The impact of inter-host links in crawling important pages early*

Md. Hijbul Alam, JongWoo Ha, Kyu-Sun Sim, SangKeun Lee
College of Information and Communications, Korea University
{hijbul, okcomputer, bluesks, yalphy}@korea.ac.kr

Abstract

The dynamic nature and exponential growth of the World Wide Web remain crawling important pages early still challenging. State-of-the-art crawl scheduling algorithms require huge running time to prioritize web pages during crawling. In this research, we proposed crawl scheduling algorithms that are not only fast but also download important pages early. The algorithms give high importance to some specific pages those have good linkages such as inlinks from different domains or host. The proposed algorithms were experimented on publically available large datasets. The results of experiments showed that propagating more importance to the inter-host links improves the effectiveness of crawl scheduling than the current state-of-the-art crawl scheduling algorithms.

1. Introduction

The number of web pages in the World Wide Web is growing at an astonishing rate, as millions of web pages are created every day. However, only some portions of these web pages are indexed by search engines due to limited resources. Moreover, most users only view the top ranked pages in the search results. Therefore, crawlers should download important pages early so that search engines could index important pages early.

Crawling important pages early possess a great challenge which is a well studied problem [5]. Recently, Cho et al. proposed the concept of PageRank lower bound in the RankMass crawler which is a powerful tool for downloading web pages effectively than any previously proposed crawlers [4]. However, the RankMass crawler propagates importance through all the links to compute PageRank lower bound which is not necessary to download important pages early. Let's consider the host A graph in Figure 1 without the link from page L to page D. Here page A, B, C are downloaded and D and E are discovered from B. The importance of page D will be computed using the importance of page B, and the importance of page E will be computed from page B and page C. However, the propagation of importance from Page C to downloaded page B while exploring page C does not have any immediate and

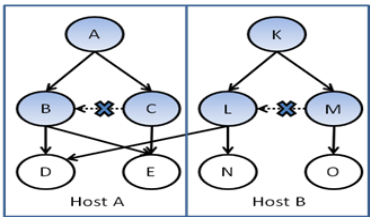

Figure 1: An example graph.

significant effect in downloading. Because when B propagates its importance again, page D and E have already downloaded with high probability. Furthermore, most of the existing crawlers assess all incoming links equally. However, inlinks of a page from other host usually are implicit conveyance of authoritative or quality information, because inlinks from other host can not be produced by the web page owner. For example, mobide is a host; korea.ac.kr is a domain in the http://mobide.korea.ac.kr, and in Figure 1 page D of host A should be more important than page E because it receives importance from a different host through the inlink (L, D).

We proposed two crawlers to deal with each of the problems. The first one is the Fractional PageRank (FPR) crawler that avoids the propagation of importance through all the links. It reduces the running time to prioritize URLs while producing almost same effectiveness with the recently published state-of-the-art crawler, RankMass crawler [4]. The second one is the Extended FPR crawler that propagates more importance through the inter-host links than the intra-host links. We compared the results of the proposed crawlers to

* This work was supported by the Korea Science and Engineering Foundation(KOSEF) grant funded by the Korea government(MEST) (No.2009-0077925)

that of the RankMass crawler and demonstrated that by propagating more importance through inter-host links web pages can be crawled more effectively than RankMass crawler.

The contribution of the paper is two folds. First, we proposed the Extended FPR crawler from the FPR crawler [6] by categorizing and weighting different types of links. Second, we thoroughly evaluated the crawling scheme by various experiments and provided the explanation why the proposed crawlers perform better.

2. Related works

A large number of studies examined the scheduling of the web crawling process such as Breadth-first crawler and Adaptive Online Page Importance Computation (OPIC) crawler. Although non PageRank based algorithms, the OPIC driven crawler and Breadth-first crawler, give effective crawling scheduling, Baeza-Yates et al. [2] showed that the PageRank based crawler is more effective than those. Cho et al. proposed the RankMass and Windowed RankMass crawlers [4] provide an importance coverage guarantee of downloaded pages during crawling. However, they need large running time to compute the PageRank lower bound of web pages. PageRank lower bound considers all the linkages information of the downloaded pages so far. Propagating importance through all link structures do not contribute significant increase in effectiveness of crawling web pages but consume huge computing resources.

