Google员工的论文

Keeping a Search Engine Index Fresh: Risk and optimality in estimating refresh rates for web pages

D. Ford, C. Grimes, and E. Tassone
Google
1600 Amphitheatre Pkwy
Mountain View, CA 94043

Abstract

Search engines strive to maintain a "current" repository of all web pages on the internet to index for user queries. However, refreshing all web pages all the time is costly and inefficient: many small websites don't support that much load, and while some pages update content very rapidly, others don't change at all. As a result, estimated frequency of change is often used to decide how frequently a web page needs to be refreshed in an offline corpus.

Here we consider a Poisson process model for the number of state changes of a page, where a crawler samples the page at some known (but variable) time interval and observes whether or not the page has changed in during that interval. We examine this model in a practical setting where the set of web pages for which rates need to be estimated is constantly evolving, where new pages enter the set and estimation must begin immediately. We first estimate the rate of change for the observed intervals using a Maximum Likelihood estimator described in Cho and Garcia-Molina (2000) and adapted in Grimes and Ford (2008), and then choose subsequent intervals to refresh the page based on the current estimate.

In this setting, choosing the subsequent crawl intervals trades off between increasing the number of crawls invested in a single page and the increasing the probability that the page will be stale in the repository. In addition, the rate at which the estimator converges to the true rate of change depends on the distribution of crawl intervals. Here we parametrize the relative cost of a page being stale versus the cost of an individual crawl, and define an optimal linear shrinkage of subsequent crawl intervals with respect to the expected total cost. We also demonstrate that varying the optimal factor can be used to bound the probability of a very stale page appearing in the repository.

全部更新费时&没效率
1. 小站扛不住
2. 有些更新很快
3. 有的不更新

后续的爬取间隔在（在一个单页上增加的抓取）和（这个页面在库里保持不变的概率的增加）之间权衡。
而且，速率的估计值最终收敛在爬取间隔上的分布

Keywords: Poisson process, Empirical Bayes, Loss function, Rate of change estimation, Risk

1 Introduction

Search engines crawl the web to download a corpus of web pages to index for user queries. Since websites update all the time, the process of indexing new or updated pages requires continual refreshing of the crawled content. In order to provide useful results for time-sensitive or new-content queries, a search engine would ideally maintain a perfectly up-to-date and comprehensive copy of the web at all times. This goal requires that a search engine acquire new links as they appear and refresh any new content that appears on existing pages.

As the web grows to many billions of documents and search engine indexes scale similarly, the cost of re-crawling every document all the time becomes increasingly high. One of the most efficient ways to maintain an up-to-date corpus of web pages for a search engine to index is to crawl pages preferentially based on their rate of content update [5].

最佳情况是根据人家的更新速度去抓取—理想状况

If a crawler had perfect foresight, it could download each page immediately after the page updates, and similarly acquire new pages as they appear. However, no complete listing or update schedule exists today, although some limited collection of new-content sites, such as news pages, provide a feed of such structured data to search engines. For most of the web, a crawler must attempt to estimate the correct time scale on which to sample a web page for new content. The most common model for estimating expected rates of change has been based on the assumption that changes arrive as a Poisson process. However, this model also implies that a crawler cannot deterministically predict the correct time to resample a page. Instead, based on the estimated average time between changes, a crawler can choose an interval between refreshes that achieves some benchmark for freshness using the probability that the page will change during the time between refresh crawls.

In this paper, we investigate a specific setting for estimating web page change rates: where new pages are added to the web corpus regularly, and estimation must begin immediately to schedule the page for refresh crawling. The estimator used for average rate of change must be able to handle cases with virtually no observations in a way that converges to the correct estimate. In the initial steps of estimation, however, the risk of a very stale page increases due to the possibility that the estimator is incorrect. First, we develop a total cost for any subsequent crawl interval when the true average time between changes is known, using two parameters to describe the cost of an additional crawl and the cost of the risk of the page being stale for one time unit. We demonstrate that for a known rate of change, an optimal linear scaling can be chosen to minimize the total cost for a fixed set of parameters. Second, we show that in the unknown case, the optimal linear scaling can be adapted to bound the probability that a page is stale over a long period of time.

提出的新方法

2 Related Work

2.1 The Evolution of a Web Page

Several studies have been done on evolution of web pages over reasonably long time periods, in particular Cho and Garcia-Molina [2] and Fetterly et al [8], [7]. Cho and Garcia-Molina downloaded over 700,000 pages from a set of popular webservers on a daily basis over 4 months, and then compared the MD5 checksum of the page to previous versions of the page. Around 23% of pages over all domains in their study changed with an average frequency of one day (the minimum granularity measured), and an additional 15% changed within a week. Fetterly, et al.[8] examined a much larger number of web pages (151M) sampled from a breadth-first crawl and re-fetched approximately once a week over 10 weeks. To compare page content, they used a more sophisticated similarity metric based on a set of "shingles" of the syntactic content of the page. For each page, information about the start and duration of the download, the HTTP status code, and other details were recorded.

