크를 통해 복잡하게 연결된 그래프 구조를 가지고 있는 웹사이트를 계층적으로 구조화하는 것은 해당 사이트를 검색하고자 할 때, 정보를 재조직화하고 고려해야 할 대안들의 개수를 감소시킨다는 점에서 매우 유용하다. 본 논문은 웹사이트의 의미론적인 계층화를 최적화하기 위하여 사용자의 순회 경로, 즉 웹아크의 중요도 합을 최대화할 수 있는 트리 구조를 생성하였다. 구체적으로 첫째 PageRank에 기반한 웹아크 중요도를 생성하였고, 둘째 Minimum-Cost Arborescence 문제를 이용하여 최적 트리 구조를 생성하였다. 사용자의 질의에 독립적으로 생성된 트리 구조는 웹사이트의 의미 있는 계층 구조로서 사용자로 하여금 해당 사이트를 보다 효과적으로 검색할 수 있도록 도와줄 것이다.

1. 서 론

월드와이드웹의 비약적인 성장은 유용한 정보를 찾고자 하는 사용자들로 하여금 "가상공간에서의 미로 찾기"를 경험하게 하며([2], [6]), 이는 사용자와 검색로봇의 웹서핑을 용이하게 할 수 있는 구조성을 웹사이트에 요구한다([1], [5]). 웹사이트를 계층적으로 구조화하는 것은 해당 사이트를 검색하고자 할 때, 정보를 재조직화하고 고려해야 할 대안들의 개수를 감소시킨다는 점에서 매우 유용하다[10].

웹사이트는 웹노드와 웹아크로 구성된 유방향 그래프로 파악할 수 있다. 웹노드는 웹페이지와 같은 html 파일에 해당하며, 웹아크는 웹노드간에 일방향으로 연결된 하이퍼링크에 대응한다. 따라서, 하나의 웹사이트는 루트노드(index.html)와 그 루트노드에 연결된 다른 노드들로 구성된 유방향 그래프로 볼 수 있다. 본 논문은 홈페이지를 루트노드로 하는 유방향 그래프로 구성된 웹사이트를 계층적인 트리 구조로 바꾸는 방법론에 대해 다룬다.

깊이우선탐색(DFA: Depth First Search)은 사이클이 없는 유방향 그래프에서 트리 구조를 발견하는데 가장 쉽게 구현이 가능한 방법이다. 그러나, 웹환경에서 대부분의 웹페이지는 다른 웹페이지들과 복잡한 상호연결을 유지하고 있기 때문에, DFA는 깊은 구조의 트리를 생성할 가능성이 많다. 이러한 깊은 구조는 웹사이트 탐색 시 긴 경로를 수반하므로, 특정 웹페이지로 접근하는데 많은 시간을 요구하는 단점이 발생한다. 이에 반해, 넓이우선탐색(BFA: Breadth First Search)은 DFA를 사용하는 것에 비해 보다 빠른 시간에 특정 웹페이지로의 접근을 가능하게 해준다. 또한, BFA는 모든 노드의 평균적인 깊이를 최소화하는 트리를 생성하는데 용이한 장점을 가지고 있다. 하지만, BFA는 극단적으로 뿌리 노드를 정점으로 한 다른 무수한 페이지로의 링크를 가지는 완전평면 트리 구조를 생성할 가능성이 있고, 이는 계층적인 트리 구조의 장점으로 언급한 대안의 증가를 가져올 수 있다.

생성된 계층적 웹구조를 의미론적으로 본다면, DFA와 BFA는 둘 다 결과적으로 트리 구조에서 특정 노드가 다른 특정 노드와 연결되어야 하는 이유가 명확하지 않다. 즉, 트리 구조를 생성할 뿐, 의미론은 확보하고 있지 않다. 웹구조의 의미론을 표현하는 트리 구조를 생성하기 위해 Wookey와 Geller는 tf-idf(term frequency-inversed document frequency) 기법을 사용하여 웹노드의 중요도를 계산한 후, 중요도간의 유클리디언 거리를 최소화하는 방법으로 트리 구조를 생성하였다.

