Internet Mathematics Vol. 1, No. 1: 115-126

Algorithmic Challenges in Web Search Engines

Monika R. Henzinger

Abstract. In this paper, we describe six algorithmic problems that arise in web search engines and that are not or only partially solved: (1) Uniformly sampling of web pages; (2) modeling the web graph; (3) finding duplicate hosts; (4) finding top gainers and losers in data streams; (5) finding large dense bipartite graphs; and (6) understanding how eigenvectors partition the web.

1. Introduction

A web search engine consists of three parts: (1) A *crawler* that retrieves web pages to be put into the engine's collection of web pages; (2) an *indexer* that builds the *inverted index* (also called the *index*), which is the main data structure used by the search engine and represents the crawled web pages; (3) and a *query handler* that answers user queries using the index.

For our purposes, the crawler views the web as a graph: Each web page is a node and each hyperlink a directed edge. A major question a crawler has to face is which pages to retrieve so as to have the "most suitable" pages in the collection. Some of the open problems posed below can lead to improvements for crawlers:

- A better understanding of the graph structure (Section 3) might lead to a more efficient way of crawling the web.

1542-7951/03 $0.50 per page

- A better understanding of various web page properties (Section 2) can indicate which populations of pages are underrepresented in the crawl so far.
- An efficient way of finding duplicate hosts (Section 4) can help the crawler avoid crawling the duplicate of an already crawled host.

Given a collection of queries that are posed to a search engine, an obvious question is which queries are asked most often. However, to detect temporal effects, it is also interesting to ask for the "top gainers" and the "top losers." This problem is posed in Section 5.

Finally, we present two problems that are related to finding a topic-dependent clustering of the web or a subgraph thereof: Section 6 discusses the problem of finding dense directed bipartite subgraphs. Section 7 raises the question of how eigenvectors of various matrices derived from the web graph relate to cuts in the web graph. We sketch each of these open problems and give references to prior work in the area.

2. Sampling Web Pages

Understanding the web and its properties has been a hot research topic since the inception of the web. How many pages are on the web? How many of them are indexed by a given search engine? How many pages are in a certain language or in a certain domain? What is the average length of a web page? What percentage of web pages are homepages? And how do these properties change over time? Search engines are trying to capture as much of the web as possible. Additionally, the proportions of different types of pages, like pages in different languages, should roughly be proportional to the proportions of the types on the web. It is straightforward to keep track of the proportions in the crawl. Thus, if the above statistics are known for the web, the crawler can determine which types of pages are grossly underrepresented so far and try to crawl more of them.

A technique for uniform sampling of web pages could be used to answer all such questions except for the first. Unfortunately, no such technique is known even though there has been a considerable amount of research on this topic. Lawrence and Giles [Lawrence and Giles 99] used an approach based on random testing of IP addresses: They selected a random IP address and checked whether it hosts a web site. If so, they try to sample the web pages accessible at this site. However, it remains an open problem how to uniformly sample web pages on a web site if one does not have an exhaustive list of these pages.

Henzinger et al. [Henzinger et al. 00] proposed performing a specific random walk on the (directed) web graph and then sampling the traversed pages inversely

proportional to their value in the stationary distribution of the random walk. There are various problems with this approach. One problem is, of course, that it is not clear how many steps one must perform in order to approximate the equilibrium distribution. Another problem is that the specific random walk that they propose cannot be implemented directly, but this could be solved by using a different random walk than the one presented in their paper [Henzinger et al. 00].

Bar-Yosseff et al. [Bar-Yossef et al. 00] converted the web graph into an undirected, connected, and regular graph. The equilibrium of a random walk on this graph is the uniform distribution. Again, it is not clear how many steps such a walk needs to perform. A more significant problem, however, is that there is no reliable way of converting the web graph into an undirected graph. Bar-Yosseff et al. proposed asking various search engines for the in-edges of a given page in order to sample all adjacent edges of a given page. However, frequently only a subset of all in-edges can be found in this way.

