

Temporal Update Dynamics under Blind Sampling

Xiaoyong Li, Daren B.H. Cline, and Dmitri Loguinov

Abstract—Network applications commonly maintain local copies of remote data sources in order to provide caching, indexing, and data-mining services to their clients. Modeling performance of these systems and predicting future updates usually requires knowledge of the inter-update distribution at the source, which can only be estimated through *blind sampling* – periodic downloads and comparison against previous copies. In this paper, we first introduce a stochastic modeling framework for this problem, where the update and sampling processes are both renewal. We then show that all previous approaches are biased unless the observation rate tends to infinity or the update process is Poisson. To overcome these issues, we propose four new algorithms that achieve various levels of consistency, which depend on the amount of temporal information revealed by the source and capabilities of the download process.

I. INTRODUCTION

Many distributed systems in the current Internet manipulate objects that experience periodic modification in response to user actions, real-time events, data-centric computation, or some combination thereof. In these cases, each source (e.g., a webpage, DNS record, stock price) can be viewed as a stochastic process N_U that undergoes *updates* (i.e., certain tangible changes) after random delays U_1, U_2, \dots

Consistent estimation of inter-update distribution $F_U(x)$ is an important problem, whose solution yields not only better caching, replication [21], and allocation of download budgets [20], but also more accurate modeling and characterization of complex Internet systems [5], [7], [9], [10], [11], [19], [25], [27], [28], [29], [33], [35], [40], [42]. Similar issues arise in lifetime measurement, where U_i represents the duration of online presence for object or user i [3], [32], [36], [39].

The first challenge with measuring update-interval dynamics is to infer their distribution using *blind* sampling, where variables U_1, U_2, \dots are hidden from the observer. This scenario arises when the source can only be queried over the network using some process N_S whose inter-download delays S_1, S_2, \dots are bounded in expectation from below (e.g., due to bandwidth and/or CPU restrictions). Unlike censored observations in statistics, which have access to truncated values of *each* U_i , the sampling process here has a tendency to miss entire update cycles and land in larger-than-average intervals.

The second challenge in blind sampling is to reconstruct the distribution of U_i from severely limited amounts of information available from each download. Specifically, the observer can only compare the two most-recent copies of the source and obtain indicator variables Q_{ij} of a change occurring between

downloads i and j , for all $i < j$. This constraint is necessary because the source generally has no ability to determine object modification timestamps (e.g., dynamic webpages served by scripts are considered new on each download). Furthermore, even for static pages, object updates are very application-specific (e.g., search engines may remove ad banners and other superfluous information before indexing), which makes variables U_1, U_2, \dots *hidden not just from the observer, but also the source*.

Existing studies on this topic [2], [7], [15], [16], [22] use Poisson N_U and constant S_i . Due to the memoryless assumption on $F_U(x)$, the problem reduces to estimating just rate $\mu = 1/E[U_i]$, rather than an entire distribution, and many complex interactions between N_S and N_U are avoided in the analysis. However, more interesting cases arise in practice, where non-Poisson updates are quite common [2], [8], [17], [23]. Furthermore, guaranteeing constant S_i is impossible in certain applications where the return delay to the same object is computed in real-time and is governed by the properties of trillions of other sources (e.g., in search engines). Thus, new analytical techniques are required to handle such cases.

A. Contributions

Our first contribution is to formalize blind update sampling using a framework in which both N_U and N_S are general renewal processes. We then consider a simplified problem where the source provides last-modification timestamps for each download. Our contribution here is to develop the necessary tools for tackling the more interesting cases that follow, build general intuition, consider conditions under which provably consistent estimation is possible, and explain the pitfalls of existing methods under non-Poisson updates.

Armed with these results, we next relax the availability of last-modified timestamps at the source. For situations where constant S_i is acceptable, we show that unbiased estimators developed earlier in the paper can be easily adapted to this environment and then suggest avenues for reducing the amount of numerical computation in the model, all of which forms our third contribution. We finish the paper by considering random S_i and arrive at our last contribution, which is a novel method that can accurately reconstruct the update distribution under arbitrary $F_U(x)$ and mildly constrained $F_S(x)$.

II. RELATED WORK

Analytical studies on estimating the update distribution under blind sampling have all assumed N_U was Poisson and focused on determining its average rate, i.e., μ for stationary cases [2], [7], [15], [16], [22] and $\mu(t)$ for non-stationary [34]. Extension to general processes was achieved by [22] under the assumption that sampling intervals S_i were infinitely small;

An earlier version of the paper appeared in IEEE INFOCOM 2015.

X. Li and D. Loguinov are with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: xiaoyong@cse.com; dmitri@cse.tamu.edu). D.B.H. Cline is with the Statistics Department at Texas A&M University, College Station, TX 77843 USA (e-mail: dcline@stat.tamu.edu).

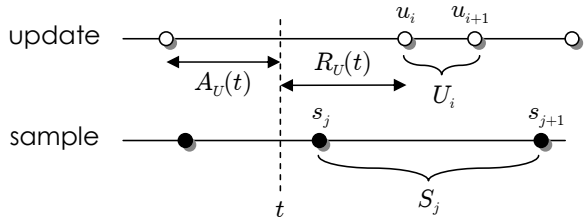


Fig. 1. Update/sample process notation.

however, the problem in these scenarios is trivial since every U_i is available to the observer with perfect accuracy.

In measurement literature, the majority of effort was spent on the behavior of web pages, including analysis of server logs [26], page-modification frequency during crawling [2], [4], [17], [23], RSS feed dynamics [33], and content change between consecutive observations [1], [14], [25]. Problems related to estimation of $F_U(x)$ have also emerged in prediction of future updates [5], [6], [13], [18], [30], [37], with a good survey in [24], and user lifetime measurement in decentralized P2P networks [3], [32], [36], [39].

