COMPUTER

www.elsevier.com/locate/comnet

On near-uniform URL sampling

Monika R. Henzinger [a,*], Allan Heydon [b], Michael Mitzenmacher [c], Marc Najork [b]

[a] Google, Inc. 2400 Bayshore Parkway, Mountain View, CA 94043, USA
[b] Compaq Systems Research Center, 130 Lytton Avenue, Palo Alto, CA 94301, USA
[c] Harvard University, Division of Engineering and Applied Sciences, Boston, MA, USA

Abstract

We consider the problem of sampling URLs uniformly at random from the Web. A tool for sampling URLs uniformly can be used to estimate various properties of Web pages, such as the fraction of pages in various Internet domains or written in various languages. Moreover, uniform URL sampling can be used to determine the sizes of various search engines relative to the entire Web. In this paper, we consider sampling approaches based on random walks of the Web graph. In particular, we suggest ways of improving sampling based on random walks to make the samples closer to uniform. We suggest a natural test bed based on random graphs for testing the effectiveness of our procedures. We then use our sampling approach to estimate the distribution of pages over various Internet domains and to estimate the coverage of various search engine indexes.

Keywords: URL sampling; Random walks; Internet domain distribution; Search engine size

1. Introduction

Suppose that we could choose a URL uniformly at random from the Web. Such a tool would allow us to answer questions about the composition of the Web using standard statistical methods based on sampling. For example, we could use random URLs to estimate the distribution of the length of Web pages, the fraction of documents in various Internet domains, or the fraction of documents written in various languages. We could also determine the fraction of Web pages indexed by various search engines by testing for the presence of pages chosen uniformly at random. However, so far, no methodology for sampling URLs uniformly, or even near-uniformly, at random from the Web has been discovered.

The contributions of this paper are threefold. First, we consider several sampling approaches, including natural approaches based on random walks. Intuitively, the problem with using a random walk in order to sample URLs from the Web is that pages that are more highly connected tend to be chosen more often. We suggest an improvement to the standard random walk technique that mitigates this effect, leading to a more uniform sample. Second, we describe a test bed for validating our technique. In particular, we apply our improved sampling approach to a synthetic random graph whose connectivity was designed to resemble that of the Web, and then analyze the distribution of these samples. This test bed may prove useful for testing other similar techniques. Finally, we apply our sampling technique to three sizable random walks of the actual Web. We then use these samples to estimate the distribution

[*] Corresponding author.

1389-1286/00/$ – see front matter © 2000 Published by Elsevier Science B.V. All rights reserved.
PII: S1389-1286(00)00055-4

of pages over Internet domains, and to estimate the coverage of various search engine indexes.

1.1. Prior work

For the purposes of this paper, the size of a search engine is the number of pages indexed by the search engine. Similarly, the size of the Web corresponds to the number of publicly accessible, static Web pages, although, as we describe in Section 4, this is not a complete or clear definition.

The question of understanding the size of the Web and the relative sizes of search engines has been studied previously, most notably by Lawrence and Giles [14,15] and Bharat and Broder [2]. Part of the reason for the interest in the area is historical: when search engines first appeared, they were often compared by the number of pages they claimed to index. The question of whether size is an appropriate gauge of search engine utility, however, remains a subject of debate [19]. Another reason to study size is to learn more about the growth of the Web, so that appropriate predictions can be made and future trends can be spotted early.

In 1995, Bray simply created (in an undisclosed way) a start set of about 40,000 Web pages and crawled the Web from them [3]. He estimated the size of the Web to be the number of unique URLs the crawl encountered.

The initial work by Lawrence and Giles used a sampling approach based on the results of queries chosen from the NEC query logs to compare relative sizes of search engines [14]. Based on published size figures, the authors estimated the size of the Web. The approach of sampling from NEC query logs leaves questions as to the statistical appropriateness of the sample, as well as questions about the repeatability of the test by other researchers. In contrast, we seek tests that are repeatable by others (with sufficient resources).

Further work by Lawrence and Giles used an approach based on random testing of IP addresses to determine characteristics of hosts and pages found on the Web, as well as to estimate the Web's size [15]. This technique appears to be a useful approach for determining characteristics of Web hosts. Given the high variance in the number of pages per host, however, and the difficulties in accessing pages from hosts by this approach, it is not clear that this technique provides a general methodology to accurately determine the size of the Web. In particular, the scalability of this approach is uncertain for future 128 bit IP-v6 addresses.

Bharat and Broder, with motivation similar to ours, suggested a methodology for finding a page near-uniformly at random from a search engine index [2]. Their approach is based on determining queries using random words, according to their frequency. For example, in one experiment, they chose queries that were conjunctions of words, with the goal of finding a single page (or a small number of pages) in the search engine index containing that set of words. They also introduced useful techniques for determining whether a page exists in a search engine index. This problem is not as obvious as it might appear, as pages can be duplicated at mirror sites with varying URLs, pages might change over time, etc. Although Bharat and Broder used their techniques to find the relative overlap of various search engines, the authors admit that their techniques are subject to various biases. For example, longer pages (with more words) are more likely to be selected by their query approach than short pages.

This paper is also related to a previous paper of ours [9], in which we used random walks to gauge the *weight* of various search engine indexes. The weight of an index is a generalization of the notion of its size. Each page can be assigned a weight, which corresponds to its importance. The weight of a search engine index is then defined to be the sum of the weights of the pages it contains. If all pages have an equal weight, the weight of an index is proportional to its size. Another natural weight measure is, for example, the PageRank measure (described below). We used the standard model of the Web as a directed graph, where the pages are nodes and links between pages represent directed edges in the natural way. With this interpretation, we used random walks on the Web graph and search-engine probing techniques proposed by Bharat and Broder [2] to determine the weight of an index when the weight measure is given by the PageRank measure. The random walks are used to generate random URLs according to a distribution that is nearly equal to the PageRank distribution. This paper extends that approach to generate URLs according to a more uniform distribution.

1.2. Random walks and PageRank

We first provide some background on random walks. Let $X = \{s_1, s_2, \ldots, s_n\}$ be a set of states. A *random walk* on X corresponds to a sequence of states, one for each step of the walk. At each step, the walk switches from its current state to a new state or remains at the current state. Random walks are usually *Markovian*, which means that the transition at each step is independent of the previous steps and depends only on the current state.