Another important web phenomenon appeared in the Fetterly et al [8] study: a set of web pages served from the same webserver that seemed to have different content, based on automatic content change detection, but in fact formed of a set of disjoint adult phrases that updated frequently. This type of "apparent" change is an extreme case of a persistent problem in assessing meaningful rates of change in page content. In this case, the content of the page actually changed (such content as there was), but in other cases the same automatically generated text problem may appear in surrounding content such as ads, links to

commercial pages, or tables of contents. In all of these cases, the page appears to change, but the new content may not need to be crawled from a user perspective. These two studies focused primarily on detected changes in the content of the page – either the raw file content or the text properties of the document, and in the status of the page. Other studies have used the "last-modified" date as recorded from the web server [6], [10] or changes in specific pieces of information on the page such as telephone numbers, HREF links, IMG tags or etc [1]. More recently, the question of information change has been treated as separate from content change [11], [12] by examining changes in link content (instead of page content) or persistent information change. In this work, we assume that a binary change detection mechanism has been chosen, and focus primarily on the statistical issues of estimation, rather than the method of identifying changes.

2.2 Models for Rate of Change

Given that an understanding of when a page changes is important to maintaining a fresh corpus efficiently, but that many pages do not change frequently, several other works have sought to accurately estimate the average time between page updates based on periodic or variable-time samples. In almost all cases, a page is not crawled at the granularity at which changes could potentially occur, and therefore a fixed sample rate may be asymptotically biased for a given actual rate of change. Cho and Garcia-Molina [4] describe this problem of bias, and compute an asymptotically less biased estimator of rate of change for regularly sampled data. The same work also defines a Maximum Likelihood Estimator for the case where data is sampled irregularly, a question of practical importance for search engine crawlers because, as Fetterly et al [8] observed in their work, the delay between crawl request and page acquisition can vary widely for different web servers and for different retry policies by the crawler. This estimation model depends heavily on the assumption that the number of page changes which arrive in any time interval for a given page is a Poisson process, with a stationary mean that can be estimated from data. Matloff [10] derives an estimator similar to the MLE of Cho and Garcia-Molina but with lower asymptotic variance, and extends the work in a different direction: toward cases where the underlying distribution of changes is not Poisson.

The Poisson model forms an important underlying element of these estimates. Fetterly et al [8], O'Brien and Grimes [11], and several others test this assumption for their chosen change measurements. While some exceptions are observed, for example, pages with perfectly regular automated updates or pages with more likely updates during local day time (see [11]), most previous work shows only small populations ($< 5\%$) of pages that are not consistent with the Poisson model.

不到5%的页面才不符合 Poisson model

Both Cho and Garcia-Molina and Matloff focus primarily on smooth distributional models in the case where some training period of data has been observed. By contrast, our methods consider the case where a new page is added to the awareness of the crawler without any previous record, and the page must be scheduled for refresh in a sensible way. Because new pages created on the web frequently contain links to other new content, or indicate a hot new topic, crawling these pages correctly from the start is important for user facing content as well as acquisition of new content.

侧重解决：当一个全新的网页被抓取到之后，这个页面的后续更新问题。

前人解决的是一个页面已经抓到了，什么时候再去抓一次

3 The Cost of Refresh Failures

Even if the average time between changes is known, there still exists a decision problem of when to refresh the page. Allowing the page to go longer between refresh cycles increases the risk that the saved version will not match the live version (the page becomes "stale"),

while crawling the page more often increases the required bandwidth and processing to maintain the web corpus.

3.1 A Total Cost function

For the i-th web page in the crawl, the number of changes in the jth crawl interval, of length t_{ij}, is $x_{ij} \sim Poisson(\lambda_i t_{ij})$. The average time between changes, or change-period, is $\Delta_i = 1/\lambda_i$. We now consider recrawling the page on a fixed interval, t_i.

Under the Poisson model, the probability that the page is unchanged after time t is e^{-t/Δ_i}. Integrating this over a time period t_i and dividing by the total time t_i gives the expected fraction of the time interval $[0, t_i]$ that the saved page matches the live version:

$$E(match|\Delta_i, t_i) = (\frac{\Delta_i}{t_i}) \times (1 - e^{-t_i/\Delta_i})$$

This formulation has the key element that the expected fraction of time when the pages match can be considered as a function, $r_i = \frac{t_i}{\Delta_i}$, of the fraction of the average time between changes, Δ_i, and the crawl interval t_i. We introduce two externally fixed costs: C_c = the cost of a single crawl, and C_s = the cost of allowing the page to be stale over one hour. The resulting total average cost per hour over the time interval $[0, t_i]$ is thus

$$C_{(total)} = P(stale) \times C_s + P(re-crawl) \times C_c = (1 - \frac{1}{r_i} \times (1 - e^{-r_i})) \times C_s + \frac{1}{r_i} \times \frac{C_c}{\Delta_i}$$

Figure 1 shows a simulation of the total cost given the parameters $C_c = C_s = 1$, and a true $\Delta = 24$ hours. The green line represents the analytic cost computed by the formula above, and the red dotted line is the mean total cost over the simulated data. The two lines align perfectly, both with a minimum total cost at a crawl interval equal to $0.3 \times \Delta$.