III. OVERVIEW

This section introduces notation, formulates objectives, and lays down a roadmap of the studied methods.

A. Notation and Assumptions

Let u_i be the time of the i -th update at the source. Define $N_U(t) = \max\{i : u_i \leq t\}$ to be the number of updates in the time interval $[0, t]$ and suppose $U_i = u_{i+1} - u_i$ represents the inter-update delay. Similarly, denote by s_j the j -th sampling time. Let $N_S(t) = \max\{j : s_j \leq t\}$ be the number of samples in $[0, t]$ and $S_j = s_{j+1} - s_j$ be the inter-sample delay. Assume that processes (N_U, N_S) are renewal and independent of each other. This allows us to define random variables $U \sim F_U(x)$ and $S \sim F_S(x)$ to represent the lengths of update/sample cycles, respectively. Furthermore, denote by $\mu = 1/E[U]$ and $\lambda = 1/E[S]$ the corresponding rates.

At time t , define *age* $A_U(t) = t - u_{N_U(t)}$ and *residual* $R_U(t) = u_{N_U(t)+1} - t$ as the backward/forward delays to the nearest update. These are illustrated in Fig. 1. Note that interval U_i in the figure cannot be seen or measured by the observer, which is why we called it “hidden” earlier. Suppose A_U and R_U are the equilibrium versions of $A_U(t)$ and $R_U(t)$, respectively, as $t \rightarrow \infty$. From renewal theory [41], they have the same CDF:

$$G_U(x) := \mu \int_0^x (1 - F_U(y)) dy, \quad (1)$$

whose density is $g_U(x) := G'_U(x) = \mu(1 - F_U(x))$. We set the goal of the sampling process to determine the distribution $F_U(x)$ based on observations at times s_1, s_2, \dots , i.e., using a single realization of the system.

B. Applications

Knowledge of $F_U(x)$ enables performance analysis in many fields that employ lazy (i.e., pull-based) data replication. For

example, search engines implement a sampling process N_S using crawlers that periodically revisit web content and merge updates into backend databases. These companies are often concerned with staleness of pages in their index and the probability that users encounter outdated results. In order to determine the download frequency needed to maintain staleness below a certain threshold, the expected number of updates by which the index is trailing the source, or the amount of bandwidth needed for a collection of pages, accurate knowledge of source dynamics is required [20].

In another example, suppose a data center replicates a quickly changing database (driven by some update process N_U) among multiple nodes for scalability and fault-tolerance reasons. Because of the highly dynamic nature of the source, individual replicas may not stay fresh for long periods of time, but their collection may offer much better performance as a whole. In such cases, questions arise about the number of replicas k that should be queried by clients to obtain results consistent with the source [21] and/or the probability that a cluster of n replicas can recover the most-recent copy of the source when it crashes [20]. Similar problems appear in multi-hop replication and cooperative caching, where service capacity of the caching network is studied as well [21].

Finally, accurate measurement of $F_U(x)$ enables better characterization of Internet systems, their update patterns in response to external traffic, and even user behavior. While it is possible to use the exponential distribution to approximate any $F_U(x)$, as typically done in the literature [2], [7], [15], [16], [22], this can lead to significant errors in the analysis. As shown in [20] using the search-engine example and Wikipedia’s update process N_U , the exponential assumption may produce errors in the download bandwidth that are two orders of magnitude. In more complicated settings, such as cascaded and cooperative systems [21], the impact of inaccurate $F_U(x)$ may be even higher.

C. Caveats

The sample-path approach, in general, leads to a possibility of *phase-lock* where the distance of download points from the last update, i.e., $\{A_U(s_j)\}_{j \geq 1}$, is not a mixing process. For example, consider $U_i = 1$ for $i \geq 1$ and $S_j = 2$ for $j \geq 1$, in which case update ages observed at $\{s_j\}_{j=1}^\infty$ are all equal to zero. Since this case cannot be distinguished from $U_i = 0.5$ or $U_i = 2$, it is easy to see how phase-lock precludes consistent estimation of $F_U(x)$. The problem can be avoided by requiring that the considered cycle lengths exhibit certain mixing properties. This leads to our next definition.

Definition 1: A random variable X is called *lattice* if there exists a constant c such that X/c is always an integer, i.e., $\sum_{i=1}^\infty P(X/c = i) = 1$.

Lattice distributions are undesirable in our context as they produce phase-lock. Therefore, for the problem to be solvable, we must ensure the following.

Assumption 1: At least one of U and S is non-lattice.

This condition is easy to satisfy with any continuous random variable, including exponential U in previous work. A more esoteric example would be a discrete variable placing mass on two numbers whose ratio is irrational, e.g., $(\pi, 3)$ or $(e, \sqrt{2})$.

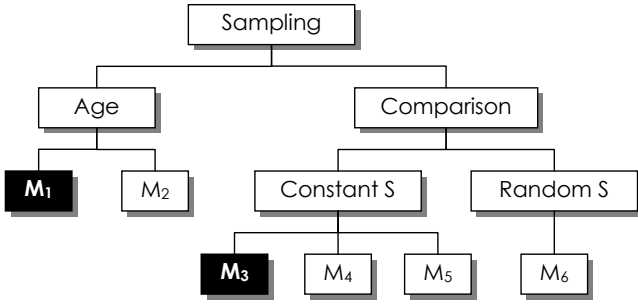


Fig. 2. Method taxonomy (shaded boxes indicate Poisson-only techniques).

D. Roadmap

As illustrated in Fig. 2, we partition the various approaches into two broad categories. In *age* sampling, the observer has access to the last-modified timestamp $u_{N_U(s_j)}$ at each download point s_j , or equivalently, the update age $A_U(s_j)$. Although now rare, this information can still be sometimes obtained from the HTTP headers, timestamps within the downloaded HTML, or sitemaps [24]. As shown in the figure, we call the two studied methods in this category M_1 and M_2 . They operate by deriving $F_U(x)$ from the collected age samples, where M_1 has been proposed in previous work [7], [22] for Poisson-only cases and M_2 is novel.