For example, consider the following standard Markovian random walk on the integers over the range $\{0 \ldots j\}$ that models a simple gambling game, such as blackjack, where a player bets the same amount on each hand (i.e., step). We assume that if the player ever reaches 0, they have lost all their money and stop, and if they reach j, they have won enough money and stop. Hence the process will stop whenever 0 or j is reached. Otherwise, at each step, one moves from state i (where i is not 0 or j) to $i+1$ with probability p (the probability of winning the game), to $i-1$ with probability q (the probability of losing the game), and stays at the same state with probability $1-p-q$ (the probability of a draw).

The PageRank is a measure of a page suggested by Brin and Page [4] that is fundamental to our sampling approach. Intuitively, the PageRank measure of a page is similar to its in-degree, which is a possible measure of the importance of a page. The PageRank of a page is high if it is linked to by many pages with a high PageRank, and a page containing few outgoing links contributes more weight to the pages it links to than a page containing many outgoing links. The PageRank of a page can be easily expressed mathematically. Suppose there are T total pages on the Web. We choose a parameter d such that $0 < d < 1$; a typical value of d might lie in the range $0.1 < d < 0.15$. Let pages $p_1, p_2, \ldots, p_k$ link to page p. Let $R(p)$ be the PageRank of p and $C(p)$ be the number of links out of p. Then the PageRank $R(p)$ of a page is defined to satisfy:

$$R(p) = d/T + (1-d)\sum_{i=1}^{k} R(p_i)/C(p_i).$$

This equation defines $R(p)$ uniquely, modulo a constant scaling factor. If we scale $R(p)$ so that the PageRanks of all pages sum to 1, $R(p)$ can be thought of as a probability distribution over pages.

The PageRank distribution has a simple interpretation in terms of a random walk. Imagine a Web surfer who wanders the Web. If the surfer visits page p, the random walk is in state p. At each step, the Web surfer either jumps to a page on the Web chosen uniformly at random, or the Web surfer follows a link chosen uniformly at random from those on the current page. The former occurs with probability d, the latter with probability $1-d$. The equilibrium probability that such a surfer is at page p is simply $R(p)$. An alternative way to say this is that the average fraction of the steps that a walk spends at page p is $R(p)$ over sufficiently long walks. This means that pages with high PageRank are more likely to be visited than pages with low PageRank.

2. Sampling-based approaches

We motivate our approach for sampling a random page from the Web by considering and improving on a sequence of approaches that clearly fail. Our approach also has potential flaws, which we discuss.

2.1. Deterministic approaches

One natural approach would be to simply try to crawl the entire Web, keeping track of all unique pages. The size of the Web prevents this approach from being effective.

Instead, one may consider crawling only a part of the Web. If one obtains a large enough subset of the Web, then perhaps a uniform sample from this subset would be sufficient, depending on the application. The question is how to obtain this sample. Notice that crawling the Web in some fixed, deterministic manner is problematic, since then one obtains a fixed subset of the Web. One goal of a random sampling approach is *variability*; that is, one should be able to repeat the sampling procedure and obtain different random samples for different experiments. A sampling procedure based on a deterministic crawl of the Web would simply be taking uniform samples from a fixed subset of the Web, making repeated experiments problematic. Moreover, it is not clear how to argue that a sufficiently large subset of the Web is representative. (Of course, the Web might change, leading to different results in different deterministic experiments, but one should not count on changes over which one has no control, and whose effect is unclear.)

2.2. Random walks with Mercator

Because of the problems with the deterministic crawling procedure, it is natural to consider randomized crawling procedures. For example, one may imagine a crawler that performs a random walk, following a random link from the current page. In the case where there are no links from a page, the walk can restart from some page in its history. Similarly, restarts can be performed to prevent the walk from becoming trapped in a cycle.

Before explaining our sampling approach, we describe our tool for performing PageRank-like random walks. We use Mercator, an extensible, multithreaded Web crawler written in Java [10,17]. We configure Mercator to use one hundred crawling threads, so it actually performs one hundred random walks in parallel, each walk running in a separate thread of control. The crawl is seeded with a set of 10,000 initial starting points chosen from a previous crawl. Each thread begins from a randomly chosen starting point. Recall that walks either proceed along a random link with probability $1 - d$, or perform a random jump with probability d (and in the case where the out-degree is 0). When a walk randomly jumps to a random page instead of following a link, it chooses a page at random from all pages visited by any thread so far (including the initial seeds).

Note that the random jumps our walk performs are different from the random jumps for the Web surfer interpretation of PageRank. For PageRank, the random Web surfer is supposed to jump to a page chosen uniformly at random from the entire Web. We cannot, however, choose a page uniformly at random; indeed, if we could do that, there would be no need for this paper! Hence we approximate this behavior by choosing a random page visited by Mercator thus far (including the seed set). Because we use a relatively large seed set, this limitation does not mean that our walks tend to remain near a single initial starting point (see Section 6 below). For this reason, we feel that this necessary approximation has a reasonably small effect.

3. Mathematical underpinnings

A problem with using the pages discovered by a random walk is that certain pages are more likely to be visited during the course of a random walk than other pages. For example, the site www.microsoft.com/ie is very likely to appear even during the course of a very short walk, because so many other pages point to it. We must account for this discrepancy in determining how to sample pages visited in the course of our random walk.

More concretely, consider a sampling technique in which we perform a random walk in order to crawl a portion of the Web, and we then sample pages from the crawled portion in order to obtain a near-uniform sample. For any page X,

$$\Pr(X \text{ is sampled}) = \Pr(X \text{ is crawled}) \cdot \Pr(X \text{ is sampled} \mid X \text{ is crawled}). \tag{1}$$

We first concentrate on finding an approximation for the first term on the right hand side. Consider the following argument. As we have already stated, the fraction of the time that each page is visited in equilibrium is proportional to its PageRank. Hence, for sufficiently long walks,

$$E(\text{number of times } X \text{ is visited}) \approx L \cdot R(X), \tag{2}$$

where L is the length of the walk.