3. Proposed Algorithms

In following section we describe how to compute Fractional PageRank (FPR) and how the FPR is used in Fractional PageRank crawling algorithm. In section 3.2, we proposed the Extended FPR crawler that gives high importance to the inter-host links.

3.1 Fractional PageRank Crawlers

To compute FPR we group web pages in two types: downloaded pages and discovered pages. After downloading a page, the URLs of that page are extracted, and unseen URLs are placed into the queue. The URLs in the queue which will be downloaded next are known as discovered pages. Therefore, the downloaded linkages information can be divided into two types: downloaded pages to downloaded pages and downloaded pages to discovered pages. The FPR is computed only for the discovered pages using the links between downloaded pages to discovered pages during crawling. The FPR of a page is the total importance received by a discovered page through the links before the page is downloaded. The discovered page with high FPR will be downloaded first. Let's consider the host A graph of the Figure 1 independently where B is downloaded before C and the links toward discovered pages are (B, D), (B, E) and (C, E). The FPR of page D and E will be computed by the importance received through the above three links. The importance of link from C to B will not contribute to the FPR of B since B is already downloaded. The computation of FPR and downloading the pages using FPR is done by the Fractional PageRank crawler that is given in Algorithm 1. Here t_i is the initial value of the seed pages, N_i is number of outlinks, and d is the probability of visiting a page following a link from the current page.

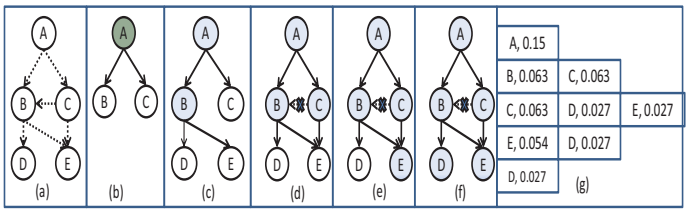

Figure 2: Downloading sequence of web pages using Fractional PageRank crawler

Algorithm 1: Fractional PageRank Crawler

```
Begin
01 : foreach p_i in the set of seed pages:
02 :     fpr_i = ( 1 − d ) * t_i;
03 : while ( Queue is not empty ):
04 :     Pick p_i with largest fpr_i
05 :     Download p_i
06 :     foreach p_j linked to by p_i and p_j is not downloaded:
07 :         fpr_j = fpr_j + ( d * fpr_i ) / N_i
08 :     fpr_i = 0
End
```

Figure 2 shows that how the FPR crawler downloads web pages. Here, shadow nodes denote downloaded pages. The real values in Figure 2(g) denote FPR values generated during crawling. Let's consider Figure 2(a) graph where the FPR crawler will run, and A is the seed page. B and C are discovered from A and placed in a queue. B and C have the same FPR, and sequentially, B is selected for downloading in Figure 2(c). From page B, D and E are discovered. Now, there are three pages in the queue. As the FPR of C is highest, it will be downloaded and explored next. Although D might discovered earlier than E, FPR of E becomes higher as a portion of importance of C is distributed to the

discovered page E. Therefore, E is downloaded before D which is shown in Figure 2(e).

3.2 Extended Fractional PageRank Crawlers

The FPR crawler avoids the propagation of probability (i.e. importance) through the links between downloaded pages to downloaded pages since they propagate insignificant importance at a time to the web pages in the crawl frontier. A small portion of hyperlinks are inter-host links. Although probability propagation through the inter-host links during crawling contributes small importance values, inter-host links play an important role in the PageRank [1] computation since most of the importance from one host to another host is propagated through these links [7]. Motivating from the fact, we proposed Extended FPR crawler named as FPR@M crawler that distributes higher FPR to inter-host links than intra-host links. For example, in Figure 1 an inter-host link from L to D will propagate M times more importance than an intra-host link L to M. Therefore, D will become more important than other pages and will be downloaded early. The Extended FPR crawler is presented on algorithm 2. The algorithm checks the host of each outgoing links with the host of the source page in line 7. If the host is not matched, the link is an inter-host link, and it will propagate higher importance such as M times more importance to inter-host links than intra-host links. Number of inter-host links in a page v is denoted by $Ninter_v$, and the remaining links are intra-host links denoted by $Nintra_v$. Total importance of a page will be divided by the summation of intra-host links plus M times inter-host links (i.e $Nintra_v + Ninter_v * M$), and an inter-host link will propagate M times of that portion whereas an intra-host link will receive only that portion.