본 논문은 웹사이트의 의미론적인 계층화를 최적화하기 위하여 기존 연구와 달리 웹아크에 우선순위, 즉 가중치(중요도)를 부여하였다. 생성된 트리 구조에서 웹아크는 사용자가 순회하는 경로를 가리키며, 따라서 웹아크 중요도의 합을 최대화하는 트리 구조는 사용자로 하여금 순회 가치가 있는 경로 구조를 보여준다는 점에서 유용하다고 할 수 있다. 이렇게 생성된 트리 구조는 사용자의 질의에 독립적인 웹사이트의 의미 있는 계층 구조로서 사용자로 하여금 해당 사이트를 효과적이고 효율적으로 검색할 수 있도록 도와줄 것이다.

본 논문은 다음과 같이 구성된다. 2장은 웹아크 중요도를 계산하는 방법을 기술하며, 3장은 min-cost arborescence 알고리즘을 통해 중요도의 합을 최대화하는 과정과 결과를 보여 준다. 마지막으로 4장을 통해 논문의 결론과 추후 연구 과제를 다룬다.

2. 웹아크 중요도

본 논문은 웹아크 중요도를 산출하기 위하여 Brin과 Page가 제안하여 Google 검색엔진의 검색방법에 적용된 PageRank 알고리즘을 이용하였다. PageRank 알고리즘을 통해 웹노드의 중요도를 계산하고, 이를 통해 웹아크 중요도를 산출하였다. PageRank 알고리즘은 본 논문의 웹아크 중요도와 같이 사용자의 질의에 독립적인 중요도를 산출하며, 또한 유용성이 입증된 검색 엔진 알고리즘

이라는 측면에서 채택하였다. 2.1절에서 PageRank 알고리즘을 리뷰하고, 이어지는 2.2절에서 웹아크 중요도를 기술하였다.

2.1 PageRank

PageRank 알고리즘은 Random Walk를 가정할 경우에 특정 웹페이지를 방문할 확률에 따라 웹문서의 중요도를 계산한다[1].

NR(i)를 웹노드 i의 중요도(PageRank), OD(i)를 웹노드 i의 out-degree, In(i)를 웹노드 i로 들어오는 아크의 시작 노드들의 집합으로 정의하면, 웹노드 i의 PageRank는 다음과 같이 계산할 수 있다.

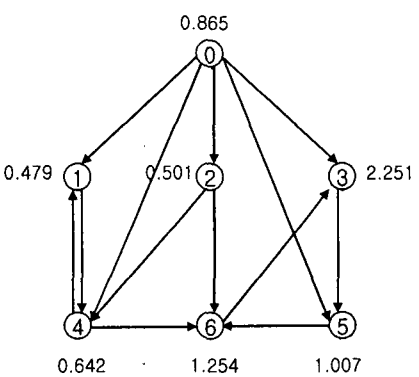
$$NR(i) = \sum_{j \in In(i)} \frac{NR(j)}{OD(j)}$$

각 노드의 PageRank값은 반복적인 계산을 통해 갱신되는 마코프 체인상의 확률분포로서 안정화될 때까지 계산된다. 상기 수식은 outgoing 링크가 없는 웹페이지에서 "sink"할 가능성이 있으므로, 연결되지 않은 특정 페이지로의 "teleport"를 고려하면 다음과 같은 수식을 사용하여 PageRank를 계산할 수 있다.

$$NR(i) = \frac{1-p}{n} + p \times \sum_{j \in In(i)} \frac{NR(j)}{OD(j)}$$

p는 "decay factor"로서 random walk를 통해 다른 웹노드를 방문할 확률이며, (1-p)는 n개의 웹페이지중 임의의 페이지로 이동할 확률이다.

[그림 1]은 6개의 노드로 이루어진 웹사이트 그래프에 PageRank 알고리즘을 반복적으로 적용하여 생성한 웹노드 중요도를 보여준다.



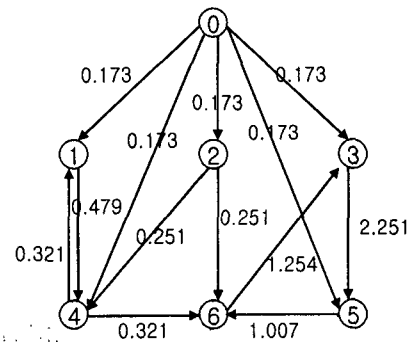
[그림 1. PageRank 생성 예제]