Unfortunately, we cannot count on being able to do long walks (say, on the order of the number of pages), for the same reason we cannot simply crawl the entire Web: the graph is too large. Let us consider a page to be *well-connected* if it can be reached by almost every other page through several possible short paths. Under the assumption that the Web graph consists primarily of well-connected pages, approximation (2) is true for relatively short walks as well. (Here, by a *short walk*, we will mean about $\mathrm{O}(\sqrt{n})$ steps, where n is the number of pages in the Web graph; see Section 4 below regarding more about this assumption.) This is because a random walk in a well-connected graph rapidly loses the memory of where it started, so the short-term behavior is like its long-term behavior in this regard.

Now, for short walks, on the order of $\mathrm{O}(\sqrt{n})$ steps, we would expect most pages to appear at most once. This is similar in intuition to the birthday paradox, which states that if everyone has a random ID from a set of n IDs, you need roughly $\sqrt{n}$ people in a room before you find two people who share the same ID. Hence, for short walks,

$$\Pr(X \text{ is crawled}) \approx E(\text{number of times } X \text{ is visited}). \tag{3}$$

Combining approximations (2) and (3), we have

$$\Pr(X \text{ is crawled}) \approx L \cdot R(X). \tag{4}$$

Our mathematical analysis therefore suggests that $\Pr(X \text{ is crawled})$ is proportional to its PageRank. Under this assumption, by equation (1) we will obtain a uniform sampling if we sample pages from the crawled subset so that $\Pr(X \text{ is sampled} \mid X \text{ is crawled})$ is inversely proportional to the PageRank of X. This is the main point, mathematically speaking, of our approach: we can obtain more nearly uniform samples from the history of our random walk if we sample visited pages with a skewed probability distribution, namely by sampling inversely to each page's PageRank.

The question therefore arises of how best to find the PageRank of a page from the information obtained during the random walk. Our random walk provides us with two possible ways of estimating the PageRank. The first is to estimate $R(X)$ by what we call the *visit ratio* of the page, or $VR(X)$, which is simply the fraction of times the page was visited during the walk. That is,

$$VR(X) = \frac{\text{number of appearances of } X \text{ in the walk}}{\text{length of the walk}}.$$

Our intuition for using the visit ratio is that if we run the walk for an arbitrarily long time, the visit ratio will approach the PageRank. If the graph consists of well-connected pages, we might expect the visit ratio to be close to the PageRank over small intervals as well.

We also suggest a second possible means of estimating the PageRank of a page. Consider the graph consisting of all pages visited by the walk, along with all edges traversed during the course of the walk. We may estimate the PageRank $R(X)$ of a page by the sample PageRank $R'(X)$ computed on this sample graph. Intuitively, the dominant factor in the value $R'(X)$ is the in-degree, which is at least the number of times the page was visited. We would not expect the in-degree to be significantly larger than the number of times the page was visited, since this would require the random walks to cross the same edge several times. Hence $R'(X)$ should be closely related to $VR(X)$. However, the link information used in computing $R'(X)$ appears to be useful in obtaining a better prediction. Note that calculating the values $R'(X)$ requires storing a significant amount of information during the course of the walk. In particular, it requires storing much more information than required to calculate the visit ratio, since all the traversed edges must also be recorded. It is therefore not clear that in all cases computing $R'(X)$ will be feasible or desirable.

4. Limitations

In this section, we consider the limitations in our analysis and framework given above. In particular, we consider biases that may impact the accuracy of our approach.

It is first important to emphasize that our use of random walks as described above limits the pages that can be obtained as samples. Hence, we must clarify the set of Web pages from which our approach is meant to sample. Properly defining which pages constitute the Web is a challenging prospect in its own right. Many Web pages lie behind corporate firewalls, and are hence inaccessible to the general public. Also, pages can be dynamically created in response to user queries and actions, yielding an infinite number of potential but not truly extant Web pages.

Our crawl-based approach finds pages that are accessible only through some sequence of links from our initial seed set. We describe this part of the Web as the *publicly accessible Web*. Implicitly, we are assuming that the bulk of the Web lies in a giant component reachable from major sites such as Yahoo. Furthermore, we avoid crawling dynamic content by stripping the query component from discovered URLs, and we log only those pages whose content type is text/html.

Finally, because our random walk involves jumping to random locations frequently, it is very difficult for our random walk to discover pages only accessible though long chains of pages. For example, if the only way to reach a page N is though a sequence of links $A \to B \to \ldots \to N$, such that the only link to B is from A, and so on, then N will almost never be discovered. Hence, our technique is implicitly biased against pages that are not well-connected. If we assume that the giant component of publicly accessible pages is well-connected, then this is not a severe problem. Recent results, however, suggest that the graph structure of the Web may be more complex, with several pages reachable only by long chains of links and a large component of pages that are not reachable from the remainder of the Web [5].

We therefore reiterate that our random walk approach is meant to sample from the publicly accessible, static, and well-connected Web.

We now consider limitations that stem from the mathematical framework developed in Section 3. The mathematical argument is only approximate, for several reasons that we outline here.

- *Initial bias*. There is an initial bias based on the starting point. This bias is mitigated by choosing a large, diverse set of initial starting points for our crawl.
- *Dependence*. More generally, there is a dependence between pages in our random walk. Given a page on the walk, that page affects the probability that another page is visited. Therefore we cannot treat pages independently, as the above analysis appears to suggest.
- *Short cycles*. This is a specific problem raised by the dependence problem. Some pages that lie in closed short cycles may have the property that if they are visited, they tend to be visited again very soon. For these pages, our argument does not hold, since there is a strong dependence in the short term memory of the walk: if we see the page we are likely to see it again. In particular, this implies that approximation (3) is inaccurate for these pages; however, we expect the approximation to be off by only a small constant factor, corresponding to the number of times we are likely to visit the page in a short interval given we have visited it.
- *Large PageRanks*. Approximation (3) is inappropriate for long walks and pages with very high PageRank. For a page with very high PageRank, the probability that the page is visited is close to one, the upper bound. Approximation (4) will therefore *overestimate* the probability that a high PageRank page is crawled, since the right hand side can be larger than 1.
- *Random jumps*. As previously mentioned, our random walk approximates the behavior of a random Web surfer by jumping to a random page visited previously, rather than a completely random page. This leads to another bias in our argument, since it increases the likelihood that a page will be visited two or more times during a crawl. This bias is similar to the initial bias.