Algorithm 2: Extended Fractional PageRank Crawler

```
Begin
01 : foreach p_i in the set of seed pages:
02 :         fpr_i = ( 1 - d ) * t_i;
03 : while ( Queue is not empty ):
04 :   Pick p_i with largest fpr_i
05 :   Download p_i
06 :   foreach p_j linked to by p_i and p_j is not downloaded:
07 :     if ( p_i -> p_j is an inter host link)
08 :       fpr_j = fpr_j + (d * M * fpr_i) / (Nintra_v + Ninter_v * M)
09 :     else fpr_j = fpr_j + ( d * fpr_i ) / (Nintra_v + Ninter_v * M)
10 :   fpr_i = 0;
End
```

4. Experimental Results

The performance of the proposed crawlers is measured with two metrics. First one is the computational overhead the algorithm introduces in terms of the running time. Second one is the summation of PageRank (cumulative PageRank) of the downloaded pages during different points of a crawl. The strategy that makes the cumulative PageRank higher by downloading fewer pages is an effective one. The following section describes the data and experimental setup to evaluate the proposed crawlers.

4.1 Data sets and Experimental setup

The experiment is performed on large datasets available on the web in [8]. They can be accessed by the WebGraph framework [3]. Table 1 shows the summary of data sets that we used. The uk-2007-05 graph consists of around 105 million pages and about 3.7 billion links that correspond to those pages under the .uk top domain. The number of inter-host links in the graph is 136,730,188, and the ratio of intra-host links and inter-host links is around 96:4. The uk-2007-05 graph was crawled in May 2007 with a maximum depth per host of 16 and 50000 pages per host. For our simulation, we used only a single Intel Core 2 Quad 2.4 GHz CPU with 4 GB RAM in a Solaris Machine. It was implemented in Java. We performed all the processing in main memory. We calculated the PageRank [1] values for every page in a given subgraph. The top 160 pages according to PageRank among the collection were selected as the seed pages for crawling. An initial value was distributed uniformly among them.

Table 1: Datasets used in Experiment

Graph	Nodes	Links	Inter-host links
uk-2007-05	105,896,555	3,738,733,648	136,730,188

We compared the performance of FPR crawlers with Windowed RankMass crawler. The RankMass crawler takes unrealistic time since for each downloaded page the probability calculation increases exponentially. Therefore, Cho et al. proposed an approximation of the RankMass crawler called Windowed RankMass crawler that reduce the overhead by batching together sets of probability calculations and downloading sets of pages at a time [4]. Authors showed that the 5% Windowed RankMass crawler downloads pages as effective as RankMass crawler, and setting the window to 100% will give the similar effect of breadth first crawler.

4.2 Efficiency

Figure 3 shows the running time required for prioritizing the URLs and download them. The time

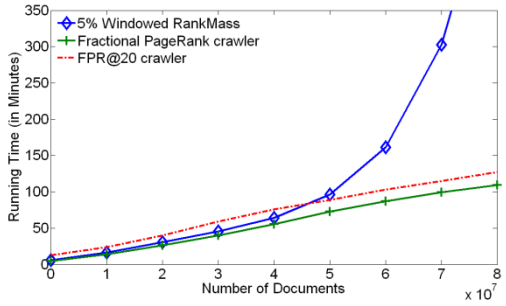

Figure 3: Examining the Running of the proposed crawlers.