3.2 Choosing an 'optimal' interval

Given a particular values for C_c, C_s and Δ_i, we wish to choose t_i so as to minimize the average cost per hour.

Differentiating $(1 - \frac{1}{r_i} \times (1 - e^{-r_i})) \times C_s + \frac{1}{r_i} \times \frac{C_c}{\Delta_i}$ with respect to r_i and setting it equal to zero gives:

$$0 = C_s \frac{1}{r_i^2} - C_s \left(\frac{1}{r_i^2} e^{-r_i} + \frac{1}{r_i} e^{-r_i} \right) - \frac{1}{r_i^2} \frac{C_c}{\Delta_i}$$

Canceling a factor of $\frac{1}{r_i}$ and rearranging gives:

$$e^{r_i} = (1 + r_i) \frac{C_s}{C_s - C_c/\Delta_i} = (1 + r_i) \frac{1}{1 - C_c/(C_s \Delta_i)}$$

Notice that this implies there is no finite positive solution for r_i when $C_c \geq C_s \Delta_i$, because $e^r = 1 + r + \frac{r^2}{2!} + \frac{r^3}{3!} + \ldots$.

For example, in the case $C_s = C_c = 1$ and $\Delta_i \leq 1$ the page goes stale so quickly that the value gained in freshness from re-crawling the page is never enough to offset the cost of that re-crawl. This is reminiscent of the result in [3] where, under an assumption of limited total crawl capacity, "To improve freshness, we should penalize elements that change too often" even to the point of never refreshing some pages which change too quickly. This behavior is, of course, contingent on our choice of loss function: pages are either fresh of stale and all stale pages are equally bad, regardless of how many times they have changed.

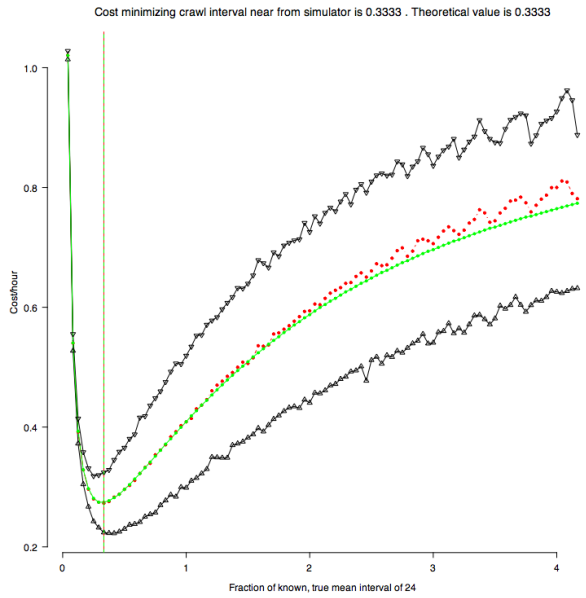

Fig. 1: Comparison of the modeled total cost per hour (green) with mean of simulated cost per hour (red) as a function of crawl interval length relative to Δ, for $\Delta = 24$ and $C_c = C_s = 1$.

Also notice that for a given change period Δ_i the fractional crawl time r_i is, unsurprisingly, a function of C_c/C_s.

In Figure 2, the colors represent the optimal r_i for each combination of C_c and C_s when $\Delta = 24$, by iteratively finding the minimum total cost. The contour lines show the levels where the optimal r_i is constant. For example, $r_i = 0.8$ is the minimum cost solution in the case where the cost of a single crawl is approximately four times the cost of an hour of staleness. By contrast, if $C_s = 4 \times C_c$, the minimum cost ratio is about $r_i = 0.15$. The solutions to the optimal cost above match those derived as a solution to the formula above.

3.3 Optimal solutions for varying Δ

As Δ varies, the optimal re-crawl ratio also varies. Based on our optimal formulation for r_i given fixed costs of crawl and staleness, the right choice of r_i quickly becomes very small as Δ get larger. This effect is primarily due to the amortization of crawl costs over longer change intervals. For example, if the true Δ of a page is 200 hours, the expected crawls per hour from crawling the page at $r = 1$ is 1/200, and unless the crawl cost, C_c, is much larger than the cost of being stale for an hour, the risk of being stale will dominate the optimization. This result makes sense intuitively: for a page that changes rarely, crawls that are "frequent" with respect to the time between changes are still quite cheap on average per hour.