Despite these problems, we feel that for most Web pages X, our approximation (4) for $\Pr(X$ is crawled$)$ will be reasonably accurate. However, approximation (4) does not guarantee uniform samples, since there are other possible sources of error in using the visit ratio to estimate the PageRank of a page. In particular, the visit ratio yields poor approximations for pages with very small PageRank. This is because the visit ratio is discrete and has large jumps compared to the smallest PageRank values.

To see this more clearly, consider the following related example. Suppose that we have two bins, one containing red balls and the other containing blue balls, representing pages with small and large PageRanks, respectively. The balls have unique IDs and can therefore be identified. There are one million red balls and one hundred blue balls. We 'sample' balls in the following manner: we flip a fair coin to choose a bin, and we then choose a ball independently and uniformly at random from the selected bin. Suppose we collect ten thousand samples in this manner.

Let us treat this sampling process as a random walk. (Note that there are no links; however, this is a random process that gives us a sequence of balls, which we may treat just like the sequence of pages visited during a random walk.) Suppose we use the visit ratio as an approximation for the long-term sample distribution. Our approximation will be quite good for the blue balls, as we take sufficient samples that the visit ratio gives a fair approximation. For any red balls we choose, however, the visit ratio will be (at it smallest) 1 in 10,000, which is much too large, since we expect to sample each red ball once in every 2,000,000 samples.

The problem is that rarely visited pages (i.e., pages with small PageRank) cannot be properly accounted for, since over a short walk we have no chance to see the multitude of such pages. Hence, our estimate for the PageRank of pages with small PageRank is too large, and hence our estimate for the inverse of the PageRank (which we use as $\Pr(X \text{ is sampled} \mid X \text{ is crawled})$ in Eq. 1) is too small. The effect is that such pages will still be somewhat under-sampled by our sampling process. Computing $R'(X)$ in place of $VR(X)$ does not solve this problem; an analogous argument shows that pages with small PageRank are also under-sampled if this estimate is used.

The best we can hope for is that this sampling procedure provides a more uniform distribution of pages. In what follows, we describe the experiments we performed to test the behavior of our sampling procedure on a random graph model, as well as the results from random walks on the Web.

5. A random test bed

In order to test our random walk approach, it is worthwhile to have a test bed in which we can gauge its performance. We suggest a test bed based on a class of random graphs, designed to share important properties with the Web.

It has been well-documented that the graph represented by the Web has a distinguishing structure. For example, the in-degrees and out-degrees of the nodes appear to have a power-law (or Zipf-like) distribution [13]. A random variable X is said to have a power-law distribution if

$$\Pr(X = k) \sim \frac{1}{k^\alpha}$$

for some real number α and some range of k. One explanation for this phenomenon is that the Web graph can be thought of as a dynamic structure, where new pages tend to copy the links of other pages.

For our test bed, we therefore choose random graphs with in-degrees and out-degrees governed by power-law distributions. The in-degrees and out-degrees are chosen at random from a suitable distribution, subject to the restriction that the total in-degree and out-degree must match. Random connections are then made from out links to in links via a random permutation. This model does not capture some of the richer structure of the Web. However, we are primarily interested in whether our sampling technique corrects for the variety of the PageRanks for the nodes. This model provides a suitable variety of PageRanks as well as in-degrees and out-degrees, making it a useful test case.

We present results for a test graph. The probability of having out-degree k was set to be proportional to $1/k^{2.38}$, for k in the range five to twenty. The probability of having in-degree k was set to be proportional to $1/k^{2.1}$. The range of the in-degrees were therefore set to lie between five and eighteen, so that the total in-degree would be close to the total out-degree. (There are a few nodes with smaller in-degree, due to the restriction that the total in-degree and out-degree must match.) The exponents 2.38 and 2.1 were chosen based on experimental results [13]. Our final graph has 10,000,000 nodes and 82,086,395 edges. Note that we chose a relatively high minimum in-degree and out-degree to ensure that with high probability our graph is strongly connected. That is, there are no small isolated components, and every page can reach every other page through some path.

To crawl this graph, we wrote a program that reads a description of the graph and acts as a Web server that returns synthetic pages whose links correspond to those of the graph. We then used Mercator to perform a random walk on this server, just as we would run Mercator on the real Web. The walk visited 848,836 distinct nodes, or approximately 8.5% of the total graph. Three sets of two thousand samples each were chosen from the visited nodes, using three different sampling techniques. A *PR sample* was obtained by sampling a crawled page X with probability inversely proportional to its apparent PageRank $R'(X)$. Similarly, a *VR sample* was obtained by sampling a crawled page X with probability inversely proportional to its visit ratio $VR(X)$. Finally, a *random sample* was obtained by simply choosing 2000 of the crawled pages independently and uniformly at random.

One way to test the efficacy of our sampling technique is to test if the sampled nodes are uniformly distributed according to certain graph attributes that may affect which nodes we sample. In particular, it seems likely that a node's out-degree, in-degree,

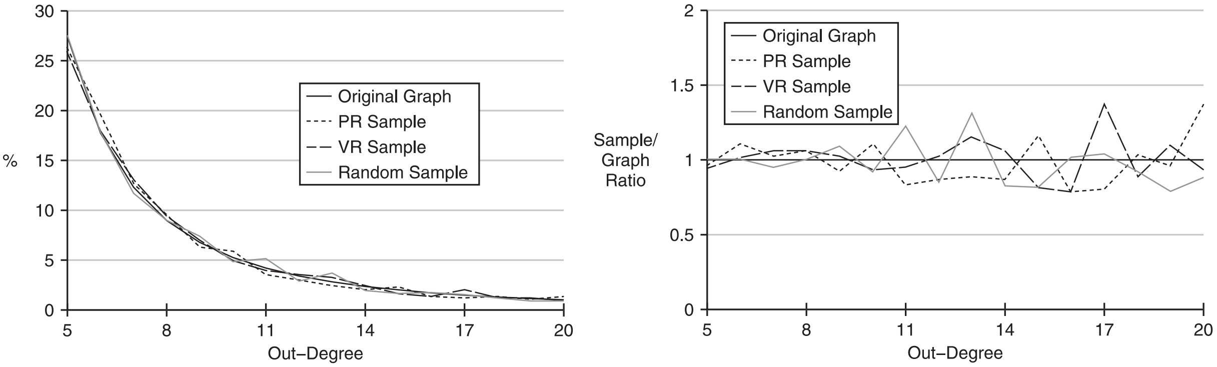

Fig. 1. Out-degree distributions for the original graph and for nodes obtained by three different sampling techniques.

and PageRank might affect how likely it is to be sampled. For example, since a node's PageRank is so closely tied to the likelihood that we crawl it, there is a good chance that the node's PageRank will be somewhat correlated with the probability that our sampling technique samples it, while this will of course not be the case if our sampling technique is truly uniform. We therefore compared the proportions of the in-degrees, out-degrees, and PageRanks of our samples with their proportions in the original graph.