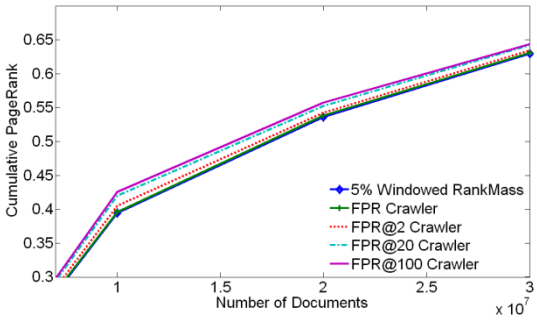

Figure 4: Examining the effectiveness the proposed crawlers

needed by Windowed RankMass crawler increases rapidly when the number of pages increases. When already downloaded pages are discovered again, Windowed RankMass crawler continues the propagation of the internal probability to compute the exact lower bound of each page. However, for the FPR crawler the propagation of the probability from downloaded pages cannot be done more than once, although it might be discovered many times. The results depicted in Figure 3 show that the FPR crawler is quick since it downloaded 80 million pages within 2 hours. However, the 5% Windowed RankMass crawler required around 9 hours. Although FPR@M crawler initially took more time, overall it took less time than the Windowed RankMass crawler.

4.3 Effectiveness

Figure 4 shows the effectiveness of the FPR crawlers and the 5% Windowed RankMass crawler. The results indicate that the FPR crawler downloaded web pages almost as effective as the 5% Windowed RankMass crawler. The FPR@2 crawler has performed better than the 5% Windowed RankMass crawler. We experimented with different values of parameter M. It showed that high values of M produce more effective crawl scheduling. If M equals to one, FPR@M crawler will turn into simply a FPR crawler which is proposed in the algorithm 1. FPR@100 outperformed other crawling policies. We found 100 as the optimal value of M. Figure 4 shows that the effectiveness achieved between FPR@20 crawler and FPR@100 crawler is not as significant as the effectiveness achieved between FPR@2 crawler and FPR@20 crawler. Therefore, the value of M greater than hundred will result almost same crawl scheduling as FPR@100. In addition to that, Figure 4 illustrates that all FPR@M crawlers achieve greater effectiveness than 5% Windowed RankMass crawler.

5. Conclusion

In this paper, we proposed two crawling algorithms that are scalable and able to prioritize the web scale frontier effectively and efficiently. The Extended FPR crawler is more effective than the RankMass crawler and overall takes significantly less time than the RankMass crawler. In the future, we will study the spam combat-ability of FPR crawlers and how to use the other available information in the downloaded pages, such as anchor text for effective crawling schedule.

References

[1] S. Brin, L. Page: The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.30, 1-7, 107-117, (1998).*

[2] R. Baeza-Yates, C. Castillo, M. Marin, A. Rodriguez: Crawling a country: Better strategies than breadth-first for web page ordering. *Proc. Int'l World Wide Web Conf. pp. 114-118, 2005.*

[3] P. Boldi, S. Vigna: The webgraph framework I: Compression techniques. *Proc. Int'l World Wide Web Conf. pp. 595-602, 2004.*

[4] J. Cho, U. Schonfeld: Rankmass crawler: A crawler with high personalized pagerank coverage guarantee. *Proc. Int'l Conf. on Very Large Data Base (VLDB), pp. 375-386, 2007.*

[5] J. Cho, H. Garcia-Molina, L. Page: Efficient crawling through url ordering. *Proc. Int'l World Wide Web Conf. pp. 161 – 172, 1998.*

[6] Md. Hijbul Alam, JongWoo Ha, and SangKeun Lee: Fractional PageRank Crawler: Prioritizing URLs Efficiently for Crawling Important Pages Early. *Proc. Int't Conf. on Database Systems for Advanced Applications (DASFAA), pp. 590 – 594, 2009.*

[7] S. D. Kamvar, T. H. Haveliwala C. D. Manning and G. H. Golub. Exploiting the Block Structure of the Web for Computing PageRank. *Technical report, Stanford University, 2003.*

[8] http://webgraph.dsi.unimi.it/