Hyperlink Classification: A New Approach to Improve PageRank

Li Cun-he, Lv Ke-qiang
School of Computer & Communication Engineering,
China University of Petroleum, Dongying 257061, China
lvkeqiang_2001@163.com

Abstract

Hyperlink Structure is widely used in the hypertext classification, but it has not been paid enough attention. We propose a hyperlink classification approach to improve PageRank algorithm which is widely used in the link analysis of search engine. The cause of the topic drift problem is analyzed and the hyperlinks are classified according to their creating motivations and effects. The improved PageRank algorithm is implemented on the open source search engine NUTCH in Chinese Internet. The experimental results show that the improved PageRank algorithm performs better than the standard PageRank.

1. Introduction

With the greatly rapid growth of Internet, more and more information can be got from the Web. It is estimated that about eight billion Web addresses have been indexed by Google. However, what Google has indexed is only a small part of the Internet. Moreover, the documents on the Web are quite different from the traditional ones in the library; we have to develop new strategies to improve Web-search query results.

The unique hyperlink structures of the documents on the Web attract people's attention. Many algorithms are developed based on the hyperlink analysis. PageRank [1] and HITS [2] are the most important and well-known ones. The former which is used in Google search engine is proved to be successful; the latter is applied in Clever System of IBM. Both of them are based on the fact that the Web pages which have more inlinks (back links) are more likely to be visited.

HITS depends on query words. Firstly HITS invokes a traditional search engine to get a set of pages related to the query, and then expands the set by hyperlinks pointing to them or pointed by them. After that, HITS tries to find the top hubs and authorities by iterative calculations. All of the processing are done online and cost lots of time .Compared to HITS, PageRank algorithm precomputes a global score for each page called PageRank score. It takes much more time to calculate the PageRank than HITS, while the calculation is done offline only once and does not take up user's query time. In addition, PageRank is generated by using the whole Web graph, rather than a subset; it is much more robust than HITS, [3] has proved this by matrix perturbation theory and corresponding experiments. It seems PageRank is more popular than HITS.

A common problem shared by HITS and PageRank is topic drift. [4] believed that PageRank giving the same weight to the hyperlinks (the edge of Web graph) is the reason of the topic drift. We agree with their idea at this point and we present a new approach to improve PageRank.

In random surfer model, if page *u* is being visited, each link in page *u* has the same probability to be clicked. In most cases the random surfer model does not accord with people's behavior habit. Take a man who is reading NBA news for example, he is more likely to click the links about basketball than the ones related to the other topics like computer or commercial ads. Based on the fact above, we try to classify all the hyperlinks from a page and give them different probability according to their categories. Compared to the original algorithm, the PageRank score of the hyperlink classification is closer to the visiting probabilities distributed on the whole Web.

The paper is structured as follows: Section 2 introduces related work. Section 3 presents our new approach. In section 4 we carried out a series of experiments and confirm our approach. In section 5 we give conclusions and discuss ongoing work.

2. Previous work related to PageRank algorithm

The basic idea of PageRank [1] comes from the Academic citation. If page *u* has a link to page *v,* then the author of *u* is implicitly conferring some

importance to page *v*. Let N_u be the out degree of page *u*, and B_v be the set pages which link to page *v*. let *PR (p)* represent the importance (*PageRank*) of page *p*. Assign all pages the initial *PR=1/N*, *N* is the total number of all pages. PR formula is as follows:

$$\forall v PR_{i+1}(v) = \sum_{u \in B_v} PR(u)_i / N_u$$

In [2], the process can also be expressed as the following eigenvector calculation. Assume *M* is the square, stochastic matrix corresponding to the directed graph *G* of the Web, assuming all nodes in *G* have at least one outgoing edge. If there is a link from page *j* to page *i*, then let the matrix entry m_{ij} have the value $1/N_j$. Let all other entries have the value 0. The iteration of the PR formula corresponds to the matrix vector multiplication M*PR.