In *comparison* sampling, we assume that the observer retains the most recent copy of the object or a fingerprint of its relevant portions (e.g., after removing ads and repeated keywords). Define Q_{ij} to be an update-indicator process:

$$Q_{ij} = \begin{cases} 1 & \text{update occurs between } s_i \text{ and } s_j \\ 0 & \text{otherwise} \end{cases}. \quad (2)$$

Unlike the previous scenario, estimation of $F_U(x)$ here must use only binary values $\{Q_{ij}\}$. Going back to Fig. 2, we study comparison sampling under two strategies. For *constant S*, we first analyze two methods we call M_3 and M_4 , which are discrete versions of M_1 and M_2 , respectively. We then propose a novel method M_5 that is both consistent and computationally efficient. For *random S*, we introduce our final approach M_6 that is unbiased under the most general conditions.

IV. AGE SAMPLING

This section is a prerequisite for the results that follow. It starts with understanding state of the art in this field and its pitfalls. It then shows that a simple modification allows prior work to become unbiased under non-Poisson updates.

A. Basics

In age sampling, the observer has a rich amount of information about the update cycles. This allows reconstruction of $F_U(x)$ in all points $x \geq 0$, which we set as our goal.

Definition 2: Suppose $\tilde{F}(x, T)$ is a CDF estimator that uses observations in $[0, T]$. Then, we call it *consistent* with respect to distribution $F(x)$ if it converges in probability to $F(x)$ as the sampling window becomes large:

$$\lim_{T \rightarrow \infty} \tilde{F}(x, T) = F(x), \quad x \geq 0. \quad (3)$$

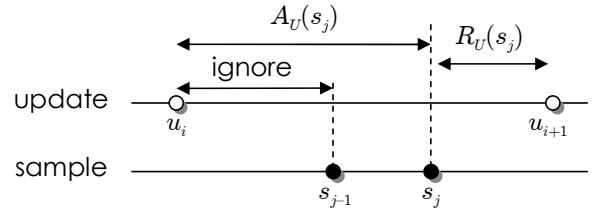


Fig. 3. Illustration of M_1 .

Note that consistent estimation of $F_U(x)$ is equivalent to that of $G_U(x)$ since there is a one-to-one mapping (1) between the two functions. Specifically, knowledge of $G_U(x)$ allows numerical differentiation and/or kernel density estimators to obtain $g_U(x) = G'_U(x)$, from which $F_U(x) = 1 - g_U(x)/g_U(0)$ follows. Furthermore, the update rate $\mu = 1/E[U]$ is also readily available as $g_U(0)$. Under Poisson N_U , the memoryless property ensures that $F_U(x) = G_U(x)$; however, in more general cases, this distinction is important.

B. Modeling M_1

To estimate the mean μ of a Poisson update process, prior studies [7], [22] proposed that only a subset of age samples $\{A_U(s_j)\}_{j \geq 1}$ be retained by the observer. Specifically, when multiple sample points land in the same update interval, only the one with the largest age is kept, while the others are discarded. As shown in Fig. 3, points s_{j-1} and s_j hit the same update cycle $[u_{i-1}, u_i]$, in which case only $A_U(s_j)$ is used in the measurement and $A_U(s_{j-1})$ is ignored. It was perceived in [7], [22] that doing otherwise would create a bias and lead to incorrect estimation, but no proof was offered. We call this method M_1 and study its performance next.

From Fig. 3, notice that M_1 collects ages $A_U(s_j)$ at such points s_j that satisfy $R_U(s_j) < S_j$, or equivalently $Q_{j,j+1} = 1$. All other age measurements are ignored. Defining $\mathbf{1}_A$ as the indicator variable of event A , the fraction of age samples retained by M_1 in $[0, T]$ is given by:

$$p(T) := \frac{1}{N_S(T)} \sum_{j=1}^{N_S(T)} \mathbf{1}_{R_U(s_j) < S_j}, \quad (4)$$

which is an important metric that determines the overhead of M_1 and its bias later in the section. Expansion of (4) in the next result follows from Assumption 1, the renewal equation for non-lattice intervals, and the law of large numbers [41].

Theorem 1: As $T \rightarrow \infty$, $p(T)$ converges in probability to:

$$p := \lim_{T \rightarrow \infty} p(T) = P(R_U < S) = E[G_U(S)]. \quad (5)$$

This result shows that p is affected not just by the update distribution $F_U(x)$, but also the sample distribution $F_S(x)$. To see this effect in simulations, we use constant and exponential S to sample Pareto $F_U(x) = 1 - (1 + x/\beta)^{-\alpha}$, where $\alpha = 3$ and $\beta = 1$ throughout the paper. Fig. 4 confirms a good match between the model and simulations. As expected, p decreases as the sampling rate $\lambda = 1/E[S]$ increases, which is caused by an increased density of points landing within each update interval and thus a higher discard rate. The figure also shows

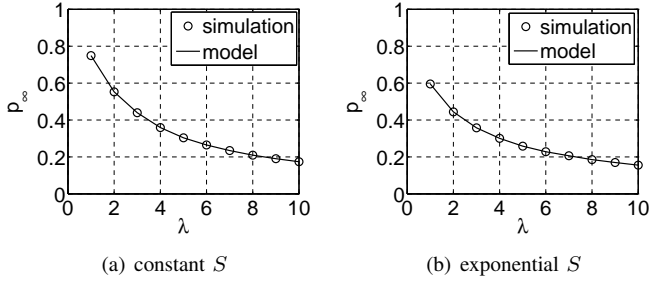


Fig. 4. Verification of (5) under Pareto U ($\mu = 2$).

that constant S samples more points than the exponential case. In fact, it is possible to prove a more general result – constant S exhibits the largest p (i.e., highest overhead) for a given λ .