Finally, Rusmevichientong et al. [Rusmevichientong et al. 01] modified the approach of [Henzinger et al. 00] to yield a method for which, in the limit, a uniform sample is generated. In practice, we believe that their approach would not work well, since there is a plethora of hosts on the web that are highly linked within the host, but with very few links leaving the host. If the random walk in [Rusmevichientong et al. 01] encounters such a host early in the walk, there is a good chance that a large fraction of the nodes are from this host, i.e., that the sample will be nonuniform.

To summarize, the open problem is to find a way that provably generates a uniform random sample and that also works in practice.

3. Web Graph Modeling

As soon as web researchers started to observe properties of the web graph, they tried to come up with a model of the web graph (see [Kleinberg et al. 99, Aiello et al. 00]). Random walks on the web graph seem to converge quickly. Additionally, when looking at links between web sites, the links look quite random. Thus, trying to model the web as a random graph was an obvious step. This led to the *copy graph* model of Kleinberg et al. [Kleinberg et al. 99] and all its modifications [Kumar et al. 00, Pandurangan et al. 02]. The web graph properties that these models try to capture are the power-law[1] indegree distribution, the fact that there is a large number of small cliques, and the power-law PageRank distribution.

[1]By a *power-law* indegree distribution, we mean that the percentage of web pages with indegree d is proportional to $1/d^{\alpha}$ for some constant α and large enough d.

However, there is a very dominant property of the web graph that is not modeled by any of these earlier graphs, namely the fact that the web is mostly a 2-level structure: Each web page belongs to a host and about 75% of the hyperlinks connect pages on the same host [Bharat et al. 01]. Edges between nodes on the same host have a lot of structure: For example, each page on a host might point to the same copyright form and to the home page of the host. To the best of our knowledge, there is no model yet that models this 2-level structure together with the other properties listed above.

Furthermore, consider the *host graph* which is created by merging all nodes on the same host into one node. The resulting graph also has a power-law indegree and outdegree distribution [Bharat et al. 01]. There is also no random graph model that models the power-law distributions on the pages as well as on the host level.

In summary, the open problem is to come up with a random graph model that models the behavior of the web graph on the pages as well as on the host level.

4. Duplicate Hosts

Web search engines try to avoid having duplicate and near-duplicate pages in their collection, since such pages increase the time it takes to add useful content to the collection. Additionally, duplicate and near-duplicate pages do not contribute new information to search results and thus annoy users.

The problem of finding duplicate or near-duplicate pages in a set of crawled pages is well studied [Brin et al 95, Broder 97]. There has also been some research on identifying duplicate or near-duplicate directory trees, called *mirrors* [Bharat and Broder 99, Cho et al. 00].

While mirror detection and individual-page detection try to provide a complete solution to the problem of duplicate pages, a simpler variant can reap most of the benefits while requiring less computational resources. This simpler problem is called *duplicate host detection*: Detect two hosts that are page-by-page identical. Duplicate hosts ("duphosts") are the single largest source of duplicate pages on the web, so solving the duplicate hosts problem can result in a significant improvement.

The duplicate host detection problem is easier than mirror detection since the URLs between duphosts differ only in the hostname component. Additionally, the pages on the two hosts are exactly identical, i.e., the algorithm does not need to detect reformatting. Finally, the set of pages on the first host is identical to the set of pages on the second host. A first set of approaches to the duphosts problem was studied by Bharat et al. [Bhart et al. 00], but the error rate of

their algorithms (both for false positives and false negatives) can probably be improved. Their general approach, however, seems valuable: Represent each host by a *sketch*. For example, a sketch can be a subset of the URLs on the host or the hyperlinks pointing to the pages on the host. Then use the sketch to compare hosts. Of course, the hard questions are, what sketch to choose and how to avoid comparing all pairs of hosts. Since there are millions of different hosts, comparing all pairs is simply infeasible. Bharat et al. [Bhart et al. 00] explore sketches based only on URL strings and the hyperlink structure.

5. Data Streams

The query logs of a web search engine contain all the queries issued at this search engine. The most frequent queries change only slowly over time. However, the queries with the largest increase or decrease from one time period over the next show interesting trends in user interests. We call them the *top gainers* and *losers*. Since the number of queries is huge, the top gainers and losers need to be computed by making only one pass over the query logs. This leads to the following data stream problem: Given two sequences of items, find the items whose absolute number increases or decreases the most when comparing one sequence with the other by reading the sequence only once. Charikar et al. [Charikar et al. 02] presented a 2-pass algorithm for this problem. Another interesting variant is to find all items above a certain frequency whose relative increase (i.e., their increase divided by their frequency in the first sequence) is the largest.