$$\vec{PR} = M \times \vec{PR}$$

One caveat is that the convergence of PageRank is guaranteed only if *M* is irreducible (*G* is strongly connected) and periodic [1]. The latter is guaranteed in practice for the Web, while the former is true if we modify the matrix *M* by adding a damping factor (*1-d)* to the rank propagation. We can define a new matrix M' in which we add transition edges of probability *d/N* between each pair of nodes in G:

$$M' = (1-d)M + d\left[\frac{1}{N}\right]_{N \times N}$$

The introducing of the decay factor *1-d* limits the effect of rank sink and guarantees the convergence to a unique rank vector. We substitute M for M', the formula is as follows:

$$\vec{PR} = M' \times \vec{PR}$$

$$\vec{PR} = (1-d)M \times \vec{PR} + d\left[\frac{1}{N}\right]_{N \times 1}$$

The modification of matrix *M* can be viewed as that the pages which have no outlinks were added virtual links to all pages in the graph. In this way, PR which is lost due to the pages with no outlinks is redistributed uniformly to all pages. If under a sufficient number of steps, the probability the surfer is on page *j* at some point in time is given by the formula:

$$\forall v PR_{i+1}(v) = (1-d) \times \frac{1}{N} + d \sum_{u \in B_v} PR(u)_i / N_u$$

There are two important improvements for the PageRank in past, one is Matthew Richardson [4] intelligent surfer. He proposed that the hop from page to page should be dependent on the content of the pages and the query terms the surfer is looking for. So the resulting probability distribution over pages is:

$$P_q(v) = (1-\beta)P'_q(v) + \beta \sum_{i \in B_v} P_q(u) P_q(u \to v)$$

where $P_q(u \to v)$ is the probability that the surfer transitions to page *v* given that he is on page *u* and is searching for the query *q*. $P_q'(v)$ specifies where the surfer chooses to jump when not following links. The $P_q(u \to v)$ and $P_q'(v)$ are derived from $R_q(v)$, a measure of *relevance* of page *j* to query *q*. $P_q(u \to v)$ and $P_q'(v)$ can be can be calculated by the following :

$$P_q(i \to j) = \frac{R_q(j)}{\sum_{k \in F_i} R_q(k)}$$

$$P'_q(j) = \frac{R_q(j)}{\sum_{k \in W} R_q(k)}$$

Where *W* is the whole set of pages, F_u is the pages that the page u pointing to. When the query is multiple-termed, the QD-PageRank can be given by combination of single term. The side effect of intelligent surfer that it takes much more time to precompute the PangRank_q for every word.

Another improvement is topic-sensitive PageRank [5]. Topic-sensitive PageRank compute a multiple importance scores for each page. Each importance score represents the importance of a page respecting to one topic. At query time, these importance scores are combined based on the topics of the query to form a composite PageRank score for those pages matching the query.

3. Link Classification PageRank

3.1 Link Classification PageRank

After analyzing PageRank and related works, we can conduct a conclusion that giving the same weight to the hyperlinks (the edge of Web graph) is the reason for the topic drift phenomenon. [6] showed that combining content and the anchor text of hyperlink with the link structure can raise the performance of information retrieval. We propose a Link Classification method to improve PageRank and carry out in the Chinese Internet environment.

We fetched about 21, 717 pages by open source search engine Nutch. We find that there are about 82 hyperlinks on each page on average, however, there are only twenty hyperlinks or even less are relating about the page's topic while most of the hyperlinks are about the information about the whole Web site map or the ads. The original idea of PageRank is scientific citation. But the authors of the Web pages on the Internet are not as so responsible and recommendable

as that of science thesis. It is essential to sort the hyperlinks and treat them separately.