Let $K(x, T)$ be the number of samples that M_1 obtains in $[0, T]$ with values no larger than x :

$$K(x, T) := \sum_{j=1}^{N_S(T)} \mathbf{1}_{R_U(s_j) < S_j} \mathbf{1}_{A_U(s_j) \leq x}. \quad (6)$$

Then, it produces a distribution in $[0, T]$ given by:

$$G_1(x, T) := \frac{K(x, T)}{K(\infty, T)}. \quad (7)$$

Theorem 2: Denoting by $\bar{F}(x) = 1 - F(x)$ the complement of function $F(x)$ and letting $T \rightarrow \infty$, the tail distribution of the samples collected by M_1 converges in probability to:

$$\bar{G}_1(x) := \lim_{T \rightarrow \infty} G_1(x, T) = \frac{E[G_U(x + S) - G_U(x)]}{E[G_U(S)]}. \quad (8)$$

Proof: Under Assumption 1 and $T \rightarrow \infty$, $A_U(s_j)$ and $R_U(s_j)$ converge to their equilibrium versions A_U and R_U , respectively. Therefore:

$$\lim_{T \rightarrow \infty} \frac{K(x, T)}{N_S(T)} = P(A_U \leq x, R_U < S). \quad (9)$$

From Theorem 1, we know that:

$$\lim_{T \rightarrow \infty} \frac{K(\infty, T)}{N_S(T)} = p = E[G_U(S)]. \quad (10)$$

Dividing (9) by (10) yields:

$$G_1(x) = \lim_{T \rightarrow \infty} G_1(x, T) = \frac{P(A_U \leq x, R_U < S)}{E[G_U(S)]}, \quad (11)$$

where $E[G_U(S)] > 0$ is guaranteed for all cases except S being zero with probability 1. To derive the numerator of (11), condition on R_U and S :

$$\begin{aligned} & P(A_U \leq x, R_U < S) \\ &= \int_0^\infty \left[\int_0^z P(A_U \leq x | R_U = y) g_U(y) dy \right] dF_S(z), \quad (12) \end{aligned}$$

Expanding the probability of event $A_U \leq x$ given a fixed residual $R_U = y$ leads to:

$$\begin{aligned} P(A_U \leq x | R_U = y) &= \frac{P(y < U \leq x + y)}{P(U > y)} \\ &= \frac{F_U(x + y) - F_U(y)}{1 - F_U(y)}. \quad (13) \end{aligned}$$

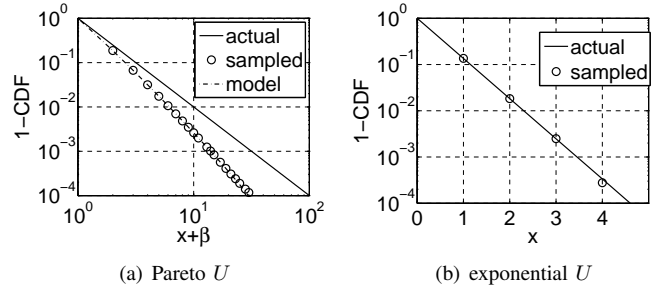


Fig. 5. Simulation results of M_1 under exponential S ($\lambda = 1, \mu = 2$).

Recalling that $g_U(y) = \mu(1 - F_U(y))$ is the residual density and applying (13), the inside integral of (12) becomes:

$$\begin{aligned} & \int_0^z P(A_U \leq x | R_U = y) g_U(y) dy \\ &= \mu \int_0^z (F_U(x + y) - F_U(y)) dy \\ &= \mu \int_0^z \bar{F}_U(y) dy - \mu \int_x^{z+x} \bar{F}_U(w) dw \\ &= G_U(z) + G_U(x) - G_U(x + z). \quad (14) \end{aligned}$$

This transforms (11) to:

$$\begin{aligned} G_1(x) &= \frac{\int_0^\infty (G_U(z) + G_U(x) - G_U(x + z)) dF_S(z)}{E[G_U(S)]} \\ &= \frac{E[G_U(S) - G_U(x + S)] + G_U(x)}{E[G_U(S)]}, \quad (15) \end{aligned}$$

which is the complement of the tail in (8). ■

Observe from (8) that M_1 generally measures neither the update distribution $F_U(x)$ nor the age distribution $G_U(x)$. To see the extent of this bias, Fig. 5(a) plots simulation results for exponential S and Pareto U in comparison to (8). Observe in the figure that our model closely tracks the simulated tail $\bar{G}_1(x)$, which remains heavy-tailed, albeit different from that of the target distribution $F_U(x)$. Fig. 5(b) shows that M_1 is indeed unbiased for Poisson N_U . We next investigate other conditions under which this approach may work well.

C. Quantifying Bias in M_1

Suppose $D_1 \sim G_1(x)$ is the random variable observed by M_1 over an infinitely long measurement period. Our goal in this subsection is to determine the relationship between D_1 , U , and A_U under different sampling strategies and update distributions. We first re-write (8) in a more convenient form.

Theorem 3: The tail distribution measured by M_1 can be expressed in two alternative forms:

$$\bar{G}_1(x) = \bar{G}_U(x) \frac{P(A_U < x + S | A_U > x)}{P(A_U < S)} \quad (16)$$

$$= \bar{F}_U(x) \frac{E[\int_0^S P(U > x + y | U > x) dy]}{E[\int_0^S P(U > y) dy]}. \quad (17)$$

Proof: We first show (16). Recalling that $G_U(x) = P(A_U < x)$ yields:

$$\bar{G}_1(x) = \frac{P(A_U < x + S) - P(A_U < x)}{P(A_U < S)}$$

$$= \bar{G}_U(x) \frac{P(x < A_U < x + S)}{P(A_U < S)P(A_U > x)}. \quad (18)$$

From the definition of conditional probability, we get:

$$\frac{P(x < A_U < x + S)}{P(A_U > x)} = P(A_U < x + S | A_U > x). \quad (19)$$

Substituting (19) into (18), we get (16).