Figure 3 demonstrates this effect for different combinations of C_s and C_c over a wide range of Δ. Each line in the plot represents the optimal r for a different combination of cost parameters, as the true Δ varies. The blue line shows the case where the two cost parameters are equal. As C_c is relatively larger than C_s (green line), the optimal r increases for all values of Δ.

The method here correctly chooses a subsequent crawl interval to balance external costs

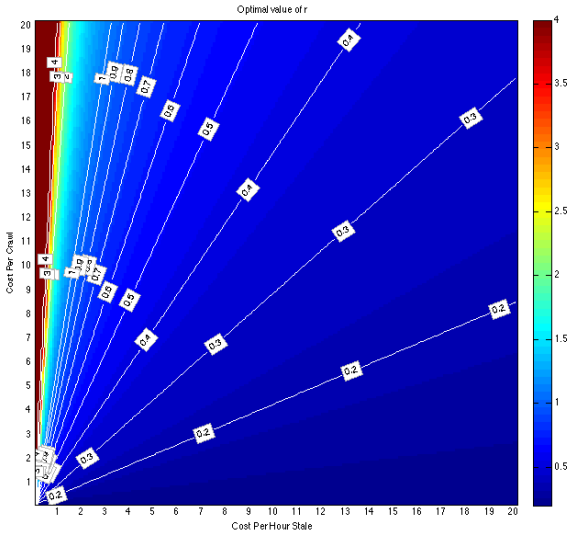

Fig. 2: Cost-minimizing crawl interval, expressed as the ratio r_i for fixed $\Delta_i = 24hours$.

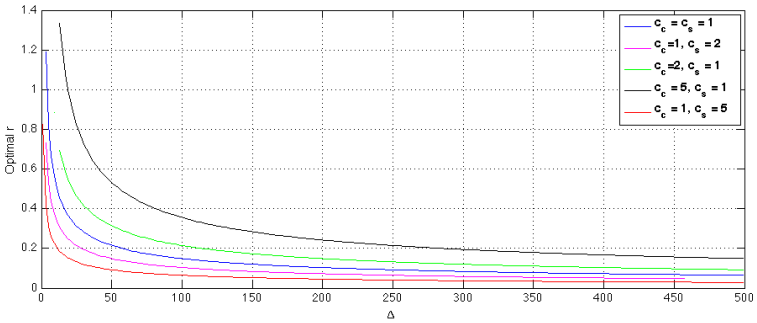

Fig. 3: Cost minimizing interval as a ratio of crawl interval to true change period.

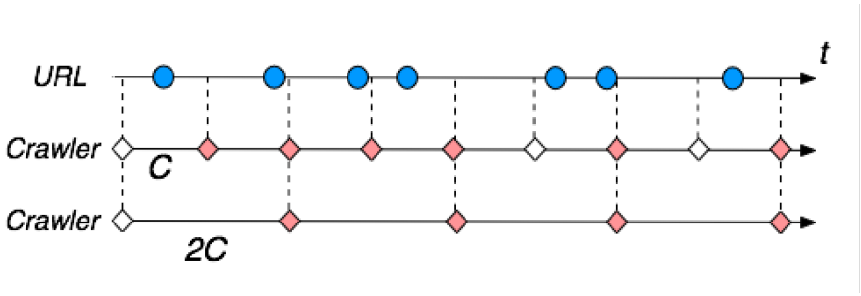

Fig. 4: Illustration of two different fixed crawl cycles over time, where each blue circle is a page update, and a filled diamond is an observation of $z_{ij} = 1$

based on a known distribution of changes. The more practical question is whether these results will hold in the case where the estimator is simultaneously attempting to converge to the true value while choosing optimal subsequent intervals to maintain freshness of the corpus.

3.4 Asymptotic Behavior

Figure 3 visually suggests that the optimal ratio of crawl interval to Δ flattens out as Δ grows larger, and therefore that there could be an asymptote where the optimal interval would continue to grow linearly with Δ. Does this hold as Δ increases? What is the behavior of the optimal interval as $\Delta_i \to \infty$?

Let $z = C_c/(C_s\Delta_i)$, which goes to 0, so that $e^r = (1+r)\frac{1}{1-C_c/(C_s\Delta_i)} = (1+r)\frac{1}{1-z} \simeq (1+r)(1+z)$. Since $r \to 0$ we may also approximate $e^r \simeq 1 + r + \frac{r^2}{2}$ to get $1 + r + \frac{r^2}{2} \simeq 1 + r + z(1+r)$. This gives $r \simeq \sqrt{2z}$. Substituting back gives an optimal crawl interval as $\Delta_i \to \infty$ of

$$t_i \simeq \sqrt{2\Delta_i \frac{C_c}{C_s}}$$

The optimal crawl interval grows as the square root of the change interval as the change interval gets large. The intuitive reason for this is that the cost of a crawl is amortized over longer and longer time periods, reducing it's cost relative to the cost of staleness.

