Graph Based Crawler Seed Selection

Shuyi Zheng[1] Pavel Dmitriev[2] C. Lee Giles[1]
[1]Pennsylvania State University, University Park, PA 16802, USA
[2]Yahoo! Labs, 2821 Mission College Blvd, Santa Clara, CA 95054, USA
shzheng@cse.psu.edu dmitriev@yahoo-inc.com giles@ist.psu.edu

ABSTRACT

This paper identifies and explores the problem of seed selection in a web-scale crawler. We argue that seed selection is not a trivial but very important problem. Selecting proper seeds can increase the number of pages a crawler will discover, and can result in a collection with more "good" and less "bad" pages. Based on the analysis of the graph structure of the web, we propose several seed selection algorithms. Effectiveness of these algorithms is proved by our experimental results on real web data.

Categories and Subject Descriptors
H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

General Terms
Algorithms, Design, Experimentation, Performance

Keywords
Crawler, Seed Selection, PageRank, Graph Analysis

1. INTRODUCTION

Crawling is one of the most important tasks of a search engine, and is also a critical part of many other web applications. The breadth, depth, and freshness of the search results depend crucially on the quality of crawling. As the number of pages and sites on the web increases rapidly, deploying an effective and efficient crawling strategy becomes critical for a search engine.

A typical crawler [2] maintains a list of unvisited URLs called *frontier*, which is initialized with seed URLs. In each crawling loop, the crawler picks a URL from the frontier, fetches the corresponding page, parses the retrieved page to extract URLs, and adds unvisited URLs to the frontier. Typically, the crawler needs to keep revisiting some or all the URLs to check if they are updated since the last crawl. Due to the infinite nature of the web and the competition between getting new URLs and refreshing the old ones, even web-scale crawlers can never crawl "all" the URLs from the web.

Different crawling strategies resulting from different ways of ordering the URLs in the frontier can explore the web in different ways. However, all of them start from the seed pages and proceed by exploring the neighborhoods of the seed pages in one way or another. Thus, to a large extent, selecting good quality seed determines the quality of the crawl.

Since there exists a wide variety of crawling strategies, in this paper, we make two assumptions. One, a crawler crawls pages only within h hops from the seed, and two, the number k of seeds is given. Both of these assumptions are not crucial for the algorithms we will describe later, but will simplify the explanation.

2. STRATEGIES FOR SEED SELECTION

Intuitively, the problem of seed selection is, given a currently known portion of the web and the desired number of seeds k, to select seeds so that as many as possible new good pages will get crawled, and as many as possible currently crawled good pages will be retained. There are several naive strategies which can achieve this goal.

The simplest strategy, used as a baseline in our experiments, is to select the seeds at random. The second strategy is to select k pages with the highest PageRank. This makes sense since one can expect to find high quality pages around other high quality pages. Another intuitive strategy is to select k pages with most outlinks.

A heuristic, which can improve the above strategies, is to first split the known portion of the web into smaller units, such as web sites, and then select seeds independently for each web site. Such strategy allows distributing seeds "evenly" across the web, allocating to each web site a fraction of the total number of seeds, which is proportional to the site's importance. It also allows performing seed computation in parallel and on a smaller data set.

However, naive seed selection strategies cannot directly optimize the criteria we are interested in. To formally define the problem of seed selection, we assume that every crawled web page has an associated value. Higher value indicates higher quality or higher potential to discover new pages calculated, for example, as the number of uncrawled URLs the page links to. In addition, the value can be negative if the page is undesirable, such as a spam page. We can now define the seed selection problem as maximizing the value of the portion of the web graph "covered" by the seeds.

DEFINITION 1 (SEED SELECTION PROBLEM). *Given a directed graph $G = (V, E)$, a function $w : V \rightarrow \mathbb{R}$, assigning a weight $w(v)$ to every $v \in V$, and edges unweighted, given the number of seeds k, and the number of hops a seed covers h, select the seeds so that $w(\cup_{i=1}^{k} A_i)$ is maximized, where $A_i = \{v | v \in V, v$ within h hops of seed $i\}$, $w(A) = \sum_{v \in A} w(v)$.*

WWW 2009, April 20–24, 2009, Madrid, Spain.
ACM 978-1-60558-487-4/09/04.

In this formalization, the seed selection problem is an instance of the *Maximum K-Coverage Problem*, which is known to be NP-hard [1]. However, a greedy iterative approximation exists for this problem, which achieves $1 - \frac{1}{e}$ approximation [1]. The algorithm is shown below.