6. Dense Bipartite Subgraphs

As was shown by Kumar et al. [Kumar et al. 99], the web contains many densely connected directed bipartite subgraphs because cyber-communities often have such a densely connected structure. The source nodes in such a subgraph are the "hubs" or directory nodes on the topic; the sink nodes are the "authorities" or content nodes on the topic. Kumar et al. also presented and implemented an algorithm to find small complete bipartite subgraphs, which they call *cores*. They use a bottom-up approach using the fact that every (i, i)-core is a combination of $(i-1, i-1)$ cores. However, their cores were relatively small, in the order of tens of nodes.

In order to more completely capture these cyber-communities, it would be interesting to detect much larger bipartite subgraphs, in the order of hundreds

or thousands of nodes. They do not need to be complete, but they should be dense, i.e., they should contain at least a constant fraction of the corresponding complete bipartite subgraphs. Are there efficient algorithms to detect them? And can these algorithms be implemented efficiently if only a small part of the graph fits into main memory?

7. Eigenvector–Induced Partitionings of Directed Graphs

Donath and Hoffman [Donath and Hoffman 73] introduced the use of eigenvectors for the purpose of partitioning an undirected graph in a balanced way. Since then, there has been a lot of work on spectral approaches for graph partitioning. See [Chung 97] for an excellent overview of the field. Shi and Malik [Shi and Malik 00] showed that the eigenvectors of different matrices based on the adjacency matrix of a graph are related to different kinds of balanced cuts in a graph. Let W be the adjacency matrix of an undirected graph (V, E) with nodes $1, 2, \ldots, n$, and let D be a diagonal matrix with $d_i = deg(i)$. Let A and B be sets of nodes and let $E(A, B)$ be the set of edges (a, b) with $a \in A$ and $b \in B$.

The *average association* of a set A is

$$|E(A, A)|/|A|.$$

The *average cut* of a set A is

$$|E(A, V - A)|/|A| + |E(A, V - A)|/|V - A|.$$

The *normalized cut* of a set A is

$$|E(A, V - A)|/|E(A, V)| + |E(A, V - A)|/|E(V - A, V)|.$$

Then Shi and Malik show that

- The second largest eigenvector of W is related to a set that maximizes the average association;
- The second smallest eigenvector of $D - W$ is related to a set that minimizes the average cut; and
- The second smallest eigenvector of the generalized eigenvector problem $(D - W)x = \lambda Dx$ gives an approximation of the smallest normalized cut.

These results hold for undirected graphs, but the web graph is a directed graph. Thus, it would be interesting to understand what the above relationships are for directed graphs, i.e., whether the eigenvectors of the corresponding matrices of a directed graph are also related to balanced decompositions of the directed graph. It is possible that this would lead to an interesting clustering of the web graph or for a topic-specific subgraph. A first step in this direction was taken by Gibson et al. [Gibson et al. 98]. They used the eigenvectors of the matrix AA^T and the matrix A^TA, where A is the adjacency matrix of a topic-specific subgraph, to decompose topic-specific subgraphs. They show anecdotally that the principal eigenvector and the top few nonprincipal eigenvectors decompose the topic graphs into multiple "hyperlinked communities," i.e., clusters of pages on the same subtopic.

References

[Aiello et al. 00] W. Aiello, F. Chung, and L. Lu. "A Random Graph Model for Massive Graphs." In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing,* pp. 171–180. New York: ACM Press, 2000.

[Bar-Yossef et al. 00] Z. Bar-Yossef, A. Berg, S. Chien, J. Fakcharoenphol, and D. Weitz. "Approximating Aggregate Queries about Web Pages via Random Walks." In *Proceedings of the 26th International Conference on Very Large Data Bases (VLDB)*, pp. 535–544. San Francisco: Morgan Kaufmann, 2000.