In the random surf model, if Page *A* has a hyperlink pointing to Page *B*, we suppose that a page give a vote for *B* .But not all the links are recommending vote, some of them comes from the commercial benefit. We introduce a hyperlink quality of recommendation to weigh or scale the hyperlinks. Let *n* be the number of link classes. They are $C_1, C_2 \ldots C_n$ .The weight factor of the link classes are $\beta_1 k, \beta_2 k \ldots \beta_n k$. Suppose we classify the links rationally, we could count the number of each classification statistically. Let $t_1, t2 \ldots t_n$ be the quantity of each class. The sum of the probability of being clicked for each hyperlink in the same page must be 1. It can be expressed by the following formula:

$$t_1 C_1 + t_2 C_2 + \cdots + t_n C_n = 1$$

$$t_1 \beta_1 k + t_2 \beta_2 k + \cdots + t_n \beta_n k = 1$$

Supposing we've got the weight factor ratio β_i of each class by mining the Web site log, the k can be computed easily. The PageRank can be expressed as following:

$$PR_{i+1}(v) = (1-d) \times \frac{1}{N} + d \sum_{u \in B_v} PR(u)_i \times f(u \to v)$$

$$f(u \to v) = \beta_j \times k \qquad u \to v \in C_j$$

$f(u \to v)$ refers to the weight factor of link $u \to v$. If $u \to v$ belongs to Class *j*, its weight is $\beta_j \times k$

3.2. A simple hyperlink Classification method for testing PageRank

[7, 8, 9] have done much work on the hypertext classification using hyperlinks, but few paper talked about the hyperlink classification.

Hyperlinks play a guide's role in the Internet. [10] has already present different criterion and classified the hyperlinks on them for different application field. For the search engine systems, [2] provided an approach that if the text in the vicinity of the heft from *p* to *q* contains text description of the topic at hand, we want to increase weight of link.

In this paper, we propose a simple hyperlink Classification based on Vector Space Model to test the approved PageRank. We categorize the hyperlinks into 4 classes as follows: (1) recommending Inner Link, the weight factor β_1 of Class , $\beta_1 = 5$; (2) other Inner Link including the navigating links, the commercial ads, etc, β_2=1.0; (3) recommending outer Link, β_3=6.0 ;(4) other outer Link, β_3=1.0. The entire weight factor is giving by assumption.

The Web pages are described by VSM (Vector Space Model). Let *S* be set of the whole Web pages, let *N* be the total number of the whole Web pages. After deleting the HTML tags, each Web page *d* can be viewed as $d = \{w_1, w_2, \ldots . w_n, \}$ let *k* is the counting number of word *w*. We calculate the top 20 words to present the main idea of the content of the document *d* by the TFIDF formula.

$$w_i = tf_i \times idf_i = \frac{m_i}{\sum m_j} \times \log\left(\frac{M}{k_i}\right)$$

Let *L* be the set of the whole hyperlink in Web pages *d*. For $link_i \in L$, we extract the anchor text of $link_i$, $link\_anchor = \{a_1, a_2, \ldots . a_n, \}$. We regards the cos distance as the relevance of $link_i$ and Web pages *d*.

$$r(\text{link}, d) = \cos(link\_anchor, d)$$

For every *link*, we firstly compute the relevance between the source page of the link and the link as *Rev(source, link*) then compute the relevance between the destination page of the link and the link as *Rev(destination, link)*, Lastly we compute the average of *Rev(source ,link)*and *Rev(destination, link)* as *score(link)*.

The steps of the Simple hyperlink Classification are as follows:

Step1: Classify all the hyperlink inner links and outer links. If the source web URL of a hyperlink and the destination web URL belongs to the same domain name, it is a inner link , or else it is an outer link.

Step2: For each inner link *i*, if score *(i)*>average inner link score, link *i* belongs to recommending Inner Link, or else link *i* belongs to other Inner Link.

Step3: For each outer link *i*, if score *(i)*>average outer link score, link *i* belongs to recommending outer Link, or else link *i* belongs to other outer Link.

The hyperlink classification method is just to test our idea. More work need to be done in the future.

4. Experiments in Chinese Internet based on Nutch

In this Section, we describe an experiment to test the hyperlink classification PageRank. It cost about 36 hours to crawl about 27,717 Web pages on the Chinese Internet by open source search engine Nutch. After that we transported the crawling Web pages to database. Table 1 shows the number of the pages and hyperlinks we crawled.