4 Estimators for Rate of Change

In the previous section, we considered the problem of deciding how often to crawl when the true rate of change was known from the beginning. However, in a practical setting, there is always a period where a page has been introduced to the corpus but the rate of change has not been accurately estimated.

4.1 Definitions

In the data observed by a crawler, we observe only $z_{ij} = Indicator_{x_{ij}>0}$, due to the fact that the crawler can only discern that the re-sampled page matches or does not match the page collected in the previous sample. Any additional changes during the time window between samples are invisible to the crawler. As a result, the collected data is censored no occurrence counts greater than 1 can be accurately observed for a single interval.

4.2 Simple Estimators

The simplest way to approach this problem is to assume that the crawler samples each page at equal intervals every time, such that the cij are equal for all i and all j. The nave estimator for λ [13] is $\hat{\lambda_i} = (\sum_j z_{ij})/(\sum_j t_{ij})$. Figure 1 shows the problem with the nave estimator under censored data. The first row of the diagram shows the updates of a single URL over time, and the second two rows show crawl arrivals of crawlers that sample at a fixed interval of C and $2C$ respectively over a total time of $T = 8C$. Given complete data, we would estimate $\hat{\lambda_i} = 7/T$. However, given censored data, Crawler 1 using the nave estimate will estimate $\hat{\lambda_i} = 6/T$, and Crawler 2 will estimate $\hat{\lambda_i} = 4/T$. As a result, the estimates are persistently biased with respect to the true value as a function of the true value and the size of the crawl interval.

4.3 Estimators for Censored Data

Cho and Garcia-Molina (2002) derive a Maximum Likelihood Estimate (MLE) for the case of a regular crawler with interval C that has significantly smaller bias than the nave estimator for larger ratios of C/Δ and a reasonably small sample size. Their estimator is created in terms of the ratio of C/Δ and is given by

$$\hat{r_i} = \frac{\Delta_i}{C} = -log\big(\frac{\sum_j (1 - z_{ij}) + 0.5}{n + 0.5}\big), \tag{1}$$

where n is the number of intervals of length C observed over the total time.

Cho and Garcia-Molina (2002) also propose an irregular sample MLE for the case of a fixed set of training samples on each page. For this estimate, we denote the length of crawl intervals where a change was observed as $t_{i,c(l)}$ for $l = 1, L$ (the total number of changed intervals observed), and $t_{i,u(k)}$ for $k = 1, K$ are the set of intervals for the ith page where no change was observed. Then solving for λ_i:

$$\sum_{l=1}^{L} \frac{t_{i,c(l)}}{e^{\lambda_i t_{i,c(l)}} - 1} = \sum_{k=1}^{K} t_{i,u(k)}. \tag{2}$$

Asymptotic performance of this estimator depends on assumptions about the distribution of crawl intervals t_{ij}. The estimator is also undefined at two key points: if no changes have yet occurred and if changes have occurred in all sampled intervals. In these cases, Cho and Garcia-Molina recommend a sensible substitute: if the page has never changed, use the sum of all time intervals sampled so far, and if the page has always changed, use the minimum interval sampled so far.

There are two important situations where irregular samples are an important component of this estimation problem. First, practical constraints may preclude an effective web crawler from producing precisely equal sample intervals. For example, a crawler may be massively optimized, and have multiple threads competing for a fixed amount of bandwidth. Similarly, a single web server may be slow or reach a maximum allowable limit of crawler fetches. As a result, individual fetches of a web page may be delayed in an unpredictable way based on the overall system. Second, a system may deliberately vary crawl intervals in an effort to establish the best estimate of rate of change or otherwise optimize the freshness of the results. .

4.4 Estimation in an Evolving Corpus

The methods considered so far assume that estimation is being done at the end of a set of training data. However, for our problem we want to consider the case where the corpus

of web pages known by the estimator is continuously evolving. Old pages drop out of the estimation, and more importantly, new pages appear. In this setting, the distribution of pages previously observed can be used to improve estimator performance on the initial observations of a page. Grimes and Ford [9] propose a modified estimator that utilizes a prior distribution to improve the initial steps of estimation compared to the the MLE in Cho and Garcia-Molina [4] alone.

Instead of using an analytic prior, the modified estimator initializes estimation by using two pseudo-datapoints that represent the prior likelihood of the true Δ, then chooses the first crawl intervals based on the prior mean. Subsequent crawl intervals are guided by the estimated Δ at each iteration. This mechanism avoids issues that arise in the MLE when no changes (or all changes) have been observed, and allows the prior to be matched to the type of pages in the corpus easily.

For our practical estimation problem, we will use this modified estimator with a prior sample of two pseudo-datapoints, a change observed after 1 hour and no change observed after 57 hours. These two points produce the best match (using the same method employed in [9]) to the prior likelihood observed in a sample of 10,000 web pages from the Google web index.