[Bharat and Broder 99] K. Bharat and A. Z. Broder. "Mirror, Mirror on the Web: A Study of Host Pairs with Replicated Content." In *Proceedings of the 8th International World Wide Web Conference,* pp. 501–512. Amsterdam: Elsevier Science, 1999.

[Bhart et al. 00] K. Bharat, A. Z. Broder, J. Dean, and M. Henzinger. "A Comparison of Techniques to Find Mirrored Hosts on the World Wide Web." *Journal of the American Society for Information Science* 31 (2000), 1114–1122.

[Bharat et al. 01] K. Bharat, B. Cheng, M. Henzinger, and M. Rühl. "Who Links to Whom: Mining Linkage between Web Sites." In *Proceedings of the IEEE International Conference on Data Mining,* pp. 51–58. Los Alamitos, CA: IEEE Press, 2001.

[Brin et al 95] S. Brin, J. Davis, and H. García-Molina. "Copy Detection Mechanisms for Digital Documents." In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 398–409. New York: ACM Press, 1995.

[Broder 97] A. Z. Broder. "On the Resemblance and Containment of Documents." In *Proceedings of Compression and Complexity of Sequences*, pp. 21–29. Los Alamitos, CA: IEEE Computer Society, 1997.

[Charikar et al. 02] M. Charikar, K. Chen, and M. Farach-Colton. "Finding Freqeunt Items in Data Streams." In *Proceedings of the 29th International Colloquium on Automata, Languages, and Programming*, pp. 693–703. New York: Springer, 2002.

[Chung 97] F. R. K. Chung. "Spectral Graph Theory." In *CBMS Regional Conference Series in Mathematics*, Volume 92. Providence, RI: American Mathematical Society, 1997.

[Cho et al. 00] J. Cho, N. Shivakumar, and H. Garcia-Molina. "Finding Replicated Web Collections." In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 355-366. New York: ACM Press, 2000.

[Donath and Hoffman 73] W. E. Donath and A. J. Hoffman. "Lower Bounds for the Partitioning of Graphs." *IBM Journal of Research and Development* 17 (1973), 420–425.

[Gibson et al. 98] D. Gibson, J. Kleinberg, and P. Raghavan. "Inferring Web Communities from Link Topology." In *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia*, pp. 225–234. New York: ACM Press, 1998.

[Henzinger et al. 00] M. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork. "On Near-Uniform URL Sampling." In *Proceedings of the 9th International World Wide Web Conference*, pp. 295–308. Amsterdam: Elsevier Science, 2000.

[Kleinberg et al. 99] J. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. S. Tomkins. "The Web as a Graph: Measurements, Models, and Methods." In *Proceedings of the International Conference on Combinatorics and Computing,* pp. 1–17. New York: Springer, 1999.

[Kumar et al. 00] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. "Stochastic Models for the Web Graph." In *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science,* pp. 57–65. Los Alamitos, IEEE Press, 2000.

[Kumar et al. 99] S.R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. "Trawling the Web for Emerging Cyber-Communities." In *Proceedings of the 8th International World Wide Web Conference*, pp. 11–16. Amsterdam: Elsevier Science, 1999.

[Lawrence and Giles 99] S. Lawrence and C. L. Giles. "Accessibility of Information on the Web." *Nature* 400 (1999), 107-109.

[Pandurangan et al. 02] G. Pandurangan, P. Raghavan, and E. Upfal. "Using PageRank to Characterize Web Structure." To appear in the *8th Annual International Computing and Combinatorics Conference*, 2002.

[Rusmevichientong et al. 01] P. , D. M. Pennock, S. Lawrence, and C. L. Giles. "Methods for Sampling Pages Uniformly from the World Wide Web." In *Proceedings of the AAAI Fall Symposium on Using Uncertainty within Computation*, pp. 121-128, Menlo Park: AAAI Press, 2001.

[Shi and Malik 00] J. Shi and J. Malik. "Normalized Cuts and Image Segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22:8 (2000), 888–905.

Monika R. Henzinger, Google Inc., 2400 Bayshore Parkway, Mountain View, CA 94043 (monika@google.com)

Received October 20, 2002; accepted December 11, 2002