To establish (17), rewrite (18) as:

$$\bar{G}_1(x) = \bar{F}_U(x) \frac{P(x < A_U < x + S)}{P(A_U < S)P(U > x)}, \quad (20)$$

whose numerator can be transformed to:

$$\begin{aligned} P(x < A_U < x + S) &= \mu E \left[\int_x^{x+S} \bar{F}_U(y) dy \right] \\ &= \mu E \left[\int_0^S \bar{F}_U(x+y) dy \right], \end{aligned} \quad (21)$$

where we use the fact that $g_U(x) = \mu \bar{F}_U(x)$. Dividing (21) by $\bar{F}_U(x)$ produces:

$$\frac{P(x < A_U < x + S)}{P(U > x)} = \mu E \left[\int_0^S P(U > x+y | U > x) dy \right].$$

Similarly, we can expand:

$$P(A_U < S) = \mu E \left[\int_0^S P(U > x) dy \right]. \quad (22)$$

Substituting the last two equations into (20), we obtain the desired result in (17). \blacksquare

Theorem 3 suggests that the tail of D_1 may indeed have some relationship to those of A_U and U . In order to establish this formally, we need to define three classes of variables.

Definition 3: Variable X is said to be NWU (new worse than used) if $P(X > x + y | X > y) > P(X > x)$ for all $x, y \geq 0$. If this inequality is reversed, X is said to be NBU (new better than used). Finally, if $P(X > x + y | X > y) = P(X > x)$ for all $x, y \geq 0$, the variable is called *memoryless*.

Note that NWU distributions are usually heavy-tailed, with two common representatives being Pareto and Weibull. Conditioning on U 's survival to some age y , its residual length $U - y$ is stochastically larger than U itself. NBU are typically light-tailed distributions, exemplified by uniform and constant. Finally, the memoryless class consists of only exponential distributions, where past knowledge has no effect on the future.

When both U and A_U are NWU, as is the case with Pareto distributions, Theorem 3 shows that $\bar{G}_1(x)$ is ‘‘sandwiched’’ between the other two tails, i.e., $\bar{F}_U(x)$ serves as a lower bound and $\bar{G}_U(x)$ as an upper. This means that D_1 is stochastically smaller than A_U , but stochastically larger than U . Fig. 6 shows an example confirming this, where the faster sampling rate in (b) moves the curve closer to $\bar{F}_U(x)$. The relationship among the tails is reversed if U and A_U are NBU. For exponential update distributions, all three tails are equal; however, this is the only obvious case where M_1 produces consistent estimation. We examine a few other cases next.

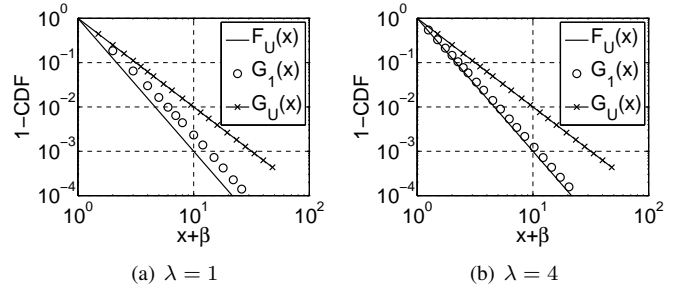


Fig. 6. Tail sandwich of M_1 under Pareto updates and constant S ($\mu = 2$).

D. Achieving Consistency in M_1

Now that we know that $\bar{G}_1(x)$ is contained between the tails of update and age distributions, there are two intuitive ways how bias can be removed. First, we could tighten the distance between tails $\bar{F}_U(x)$ and $\bar{G}_U(x)$; however, this can only be achieved by forcing the source to undergo updates with U that is ‘‘closer’’ to exponential. As this is usually impractical, the second technique is to adjust the sampling distribution $F_S(x)$ such that the distance of $\bar{G}_1(x)$ to one of U 's tails shrinks to zero. To this end, our next result demonstrates that D_1 ‘‘leans’’ towards U or A_U solely based on the fraction of retained samples p .

Theorem 4: For $p \rightarrow 1$, variable D_1 sampled by M_1 converges in distribution to A_U . For $p \rightarrow 0$ and mild conditions on S , variable D_1 converges in distribution to U .

Proof: Recall that $p = E[G_U(S)]$. When $E[G_U(S)] \rightarrow 1$, so does $E[G_U(S+x)]$. Therefore:

$$\bar{G}_1(x) = \frac{E[G_U(S+x)] - G_U(x)}{E[G_U(S)]} \rightarrow \bar{G}_U(x). \quad (23)$$

To prove the second part, assume that $S/E[S]$ converges to a random variable with mean 1. Since $p \rightarrow 0$ implies that $S \rightarrow 0$ almost surely, we get:

$$\frac{G_U(S)}{E[S]} = \frac{\int_0^S \bar{F}_U(y) dy}{E[U]E[S]} = \frac{S \int_0^1 \bar{F}_U(Sy) dy}{E[U]E[S]} \rightarrow \mu, \quad (24)$$

where we use the fact that $\bar{F}_U(Sy) \rightarrow 1$ for all fixed y .