5 What are the costs and risks of with not knowing Δ?

As developed above, in the general case we are presented with a new URL for which we do not know the true value of Δ. In this section we present simulation and some analytic results which quantify the cost of not knowing Δ, and hence having to iteratively estimate it. First, we investigate the issue of the total cost of not knowing Δ. Second, we study *when* we pay such a price in various scenarios. Finally, we look at the risk of extremely stale webpages in the case of unknown Δ.

5.1 The total cost of estimating Δ

A starting point for assessing the costs of not knowing Δ is the comparison of the overall cost for comparable cases of known and unknown Δ. Figure 5 shows the average cost per hour based on simulating $N = 1000$ observation periods of 1000 hours when $C_s = C_c = 1$, $r = \frac{1}{3}$ (so $C = 8$), and $\Delta = 24$ in the known (blue) and unknown (red) cases. Also displayed the the theoretical results (green). Two aspects of Figure 5 deserve comment. First, the lines shown all suggest the same minimizing fraction (i.e., value along the x-axis). Second, the vertical difference in the known and unknown Δ case curves may be regarded as the average per hour cost difference for the cases (in 1000 hour observation periods), and shows that we pay a price for having to estimate Δ in the unknown case.

5.2 The cost over time of estimating Δ

The next question looks at *when* these extra cost of the unknown case arises. As alluded to above, it may be that the average cost depends on the length of the observation period. Here we focus on two main scenarios, beginning with the same simulation setup as in Section 5.1. That is, the crawl fraction for both the unknown and known case is the cost-minimizing value. This is an artificial best case; in reality the cost-minimizing value would be unknown if the estimate were also unknown. However, here we use the optimal factor in simulation to demonstrate the effect of the estimation variability separate from the choice of interval scaling. The results give us a lower bound on the cost paid for estimation. In Figure 6, we ran the simulations over observation periods of 12,000 hours. We illustrate a much longer observation period to show the asymptotic behavior of the cost difference over time.

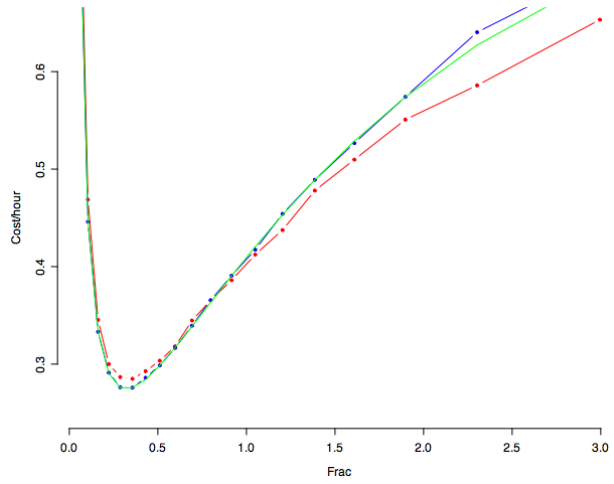

Fig. 5: The overall cost of estimating Δ. This figure shows theoretical (green), simulated known Δ (blue), and simulated unknown Δ (red) average costs per hour using the cost-minimizing fraction.

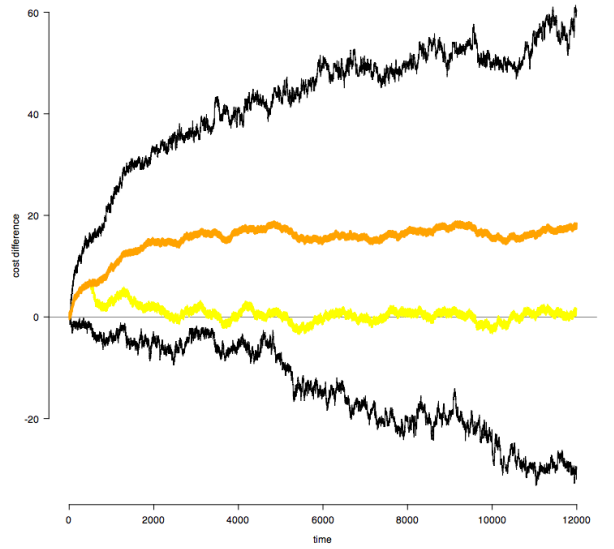

Fig. 6: Difference in cost over time for known minus unknown Δ cases, with $C_s = C_c = 1$, $r = \frac{1}{3}$, $\Delta = 24$, and $N = 200$ simulations of 12,000 hour observation periods. The orange line marks the mean of the simulations, the yellow a difference with lag=500, and the black lines the interquartile range of the simulation values.