For example, we consider first the out-degrees, shown in Fig. 1. The graph on the left shows the distributions of node out-degrees for the original graph and nodes collected by three sampling techniques. The graph on the right hand side normalizes these distributions against the percentages from the original graph (shown as a horizontal solid line with value 1). In both graphs, sample curves closer to the graph curve (shown in solid black) are better. Although the distributions for the samples differ somewhat from that of the original graph, the differences are minor, and are due to the variation inherent in any probabilistic experiment. As might be expected, there does not appear to be any systematic bias against nodes with high or low out-degree in our sampling process.

In contrast, when we compare our samples to the original graph in terms of the in-degree and PageRank, as shown in Figs. 2 and 3, there does appear to be a systematic bias against pages with low in-degree and low PageRank. (Note that in Fig. 3, the PageRank values are scaled as multiples of the average PageRank, namely 10^{-6}, the inverse of the number of nodes in the graph. For example, the percentage of pages in the PageRank range 1.0–1.2 corresponds to the percentage of pages whose PageRanks lie between 1.0 and 1.2 times the average.) This systematic bias against pages with low in-degree or PageRank is naturally understood from our previous discussion in Section 4. Our random walk tends to

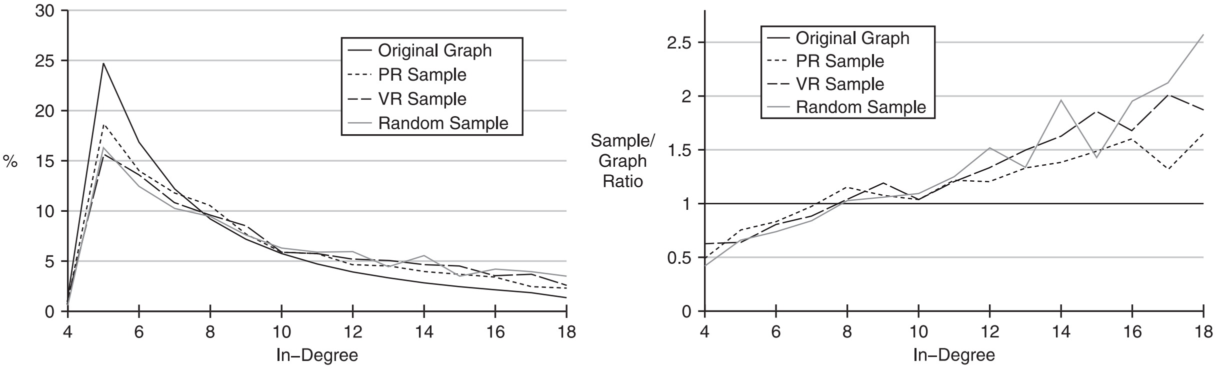

Fig. 2. In-degree distributions for the original graph and for nodes obtained by three different sampling techniques.

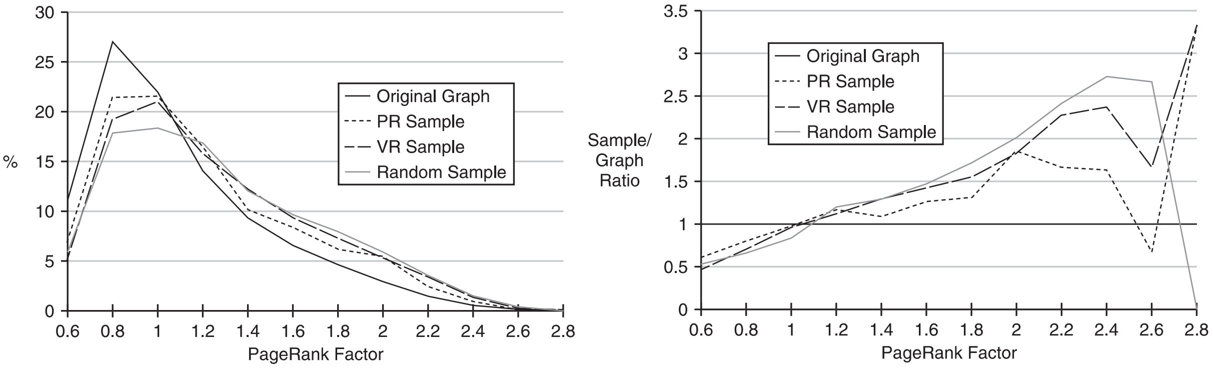

Fig. 3. PageRank distributions for the original graph and for nodes obtained by three different sampling techniques.

discover pages with higher PageRank. Our skewed sampling is supposed to ameliorate this effect but cannot completely correct for it. Our results verify this high-level analysis.

As can be seen from the right-hand graphs in Figs. 2 and 3, the most biased results for in-degree and PageRank appear in the random samples. In other words, both PR and VR sampling produces a net sampling that is more uniform than naive random sampling of the visited sub-graph.

We have similarly experimented with random graphs with broader ranges of in- and out-degrees, more similar to those found on the Web. A potential problem with such experiments is that random graphs constructed with small in- and out-degrees might contain disjoint pieces that are never sampled, or long trails that are not well-connected. Hence graphs constructed in this way are not guaranteed to have all nodes publicly accessible or well-connected. In such graphs we again find that using the values $VR(X)$ or $R'(X)$ to re-scale sampling probabilities makes the resulting sample appear more uniform. However, the results are not exactly uniform, as can be seen by comparing the distribution of in-degrees and PageRanks of the samples with those of the original graph.