Algorithm 1 Seed Selection Algorithm

Input: Weighted Graph G of (a portion of) the web, Maximal Seed Number k, Number of hops allowed h
Output: Selected Seeds $\mathcal{S}$
Algorithm:

1: FOR $i = 1$ to k
2: Find s which covers maximal value within h hops
3: Make h hops on graph G, mark all covered vertices
4: Remove s and all covered vertices; Update G
5: Add s to $\mathcal{S}$
6: IF G is empty THEN break

Unfortunately, performing even a single iteration of the algorithm is computationally expensive for large h, due to the exponential in h complexity of step 2. Therefore we resort to an approximation again. We propose and evaluate four approximation procedures for performing step 2:

L1: Choose a page p with the highest $numOutlinks(p)$

L2: Choose a page p with the highest sum of values of all pages within 1 hop of p

L3: Choose a page p with the highest sum of values of all pages within 2 hops of p

SCC: Choose a strongly connected component in the graph with the highest sum of page values, select a page p within that component according to the $L2$ algorithm.

Below we describe our experimental setup and presents results evaluating the quality of the above algorithms.

3. EVALUATION

To evaluate the performance of the algorithms, we selected a random sample of 1000 web sites from Malaysian web, each containing at least 100 pages and having at least 1 external link into the site. We used an experimental web-scale crawl from the summer 2008 to obtain all pages crawled from these web sites. We also obtained other page attributes, such as whether the page was determined to be spam, whether the page was clicked on in search engine results, and how many uncrawled pages the page links to. We assigned a value to each page according to the following rules:

For every page p set $w(p) = 1$. If p is spam, set $w(p) = -10$. If p was clicked, set $w(p) = 10$. Let n be the number of uncrawled pages the page links to. Set $w(p) = w(p) + n * P_{uniq}$, where P_{uniq} is the probability of uniqueness of a newly crawled page on the site. This probability is calculated based on past crawl statistics.

We implemented the three naive algorithms and the four graph-based algorithms described above. By measuring the coverage of pages and of value independently on each web site, we evaluated the performance of the algorithms in terms of the percentage of improvement over the random seed selection strategy. In the experiment, we varied the number of seeds generated for each site from 1 to 10, keeping the number of hops fixed at 5. The figures shown below present the average performance over 1000 hosts.

Figure 1 shows the results for the page coverage and Figure 2 shows the results for the value coverage (error bars

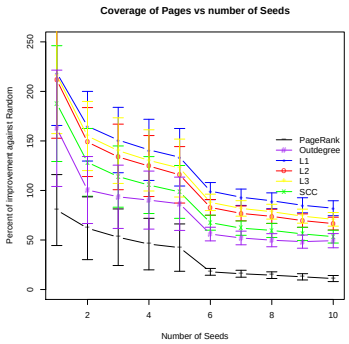

Figure 1: Coverage of Pages

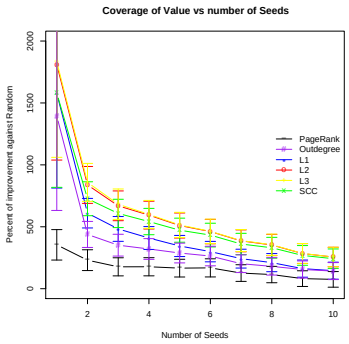

Figure 2: Coverage of Value

indicate standard errors). The following conclusions can be drawn from these results: First, random strategy performs poorly, while all other strategies outperform it significantly. As one may expect, the improvement over the random strategy is larger when the number of seeds is small, and it decreases as the number of seeds grows. Second, all four of the graph based algorithms outperform the naive seed selection strategies. Third, $L1$ algorithm, which uses outdegree instead of value, produces best results in terms of the coverage of pages, but performs much worse in terms the coverage of value. Fourth, $L2$ and $L3$ algorithms perform similarly ($L3$ slightly outperforms $L2$), suggesting that using 1 hop approximation is enough to identify a good seed. Finally, the SCC algorithm does not perform as well as $L2$ and $L3$.

4. CONCLUSION

In this paper we discuss the problem of seed selection for a web-scale crawler. We formalize this problem as a graph theoretic problem, analyze its complexity and propose several approximate algorithms. Experimental results on a dataset of 1000 real web sites demonstrates that our algorithms significantly outperform the naive seed selection strategies.

5. REFERENCES

[1] D. Hochbaum and A. Pathria. Analysis of the Greedy Approach in Problems of Maximum k-Coverage. *Naval Research Logistics*, 45(6):615–627, 1998.

[2] G. Pant, P. Srinivasan, and F. Menczer. Crawling the Web. *Web Dynamics*, pages 153–178, 2004.