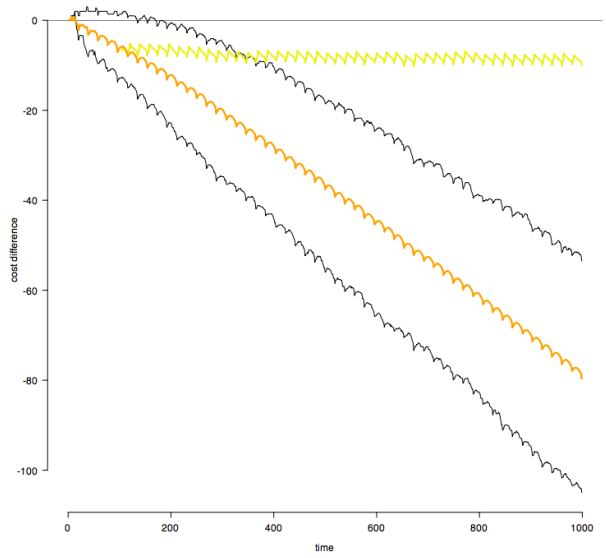

Fig. 7: Difference in cost over time for known minus unknown Δ cases, with $C_s = C_c = 1$, $r = 0.8$ for the known case and $r = \frac{1}{3}$ for the unknown case, $\Delta = 24$, and $N = 1000$ simulations of 1000 hour observation periods. he orange line marks the mean of the simulations, the yellow a difference with lag=100, and the black lines the interquartile range of the simulation values.

The flatness of the orange mean curve and the lag=500 differences shown in the yellow line are consistent with asymptotic behavior. This means that we pay a price for having to estimate Δ in the unknown case, but the convergence of the estimator eventually reduces the marginal cost (over time) to zero. The figure suggests the cost of not knowning Δ is largely borne in the first 1500 hours or so.

We next consider a different simulation setup. Here we still let $C_s = C_c = 1$ and $\Delta = 24$, but we use the suboptimal fraction 0.8 (so $C = 19.2$) for the known Δ case and the cost-minimizing fraction $r = \frac{1}{3}$ (so $C = 8$) for the unknown Δ case. Figure 7 shows that cost for the unknown Δ case with optimal fraction is less, over time, than the known Δ case with suboptimal fraction. Rather than demonstrating asymptotic behavior, as in Figure 6, we see a cost difference that increases linearly in time and favors the unknown Δ case, after an initial period in the first few hours where the known Δ case had lesser cost. Indeed, the approximate slope of this linear trend is directly attributable to the difference in expected costs given the fractions used. This result suggests that for a given observation time and with the unknown Δ case using the cost-minimizing fraction, we could find a (non-optimal) fraction for the known Δ case that would put the total costs over the period at parity.

5.3 The chance of being “very” wrong about Δ

The initial estimation of Δ introduces an increase in the average total cost that naturally disappears for individual intervals as the estimator converges. However, a more interesting question is whether the estimation process significantly increases the tail of the risk distribution, that is, does estimation significantly increase the probability of a page being very

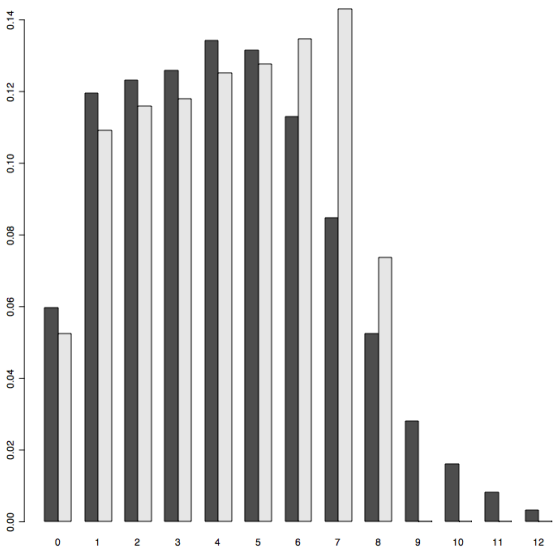

Fig. 8: Side-by-side probability histograms of observed staleness periods for known (light bars) and unknown (dark bars) Δ cases, with $C_s = C_c = 1$, $r = \frac{1}{3}$, $\Delta = 24$, and $N = 1000$ simulations of 1000 hour observation periods. Bins on the x-axis are marked by the leading digit of the staleness time (e.g., 3.23 maps to bin 3.)

stale compared to the optimal choice of crawl interval relative to the true Δ?

When Δ is known for a URL which is crawled with an unchanging fraction $r = \frac{C}{\Delta}$, the crawl interval never varies and so the maximum amount of time a page can be stale is C. However, when Δ is unknown and estimated iteratively, even when the fraction does not change[1] the time selected for the next crawl does change with evolving estimates of Δ, which can be large, thereby leading to a risk of being "very" stale that is not present in the known Δ case. Such a threshold of "very" stale might vary in different contexts, but a natural definition is the maximum amount of time a page can be stale in the known Δ case, C.[2]

In a simulation we tracked the risk of being "very" stale using this natural definition (i.e., risk of being stale C or more hours.) In particular, we let $C_s = C_c = 1$, $\Delta = 24$, and used the cost-minimzing fraction, $r = \frac{1}{3}$, for both known and unknown Δ cases with the same simulated data. We ran $N = 1000$ simulations for 1000 hours each for both cases. Figure 8 shows side-by-side probability histograms (truncated at 12 hours of staleness on the x-axis) for the known (light bars) and unknown (dark bars) cases and reveals the heavier tails of the unknown case compared with the known case. This result follows from the iterative, changing nature of the crawl times in the unknown case and the fixed 8 hour crawl time in the known case.