6. Sampling random walks of the Web

To collect URLs for sampling, we performed three random walks of the Web that lasted one day each. All walks were started from a seed set containing 10,258 URLs discovered by a previous, long-running Web crawl. From the logs of each walk, we then constructed a graph representation that included only those visited pages whose content type was text/html. Finally, we collected PR, VR, and uniformly random samples for each walk using the algorithms described above.

Various attributes of these walks are shown in Table 1. For each walk, we give the walk's start date, the total number of downloaded HTML pages (some of which were fetched multiple times), as well as the number of nodes and (non-dangling) edges in the graph of the downloaded pages. Note that *Walk 3* downloaded pages at roughly twice the rate of the other two walks; we attribute this to the variability inherent in network bandwidth and DNS resolution.

Given any two random walks starting from the same seed set, one would hope that the intersection of the sets of URLs discovered by each walk would be small. To check how well our random walks live up to this goal, we examined the overlaps between the sets of URLs discovered by the three walks. Fig. 4 shows a Venn diagram representing this overlap. The three differently hatched regions represent the sets of URLs encountered by Walks 1, 2, and 3, respectively. The values in each region denote the

Table 1
Attributes of our three random Web walks

Name	Date	Downloads	Nodes	Edges
Walk 1	11/15/99	2,702,939	990,251	6,865,567
Walk 2	11/17/99	2,507,004	921,114	6,438,577
Walk 3	11/18/99	5,006,745	1,655,799	12,050,411

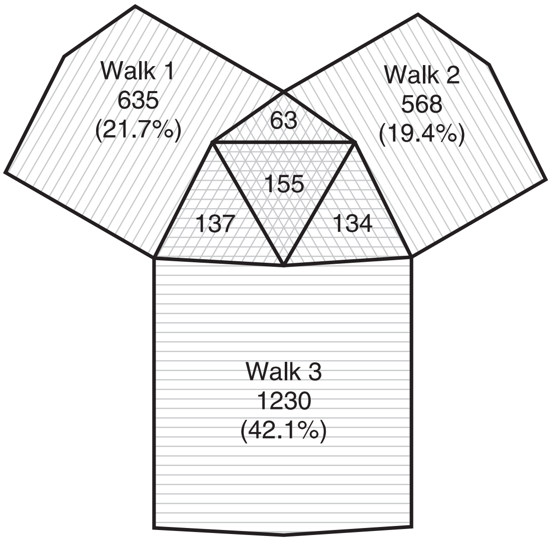

Fig. 4. Overlap of the URLs (in thousands) visited during the three walks.

number of URLs (in thousands), and the areas accurately reflect those values. The main conclusion to be drawn from this figure is that 83.2% of all visited URLs were visited by only one walk. Hence, our walks seem to disperse well, and therefore stand a good chance of discovering new corners of the Web.

7. Applications

Having a set of near-uniformly sampled URLs enables a host of applications. Many of these applications measure properties of the Web, and can be broadly divided into two groups: those that determine characteristics of the URLs themselves, and those that determine characteristics of the documents referred to by the URLs. Examples of the former group include measuring distributions of the following URL properties: length, number of arcs, port numbers, filename extensions, and top-level Internet domains. Examples of the latter group include measuring distributions of the following document properties: length, character set, language, number of out-links, and number of embedded images. In addition to measuring characteristics of the Web itself, uniformly sampled URLs can also be used to measure the fraction of all Web pages indexed by a search engine. In this section we report on two such applications, top-level domain distribution and search engine coverage.

7.1. Estimating the top-level domain distribution

We analyzed the distribution of URL host components across top-level Internet domains, and compared the results to the distribution we discovered during a much longer deterministic Web crawl that downloaded 80 million documents. Table 2 shows for each walk and each sampling method (using 10,000 URLs) the percentage of pages in the most popular Internet domains.

Note that the results are quite consistent over the three walks that are sampled in the same way. Also, as the size of the domain becomes smaller, the variance in percentages increases, as is to be expected by our earlier discussion.

There appears to be a relatively small difference between the various sampling techniques in this exercise. Although this may be in part because our skewed sampling does not sufficiently discount high PageRank pages, it also appears to be because the PageRank distributions across domains are sufficiently similar that we would expect little difference between sampling techniques here. We have found in our samples, for example, that the average sample PageRank and visit ratio are very close (within 10%) across a wide range of domains.

7.2. Search engine coverage

This section describes how we have used URL samples (using 2,000 pages) to estimate search engine coverage. For each of the URL samples produced as described in Section 6 above, we attempt to determine if the URL has been indexed by various search engines. If our samples were truly uniform over the set of all URLs, this would give an unbiased estimator of the fraction of all pages indexed by each search engine.

To test whether a URL is indexed by a search engine, we adopt the approach used by Bharat and Broder [2]. Using a list of words that appear in Web documents and an approximate measure of their frequency, we find the r rarest words that appear in each document. We then query the search engine using a conjunction of these r rarest words and check

Table 2
Percentage of sampled URLs in each top-level domain

Domain	Deterministic	Uniform sample			PR sample			VR sample		
	crawl	Walk 1	Walk 2	Walk 3	Walk 1	Walk 2	Walk 3	Walk 1	Walk 2	Walk 3
com	47.03	46.79	46.48	47.02	46.59	46.77	47.53	45.62	46.01	45.42
edu	10.25	9.01	9.02	8.90	9.31	9.36	9.13	9.84	9.08	9.96
org	8.38	8.51	8.82	8.99	8.66	8.74	8.38	9.12	8.91	8.65
net	6.41	4.80	4.52	4.39	4.96	4.63	4.62	4.74	4.50	4.52
jp	3.99	3.83	3.74	3.41	3.70	3.22	3.61	3.87	3.62	3.62
gov	2.75	2.97	3.04	2.74	3.13	3.09	2.53	3.42	3.53	2.89
uk	2.53	2.46	2.65	2.70	2.73	2.77	2.76	2.59	3.08	2.83
us	2.44	1.73	1.86	1.53	1.65	1.73	1.62	1.77	1.52	1.80
de	2.14	3.24	2.93	3.29	3.21	3.25	3.06	3.26	3.13	3.52
ca	1.93	2.07	2.31	1.94	2.13	1.85	1.86	2.05	1.89	2.07
au	1.51	1.85	1.87	1.64	1.75	1.66	1.66	1.74	1.49	1.71
fr	0.80	0.96	1.04	0.99	0.84	0.69	0.89	0.99	1.01	0.90
se	0.72	0.81	1.33	1.04	0.86	1.27	1.06	0.84	1.10	1.05
it	0.54	0.65	0.63	0.80	0.91	0.82	0.70	0.82	0.82	0.83
ch	0.37	0.87	0.71	0.99	0.64	0.71	0.87	0.92	0.72	0.89
Other	8.21	9.45	9.05	9.63	8.93	9.44	9.72	8.41	9.59	9.34

for the appropriate URL. In our tests, we use $r = 10$. Following their terminology, we call such a query a *strong query*, as the query is designed to strongly identify the page.