Noticing that $G_U(S)/E[S]$ is upper bounded by random variable $\mu S/E[S]$, the latter of which has a finite mean, and applying the dominated convergence theorem (DCT), we get:

$$\lim_{p \rightarrow 0} \frac{E[G_U(S)]}{E[S]} = \mu. \quad (25)$$

Similarly, we obtain:

$$\begin{aligned} \frac{G_U(S+x) - G_U(x)}{E[S]} &= \frac{\int_x^{S+x} \bar{F}_U(y) dy}{E[U]E[S]} \\ &= \frac{S \int_0^1 \bar{F}_U(Sy+x) dy}{E[U]E[S]}, \end{aligned} \quad (26)$$

which converges to $\mu \bar{F}_U(x)$. Applying the DCT again, we get:

$$\lim_{p \rightarrow 0} \frac{E[G_U(S+x) - G_U(x)]}{E[S]} = \mu \bar{F}_U(x). \quad (27)$$

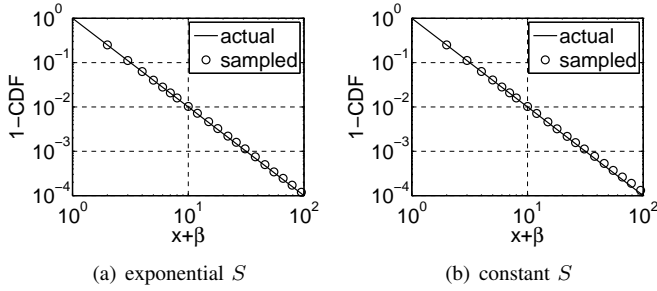


Fig. 7. Verification of (30) under Pareto updates and $\lambda = 1$.

Combining (25) and (27) produces:

$$\lim_{p \rightarrow 0} \frac{E[G_U(S+x) - G_U(x)]}{E[G_U(S)]} = \bar{F}_U(x), \quad (28)$$

which is what we intended to prove. ■

To understand this result, we discuss several examples. In order to converge p to 1, method M_1 has to sample with sufficiently large S to achieve $P(S > R_U) = 1$. For general $F_U(x)$, this can be guaranteed only if S converges to infinity, in which case the measurement process will be quite slow. If an upper bound on U is known, then setting S to be always larger can also produce $p = 1$. In these scenarios, however, M_1 will sample $G_U(x)$ and additional steps to recover $F_U(x)$ must be undertaken.

To achieve $p = 0$, M_1 has to use high sampling rates such that each update interval contains an infinite number of samples, i.e., S must converge to zero. In this case, the method may consume exorbitant network resources and additionally create undesirable load conditions at the source.

E. Method M_2

Instead of using the largest age sample for each detected update, a more sound option is to use *all* available ages. While extremely simple, this method has not been proposed before. We call this strategy M_2 and define $G_2(x, T)$ to be the fraction of its samples with values smaller than or equal to x in $[0, T]$:

$$G_2(x, T) := \frac{1}{N_S(T)} \sum_{j=1}^{N_S(T)} \mathbf{1}_{A_U(s_j) \leq x}. \quad (29)$$

The next result follows from Assumption 1 and the renewal equation [41].

Theorem 5: Method M_2 is consistent with respect to the age distribution:

$$G_2(x) := \lim_{T \rightarrow \infty} G_2(x, T) = G_U(x). \quad (30)$$

Next we use simulations to verify the usefulness of (30). From Fig. 7, observe that the sampled distribution of M_2 does in fact equal $G_U(x)$. To obtain $F_U(x) = 1 - g_U(x)/g_U(0)$ from an empirical CDF $G_U(x)$, we adopt numerical differentiation from [38]. This method uses bins of size h and k -point derivatives, bounding Taylor-expansion errors to $O(h^k/k!)$. For the estimator to work, it must first accurately determine $g_U(0) = 1/E[U]$. Using $k = 5$ and non-symmetric (i.e., one-sided) derivatives around $x = 0$, Fig. 8(a) demonstrates

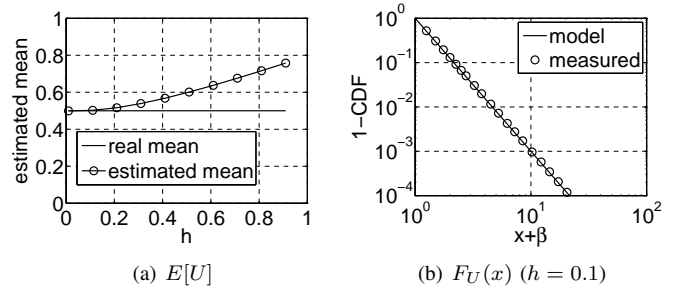


Fig. 8. Performance of M_2 under Pareto U and constant S ($\mu = 2, \lambda = 100$).

that the estimated $E[U]$ monotonically decreases in h and eventually stabilizes at the true value. Since h is a user-defined parameter independent of (N_U, N_S) , it can be arbitrarily small. Thus, a binary search on h to find the flat region in $E[U]$ can always determine its value with high accuracy. Applying this technique, the update distribution estimated by M_2 is shown in Fig. 8(b) in comparison to $F_U(x)$. Notice that the two curves are indistinguishable.

F. Discussion

Although M_1 has fewer samples, its network traffic remains the same as that of M_2 , because they both have to contact the source $N_S(t)$ times in $[0, t]$. However, the smaller number of *retained* values in M_1 may lead to lower computational cost and better RAM usage in density-estimation techniques that utilize all available samples (e.g., kernel estimators). For the route we have taken, i.e., differentiation of $G_2(x)$, the two methods exhibit the same overhead.

We now focus on the performance of M_2 in finite observation windows $[0, T]$. One potential issue is the redundancy (and high dependency) of samples that it collects (i.e., all ages within the same update interval are deterministically predictable), which is what M_1 tried to avoid. While necessary, can this redundancy lead to slower convergence? For a given T , would it be better to collect fewer samples that are spaced further apart?

Define

$$\zeta(T) := \frac{1}{N_S(T)} \sum_{j=1}^{N_S(T)} A_U(s_j) \quad (31)$$

to be the average age observed by M_2 in $[0, T]$ using one realization of the system. We now use deviation of $\zeta(T)$ from $E[A_U] = \mu E[U^2]/2$ as indication of error. Specifically, let

$$\epsilon(T) := E \left[\left| 1 - \frac{\zeta(T)}{E[A_U]} \right| \right]. \quad (32)$$

be the expected relative error computed over m sample-paths.