2.2 Link ranking

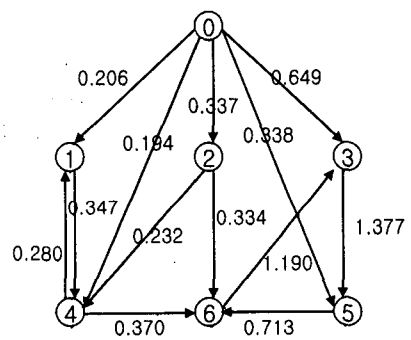
2.1절에서 기술한 PageRank는 Random Walk 모형을 기반으로 한 정적 중요도이다. 동일한 모형을 기반으로 웹아크 중요도를 고려한다면, 각 아크의 중요도는 사용자가 해당 아크를 경유할 확률이 된다. 사용자가 특정 아크를 경유할 확률은 아크의 시작(source) 노드를 방문할 확률에 종속적이므로 다음과 같이 간단하게 정의할 수 있다. LR(i,j)는 노드 i에서 j로 연결된 아크의 중요도이다.

$$LR(i,j) = \frac{NR(i)}{OD(i)}$$

시작 노드가 같은 모든 아크는 동일한 확률의 방문 가능성을 가지게 된다. [그림 2]는 [그림1]의 예제를 LR(i,j)로 계산한 결과이다.



[그림 2. random walk형 link ranking]



[그림 3. 가중된 link ranking]

LR(i,j]는 Random Walk 모형의 충실한 표현이지만, 아크의 종점(target) 노드의 중요도를 간과하는 측면이 있다. LR(i,j)에 의하면 시작 노드가 동일하면 높은 중요도를 가지는 노드로의 아크와, 낮은 중요도를 가지는 노드로의 아크가 동일하게 취급된다. 의미론적으로 볼 때, 높은 중요도를 가지는 노드로의 아크는 시작 노드의 중요도가 낮더라도 높게 산정될 필요가 있다. 또한, 종점 노드의 in-degree가 1이라면, 그 아크가 제외되면 달리 접근할 방법이 존재하지 않으므로, 이러한 상황에서의 아크는 중요하게 평가될 필요가 있다. 이러한 점을 감안하여 시작 노드와 종점 노드의 중요도를 가중평균한 웹아크 중요도를 개선하면 다음 LR*(i,j)와 같다.

$$LR^*(i,j) = p \times \frac{NR(i)}{OD(i)} + (1-p) \times \frac{NR(j)}{ID(j)}$$

p는 가중평균지수로서, 시작노드에서 종점노드로의 "push" 확률이며, (1-p)는 종점노드에서 시작노드를 "pull"하는 확률에 해당한다. p=1일 경우에는 LR(i,j)와 동일하게 된다. [그림 3]은 p=0.5로 균일한 가중치를 부여한 경우의 웹아크 중요도를 보여준다.

3. 웹사이트 구조화

3장은 2장에서 생성된 웹아크 중요도를 바탕으로 사용자의 순회 경로상의 중요도합을 최대화하는 트리 구조를 생성한다. 이러한 트리 구조를 구하는 문제는, 유방향 그래프의 루트 노드를 정점으로 하는 min-cost arborescence 문제로 정형화된다. min-cost arborescence 문제란 주어진 아크 비용의 합을 최소화하고 루트 노드를 제외한 모든 노드의 in-degree가 1이 되는 트리를 찾는 문제이다. 이 문제에 대한 해법은 Edmonds의 연구가 대표적인데, 이 해법은 $O(mn)$의 계산복잡도를 갖는 다항시간 해법이다. 이후, Tarjan은 Edmonds알고리듬을 $O(m \log n)$의 계산시간으로 구현할 수 있는 방법을 제안하였다. 여기서 m,n은 각각 아크수와 노드수를 의미한다.

단계 0 : $c_{ij} \leftarrow -c_{ij}$(최대화 문제로 변형). $InA(r)$에 있는 모든 아크 삭제. k=0. $G_0 = G, T_0 = \emptyset$

단계 1 : $v(\neq r) \in \{i \mid T_k \cap InA(i) = \emptyset, i \in V_k\}$인 모든 v에 대해, 다음 연산 수행.