In practice, strong queries do not always uniquely identify a page. First, some sampled pages contain few rare words; therefore, even a strong query may produce thousands of hits. Second, mirror sites, duplicates or near-duplicates of the page, or other spurious matches can create difficulties. Third, some search engines (e.g., Northern Light) can return pages that do not contain *all* of the words in the query, despite the fact that a conjunctive query was used.

To deal with some of these difficulties, we adopt an approach similar to one suggested by Bharat and Broder [2]. In trying to match a URL with results from a search engine, all URLs are normalized by converting to lowercase, removing optional extensions such as `index.htm[l]` and `home.htm[l]`, inserting defaulted port numbers if necessary, and removing relative references of the form '`#...`'. We also use multiple matching criteria. A match is *exact* if the search engine returns a URL that, when normalized, exactly matches the normalized target URL. A *host* match occurs if a search engine returns a URL whose host component matches that of the target URL. Finally, a *non-zero* match occurs if a search engine returns any URL as a result of the strong query. Non-zero matches will overestimate the number of actual matches; however, the number of exact matches may be an underestimate if a search engine removes duplicate pages or if the location of a Web page has changed.

To measure the coverage of several popular search engines, we fetched the 12,000 pages corresponding to the URLs in the PR and VR samples of the three walks described in Section 6. We then determined the 10 rarest words in each fetched page, and performed queries on the following eight search engines: AltaVista [1], Excite [6], FAST Search [7], Google [8], HotBot [11], Infoseek [12], Lycos [16], and Northern Light [18].

The results of these experiments are shown in Figs. 5–7, which show the exact, host, and non-zero match percentages, respectively. Note that the results are quite consistent over the three walks and the two sampling methods.

An issue worth remarking on is that Google appears to perform better than one might expect from reported results on search engine size [19]. One possible reason for the discrepancy is that Google sometimes returns pages that it has not indexed based on key words in the anchor text pointing to the page. A second possibility is that Google's index may contain pages with higher PageRank than other

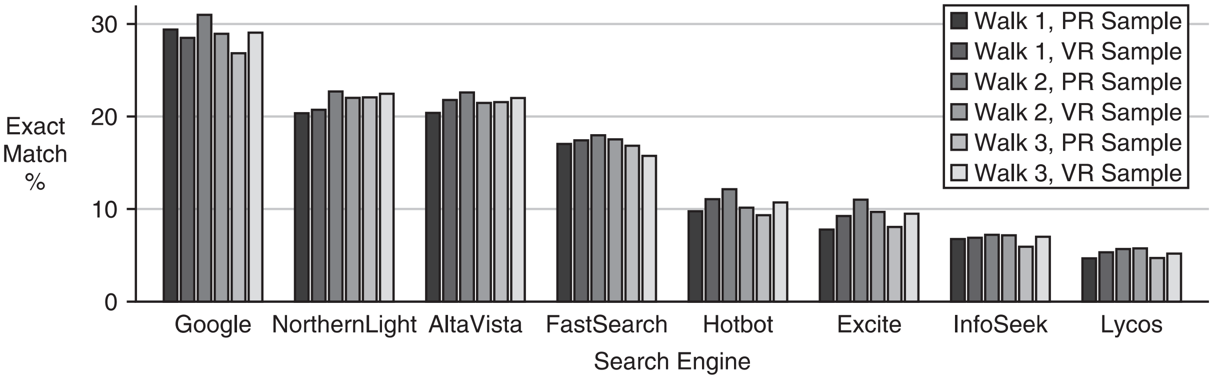

Fig. 5. Exact matches for the three walks.

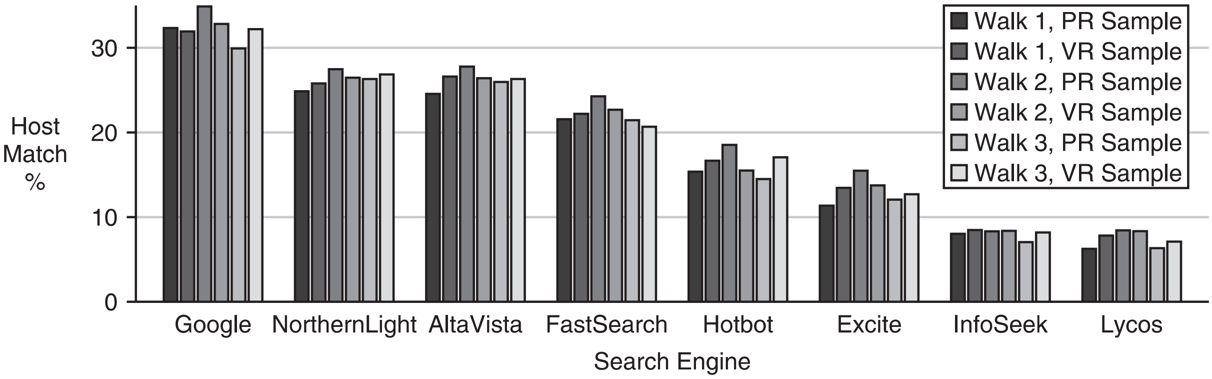

Fig. 6. Host matches for the three walks.