At first, we compute the value of the two PageRanks. Secondly we obtain the Web pages containing the query word and then order them by the two PageRanks separately. We found 5 volunteers and they were asked to give each of the searching result a satisfying score. The average score is the score of the algorithm. Figure

1 is the comparison of the two algorithms. Obviously the precision of the improved one is higher to the original one.

Table1. **The number of the pages and hyperlinks**

Crawling seed	Number of hypertext	Number of links
http://www.hao123.com	27717	1852613

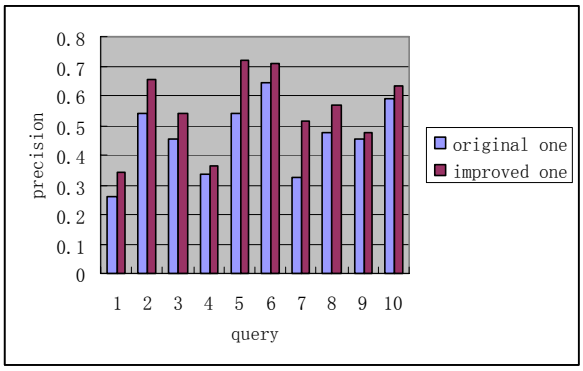

Figure1. **The contrast result**

At last, we analyzed the two algorithms on the time requirements. For the hyperlink classification PageRank, we need more time to classify the hyperlinks. However, when the classification is done, it takes the same time to compute the PageRanks.

5. Summery and ongoing work

In this paper, we propose a Hyperlink classification way to improve PageRank algorithm. The experiment results show that it works better than the original one. Hyperlink classification is a new filed which is ignored by most of us. There is much work need to be done in the future work. We plan to introduce ontology or conception to the hyperlink classification. We believe the hyperlink classification will make the information retrieve systems perform much better.

References

[1] Sergey Brin and Larry Page. "The anatomy of a large-scale hypertextual Web search engine". Computer Networks and ISDN Systems, Volume 30, April 1998, pp.107-117

[2] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan. "Automatic resource compilation by analyzing hyperlink structure and associated text". Computer Networks and ISDN Systems, Volume 30,April 1998,pp.65-74

[3] Andrew Y. Ng, Alice X. Zheng, Michael I. Jordan, Link Analysis, Eigenvectors and Stability, International Joint Conference on Artificial Intelligence, 2001. pp.903-910

[4]Matthew Richardson, Pedro Domingos. "The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank", volume 14. MIT Press, Cambridge, MA, 2002.

[5] Haveliwala T H. "Topic-sensitive PageRank".IEEE Transactions on Knowledge and Data Engineering, ,Volume 15, August, 2003, pp.784-796

[6] ZHANG Min, GAO Jian-Feng , and MA Shao-Ping. "Anchor Text and Its Context Based Web Information Retrieval". JOURNAL OF COMPU TER RESEARCH AND DEVELOPMENT, China,January 2004, vol.41,No.1, pp.221~225.

[7] S Chakrabarti, B Dom, P Indyk . "Enhanced hypertext categorization using hyperlinks", ACM SIGMOD Record, ACM Press New York, NY, USA, Volume 27, June 1998, pp. 307~318.

[8] L Getoor, E Segal, B Taskar, D Koller. "Probabilistic Models of Text and Link Structure for Hypertext Classification", Workshop Notes of IJCAI-01 Workshop on Text Learning: Beyond Supervision, Washington, USA, 2001.

[9] J Furnkranz. "Hyperlink ensembles: A case study in hypertext classification". Technical Report OEFAI-TR-2001-30, Austrian Research Institute for Artificial Intelligence, Wien, Austria, 2002

[10] MIZUUCHI Y, TAJIMA K. "Finding context paths for Web pages". Proceedings of the Tenth ACM Conference on Hypertext and Hypermedia. ACM Press New York, USA, 1999 , pp.13-22.

[11] Krishna Bharat, George A. Mihaila. "Hilltop: A Search Engine based on Expert Documents". Http://www.cs.toronto.edu./~georgem/hilltop/. 2005

[12] http://lucene.apache.org/nutch/