First, we fix the sampling rate $\lambda = 1$ and change T from 100 to 10M time units. As expected, $\epsilon(T)$ in Fig. 9(a) monotonically decreases as the observation window gets larger, confirming asymptotic convergence of M_2 discussed throughout this section. Next, we keep T constant at 10K and vary $E[S]$. As shown in Fig. 9(b), the error drops with $E[S]$, but then stabilizes. This means that having more samples, regardless of how redundant, improves performance only up to a certain point.

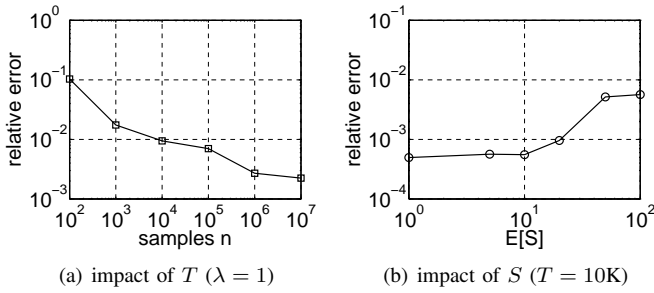


Fig. 9. Average relative error of $\zeta(T)$ of M_2 under Pareto U and exponential S ($\mu = 2, m = 1000$).

V. COMPARISON SAMPLING: CONSTANT INTERVALS

In contrast to the previous section, the remaining methods do not have access to age; instead, they must work with binary observations Q_{ij} , which indicate whether an update occurred between two sampling points s_i and s_j . This section deals with constant inter-download delay. This special case of comparison sampling is not just simple to implement and the only one considered in the literature, but also maximally polite (i.e., least bursty) for a given download rate λ .

A. Basics

Assume constant inter-sample delays $S = \Delta$ and notice that all observations related to update intervals must be multiples of Δ . It is therefore impossible to reconstruct $F_U(x)$, or even $G_U(x)$, in every point x . This requires an adjustment in objectives.

Definition 4: An estimator $\tilde{F}(x, T)$ is Δ -consistent with respect to distribution $F(x)$ if it can correctly reproduce it in all discrete points $x_n = n\Delta$ as $T \rightarrow \infty$:

$$\lim_{T \rightarrow \infty} \tilde{F}(x_n, T) = F(x_n), \quad n = 1, 2, \dots \quad (33)$$

As we discuss above and confirm below, none of the methods can measure $F_U(x)$ directly (unless the sampling rate is infinite or U is exponential). As a result, all algorithms generally face the issue of recovering $F_U(x)$ from $G_U(x)$. The main caveat of this section is that knowledge of the age distribution in discrete points is generally insufficient for Δ -consistent estimation of $F_U(x)$. This occurs because the estimated $G_U(x)$ lacks data in every interval (x_n, x_{n+1}) , which precludes differentiation and leaves $g_U(x)$ unobtainable.

Depending on the smoothness of $G_U(x)$ and/or prior knowledge about the target distribution, one can use interpolation between the known points $G_U(x_n)$. In such cases, $F_U(x)$ may be reconstructed with high accuracy using kernel density-estimation techniques; however, the result is application-specific. We thus do not dwell on numerical methods needed to perform these manipulations and instead focus on Δ -consistency in regard to $G_U(x)$.

B. Method M_3

Prior work in several fields [3], [7], [22], [32], [36] has suggested an estimator, which we call M_3 , that rounds the distance between each adjacent pair of detected updates to

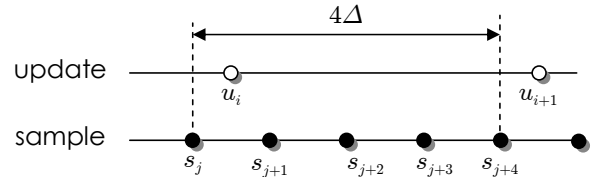


Fig. 10. Comparison sampling in M_3 with constant intervals of size Δ .

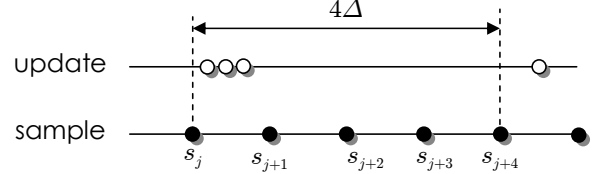


Fig. 11. Pitfalls of M_3 .

the nearest multiple of Δ , from which it builds a distribution $G_3(x)$. This technique was used in [7], [22] to track webpage updates, in [31] to estimate lifetimes of storage objects, and in [3], [32], [36] to sample user lifetimes in P2P networks. In the OS/networking literature, the approach is known as *Create-Based Method* (CBM) because it tracks each object from its creation, as opposed to other methods that track deletions.

Define r_k to be the number of downloads after which the k -th update is detected, i.e.,

$$r_k := \min \left\{ m \geq 1 : \sum_{j=1}^m Q_{j,j+1} = k \right\}. \quad (34)$$

Then, the samples collected by M_3 are $(r_{k+1} - r_k)\Delta$ for $k = 1, 2, \dots$. To understand this better, Fig. 10 shows an example where updates are detected after downloads j and $j+4$, which produces $r_{k+1} - r_k = 4$ and a single sample 4Δ . Based on the description in prior work, this technique serves the purpose of directly measuring U_i by counting full intervals of size Δ that fit in $[u_i, u_{i+1}]$. As a result, the output of M_3 is usually expected to produce the update distribution $F_U(x)$.