1.1 $c_{\min} = \{c_{uv} \mid (u,v) \in \in A(v)\}$

1.2 $c_{uv} \leftarrow c_{uv} - c_{\min}$, $\forall (u,v) \in InA(v)$

1.3 $T_k \leftarrow T_k \cup \{(u,v) \mid c_{uv} = 0\}$

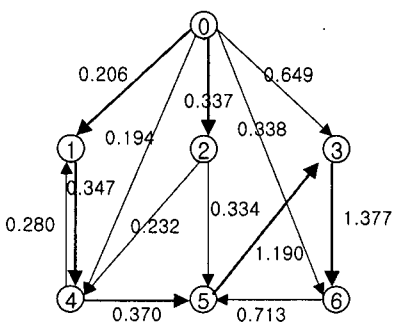
단계 2 : T_k에 어떤 환(cycle)도 없으면, T_k상의 모든 가상 노드를 확장(expand)하여 arborescence 구성. 그렇지 않으면 단계 3으로.

단계 3 : T_k상의 모든 환을 가상노드(pseudo node)로 압축(contraction). $T_{k+1} \leftarrow T_k \setminus \{T_k$상의 환에 포함된 아크$\}$. 새로 형성된 그래프를 G_{k+1}. $k \leftarrow k+1$로 두고, 단계 1로.

[그림 4. maximum-cost arborescence 알고리즘]

Edmonds알고리듬을 이용하여 본 논문의 웹그래프의 웹아크 중요도를 최대화하는 알고리즘을 구성하면 [그림 4]와 같다. 주어진 유방향 웹그래프를 G(V,A), 아크 중요도를 c_{ij}, 루트 노드를 r, 노드 i에 대해 $InA(i)$를 i로 들어오는(incoming) 아크의 집합이라 하자.

arborescence 알고리즘을 사용하여 [그림 3]의 웹그래프를 트리 구조로 변형하면 [그림 5]와 같다.



[그림 5. 알고리즘 적용 예]

4. 결론 및 추후 과제

본 논문은 유방향 웹그래프로부터 웹사이트의 순회 경로상의 중요도합을 최대화하는 트리 구조를 생성하였다. 이를 수행하기 위해 우선 PageRank의 Random Walk형 아크 중요도와 이를 개선한 가중된 아크 중요도를 제시하였다. 계산된 아크 중요도를 입력으로 min-cost arborescence 알고리즘을 적용하여 최대중요도합을 가지는 트리 구조를 생성함으로써, 사용자의 웹사이트 검색과 조회에 의미론적인 최적성을 부여하였다. 본 논문의 결과로 생성된 트리 구조는 사용자의 질의에 독립적인 구조로서 Google에서의 PageRank 활용과 유사하게 동적인 사용자 질의에 대응하는 알고리즘과 함께 활용될 수 있을 것이다.

본 논문은 일반적인 웹그래프 구조를 가정한 것으로서, 향후 실제 월드와이드웹상의 degree 분포를 고려한 알고리즘의 수정 보완이 필요하다[8]. 또한, 웹아크 중요도의 의미론을 확충하기 위해 hub and authority를 고려하는 것이 필요하다[4].

참고 문헌

1. Brin, S., Page, L., "The anatomy of a large-scale hypertextual web search engine.", Computer Networks, 30, pp.107-117, 1998

2. Conklin, J., "Hypertext, an introdution and survey", IEEE Computer, 20(9), pp.17-41, 1987.

3. Edmonds, J., "Optimum branching", J. Research of the National Bureau of Standards, 71B, pp.233-240, 1967.

4. Kleinberg, J. M., "Authoritative sources in a hyperlinked environment", Journal of the ACM, 46(5), pp.604-632, 1999.

5. Kumar, R., Raghavan, P., Rajagopalan, S. and Tomkins, A., "Trawling the web for emerging cyber-communities", WWW, pp.403-415, 1999.

6. Lau, T., Etzione, O., and Weld, D.S., "Privacy interfaces for information management", Communications of the ACM, 42(10), pp.89-94, 1999.

7. Page, L., Brin, S. Motwani, R. and Winograd, T., "The PageRank citation ranking: Brining order to the web", Technical Report, Computer Science Department, Standford University, 1998.

8. Pandurangan, G., Raghavan, P., Upfal, E., "Using PageRank to characterize web structure", LNCS 2387, pp.330-339, 2002.

9. Tarjan, R. E., "Finding optimum branchings", Networks, 7, pp.25-35, 1977.

10. Wookey, L. and Geller, J., "Semantic Hierarchical Abstraction of Web Site Structures for Web Searchers", Journal of Research and Practice in Information Technology, 36(1), pp.71-82, 2004.