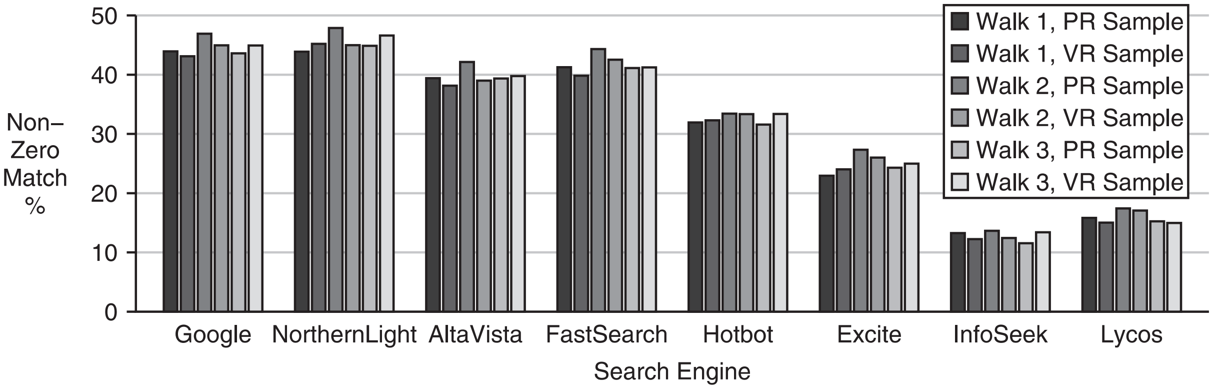

Fig. 7. Non-zero matches for the three walks.

search engines, and the biases of our approach in favor of such pages may therefore be significant. These possibilities underscore the difficulties in performing accurate measurements of search engine coverage.

8. Conclusions

We have described a method for generating a near-uniform sample of URLs by sampling URLs discovered during a random walk of the Web. It is known that random walks tend to over-sample URLs with higher connectivity, or PageRank. To ameliorate that effect, we have described how additional information obtained by the walk can be used to skew the sampling probability against pages with high PageRank. In particular, we use the visit ratio or the PageRanks determined by the graph of the pages visited during the walk.

In order to test our ideas, we have implemented a simple test bed based on random graphs with Zipfian degree distributions. Testing on these graphs shows that our samples based on skewed sampling probabilities yield samples that are more uniform over the entire graph than the samples obtained by sampling uniformly over pages visited during the random walk. Our samples, however, are still not uniform.

Currently, we have focused attention on making our approach universal, in that we do not take advantage of additional knowledge we may have about the Web. Using additional knowledge could significantly improve our performance, in terms of making our samples closer to uniform. For example, we could modify our sampling technique to more significantly lower the probability of sampling pages with apparently high PageRank from our random walk, and similarly we could significantly increase the probability of sampling pages with apparently low PageRank from our random walk. Our sampling probabilities could be based on information such as the distribution of in-degrees and out-degrees on the Web. However, such an approach might incur other problems; for example, the changing nature of the Web makes it unclear whether additional information used for sampling can be trusted to remain accurate over time.

References

[1] AltaVista, http://www.altavista.com/.

[2] K. Bharat and A. Broder, A technique for measuring the relative size and overlap of public Web search engines, in: Proc. of the 7th International World Wide Web Conference, Brisbane, Australia, Elsevier, Amsterdam, 1998, pp. 379–388.

[3] T. Bray, Measuring the Web, World Wide Web J. 1 (3) (1996).

[4] S. Brin and L. Page, The anatomy of a large-scale hypertextual Web search engine, in: Proc. of the 7th International World Wide Web Conference, Brisbane, Australia, Elsevier, Amsterdam, 1998, pp. 107–117.

[5] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins and J. Wiener, Graph structure in the Web, Comput. Networks 33 (2000) 309–320, this volume.

[6] Excite, http://www.excite.com/.

[7] FAST Search, http://www.alltheweb.com/.

[8] Google, http://www.google.com/.

[9] M. Henzinger, A. Heydon, M. Mitzenmacher and M. Najork, Measuring search engine quality using random walks on the Web, in: Proc. of the Eighth International World Wide Web Conference, Elsevier, Amsterdam, 1999, pp. 213–225.

[10] A. Heydon and M. Najork, Mercator: a scalable, extensible Web crawler, World Wide Web 2 (4) (1999) 219–229.

[11] HotBot, http://www.hotbot.com/.

[12] Infoseek, http://www.infoseek.com/.

[13] S.R. Kumar, P. Raghavan, S. Rajagopalan and A. Tomkins, Extracting large scale knowledge bases from the Web, in: IEEE International Conference on Very Large Databases (VLDB), Edinburgh, Scotland, September 1999, pp. 639–650.

[14] S. Lawrence and C.L. Giles, Searching the World Wide Web, Science 280 (536) (1998) 98.

[15] S. Lawrence and C.L. Giles, Accessibility of information on the Web, Nature 400 (1999) 107–109.

[16] Lycos, http://www.lycos.com/.

[17] Mercator Home Page, http://www.research.digital.com/SRC/mercator/.

[18] Northern Light, http://www.northernlight.com/.

[19] Search Engine Watch, http://www.searchenginewatch.com/reports/sizes.html.

Monika R. Henzinger received her Ph.D. from Princeton University in 1993 under the supervision of Robert E. Tarjan. Afterwards, she was an assistant professor in Computer Science at Cornell University. She joined the Digital Systems Research Center (now Compaq Computer Corporation's Systems Research Center) in 1996. Since September 1999 she is the Director of Research at Google, Inc. Her current research interests are information retrieval on the World Wide Web and algorithmic problems arising in this context.

Allan Heydon received his Ph.D. in Computer Science from Carnegie Mellon University, where he designed and implemented a system for processing visual specifications of file system security. In addition to his recent work on Web crawling, he has also worked on the Vesta software configuration management system, the Juno-2 constraint-based drawing editor, and algorithm animation. He is a senior member of the research staff at Compaq Computer Corporation's Systems Research Center.

Michael Mitzenmacher received his Ph.D. in Computer Science from the University of California at Berkeley in 1996. He then joined the research staff of the Compaq Computer Corporation's

Systems Research Center. Currently he is an assistant professor at Harvard University. His research interests focus on algorithms and random processes. Current interests include error-correcting codes, the Web, and distributed systems.

Marc Najork is a senior member of the research staff at Compaq Computer Corporation's Systems Research Center. His current research focuses on 3D animation, information visualization, algorithm animation, and the Web. He received his Ph.D. in Computer Science from the University of Illinois at Urbana-Champaign in 1994, where he developed Cube, a three-dimensional visual programming language.