All mass beyond 8 hours for the unknown case falls into the "very" stale category, and for the simulation accounted for 9.1% of all crawl intervals. We note that this percentage is

[1] From Section 3.2, calculating the cost minimizing fraction requires knowledge of Δ or estimation. In the present section we use the ostensibly unknown cost-minimizing fraction to present a best-case scenario for the unknown Δ case. In practice we would suffer additional costs for uncertainty in estimating the fraction.

[2] We do not mathematically formalize this notion, but we could construct a cost function $C(m)_s$ that is linear until time m and then is significantly more costly once M is reached.

related to the rate of convergence of the estimator of Δ and declines when we look at longer time periods. For example, when using 12,000 hours of observation for each simulated run, the percentage of "very" stale crawl intervals falls to 3.8%.

Finally, we note that the known and unknown Δ cases had somewhat similar proportions of crawl intervals with no change, 28.3% for the known and 27.1% for the unknown cases, respectively (for the 1000 hour observation simulations). This probability is available analytically for the known case, and the unknown value tends to the known value for longer observation periods; for example, when using 12,000 hour observation periods the values are (again) 28.3% and 28.2% for the known and unknown cases, respectively.

6 Discussion and Extensions

The primary goal of this work is to determine optimal crawl intervals under a number of situations: while doing estimation, while fairly sure of the estimate, when crawls are expensive, and when crawls are cheap. We find that, intuitively, estimating the parameters Δ increases the probability of paying a much higher staleness cost, though estimating too low can also increase the crawl cost. We also find that the optimal ratio of crawl interval to true delta, on average, is the same in the estimation as in the known Δ case, but just with higher total cost.

In practice, bounding the probability of being overly stale due to estimation, as in Section 5.3 can be used to choose a more conservative ratio of crawl interval to reduce the risk of individual failures rather than the aggregate cost. Although an exact strategy is not discussed here, experimentation suggests that this technique can be applied effectively based on either the risk of staleness of the number of crawls should a hard constraint on total crawls per page be required.

References

[1] B. E. Brewington and G. Cybenko. How dynamic is the Web? *Computer Networks (Amsterdam, Netherlands: 1999)*, 33(1–6):257–276, 2000.

[2] J. Cho and H. Garcia-Molina. The evolution of the web and implications for an incremental crawler. In *VLDB '00: Proceedings of the 26th International Conference on Very Large Data Bases*, pages 200–209, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[3] J. Cho and H. Garcia-Molina. Effective page refresh policies for web crawlers. *ACM Transactions on Database Systems*, 28(4), 2003.

[4] J. Cho and H. Garcia-Molina. Estimating frequency of change. *ACM Trans. Inter. Tech.*, 3(3):256–290, 2003.

[5] J. Cho, H. García-Molina, and L. Page. Efficient crawling through URL ordering. *Computer Networks and ISDN Systems*, 30(1–7):161–172, 1998.

[6] F. Douglis, A. Feldmann, B. Krishnamurthy, and J. Mogul. Rate of change and other metrics: a live study of the world wide web. In *USITS'97: Proceedings of the USENIX Symposium on Internet Technologies and Systems on USENIX Symposium on Internet Technologies and Systems*, pages 14–14, Berkeley, CA, USA, 1997. USENIX Association.

[7] D. Fetterly, M. Manasse, M. Najork, and J. Wiener. A large-scale study of the evolution of web pages. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 669–678, New York, NY, USA, 2003. ACM Press.

[8] D. Fetterly, M. Manasse, M. Najork, and J. L. Wiener. A large-scale study of the evolution of web pages. *Software: Practice and Experience*, 34(2):213–237, 2004.

[9] C. Grimes and D. Ford. Estimation of web page change rates (forthcoming). In *Proceedings of the Joint Statistical Meetings, 2008*, 2008.

[10] N. Matloff. Estimation of internet file-access/modification rates from indirect data. *ACM Trans. Model. Comput. Simul.*, 15(3):233–253, 2005.

[11] S. O'Brien and C. Grimes. Microscale evolution of web pages. In *WWW '08: Proceedings of the 17th International World Wide Web Conference*, 2008.

[12] C. Olston and S. Pandey. Recrawl scheduling based on information longevity. In *WWW '08: Proceedings of the 17th International World Wide Web Conference*, 2008.

[13] H. Taylor and S. Karlin. *An Introduction to Stochastic Modeling*. Academic Press, 2004.