While this makes sense for the case in Fig. 10, the method becomes grossly inaccurate when multiple updates occur within Δ time units of each other, which brings us back to the issue of hidden variables U_i . Consider Fig. 11, where $2/3$ of the visible update durations are less than Δ . Since M_3 in this scenario produces one sample 4Δ , it skews the mass of the distribution to much higher values than needed.

We now model the performance of M_3 under general U and obtain the limiting distribution of its samples. Define $G_3(x, T)$ to be the CDF of observed durations in $[0, T]$:

$$G_3(x, T) := \frac{\sum_{k=1}^{\infty} \mathbf{1}_{r_k \leq T} \mathbf{1}_{(r_{k+1} - r_k)\Delta \leq x}}{\sum_{k=1}^{\infty} \mathbf{1}_{r_k \leq T}}. \quad (35)$$

Then, we have the following result.

Theorem 6: The tail distribution of M_3 is a step-function:

$$\bar{G}_3(x_n) := \lim_{T \rightarrow \infty} \bar{G}_3(x_n, T) = \frac{G_U(x_{n+1}) - G_U(x_n)}{G_U(\Delta)}. \quad (36)$$

Proof: Notice from Fig. 11 that age samples collected by M_3 can be viewed as discrete versions of those in M_1 . Indeed,

define $x^+ = \Delta \lceil x/\Delta \rceil$ to be x rounded-up to the nearest multiple of Δ . Then, the sample obtained by M_3 at download instance s_j is $A_U^+(s_j)$. Since condition $A_U^+(s_j) < x_n$ is equivalent to $A_U(s_j) < x_n$ for $x_n = n\Delta$, we obtain:

$$G_3(x_n, T) = \frac{\sum_{j=1}^{N_S(T)} \mathbf{1}_{R_U(s_j) < S_j} \mathbf{1}_{A_U(s_j) \leq x_n}}{\sum_{j=1}^{N_S(T)} \mathbf{1}_{R_U(s_j) < S_j}}, \quad (37)$$

which is exactly the same as $G_1(x_n, T)$ in (7). Therefore, the tail of $G_3(x_n, T)$ converges to the result in (8), with S replaced by Δ . Doing so produces (36). Since $G_3(x)$ has no information between discrete points x_n , it must be constant in each interval $[x_n, x_{n+1})$, which means it is a step-function. ■

Define a random variable $D_3 \sim G_3(x)$. With the result above, its average becomes readily available.

Theorem 7: The expectation of D_3 is given by:

$$E[D_3] = \frac{\Delta}{G_U(\Delta)}. \quad (38)$$

Proof: It is well-known that the mean of a non-negative lattice random variable can be obtained by summing up its tail distribution:

$$E[D_3] = \Delta \sum_{n=0}^{\infty} \bar{G}_3(x_n). \quad (39)$$

Expanding $\bar{G}_3(x_n)$ using (36) and canceling all but two remaining terms leads to the desired result. ■

Similar to M_1 , method M_3 is consistent when $F_U(x)$ is exponential. However, in broader NWU/NBU settings, its distribution lies between $F_U(x)$ and $G_U(x)$. As sampling interval $\Delta \rightarrow \infty$, which corresponds to $p \rightarrow 1$, variable D_3 converges in distribution towards A_U . When $\Delta \rightarrow 0$, which reflects $p \rightarrow 0$, D_3 tends to U . Unfortunately, neither scenario is usable in practice, which makes the method generally biased.

C. Method M_4

Using the rationale behind M_2 , we now propose another new method, which we call M_4 . At each sampling point s_j , the obtained value is:

$$D_4(s_j) := \begin{cases} \Delta & Q_{j-1,j} = 1 \\ D_4(s_{j-1}) + \Delta & \text{otherwise} \end{cases}. \quad (40)$$

For the example in Fig. 10, this method collects four samples – Δ , 2Δ , 3Δ and 4Δ . Denote by $G_4(x, T)$ the distribution generated by M_4 in $[0, T]$. Then, we have the following result.

Theorem 8: Method M_4 is Δ -consistent with respect to the age distribution:

$$G_4(x_n) := \lim_{T \rightarrow \infty} G_4(x_n, T) = G_U(x_n). \quad (41)$$

Proof: It is not difficult to see that M_4 collects samples $A_U^+(s_j)$ in all points s_j . Therefore,

$$G_3(x_n, T) = \frac{\sum_{j=1}^{N_S(T)} \mathbf{1}_{A_U^+(s_j) \leq x_n}}{N_S(T)} \quad (42)$$

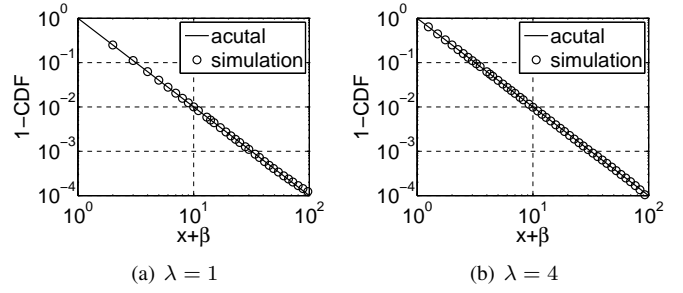


Fig. 12. Verification of (45) under Pareto U ($\mu = 2$).

where $x^+ = \Delta \lceil x/\Delta \rceil$ as before. Since the CDF is computed only in discrete points x_n , the above can be written as:

$$G_3(x_n, T) = \frac{\sum_{j=1}^{N_S(T)} \mathbf{1}_{A_U(s_j) \leq x_n}}{N_S(T)} = G_2(x_n, T), \quad (43)$$

which converges to $G_U(x)$ using (30). ■

Define a random variable $D_4 \sim G_4(x)$, where $G_4(x)$ is a continuous step-function taking jumps at each x_n . Interestingly, even though M_3 keeps the largest age sample in each detected update interval $[u_i, u_{i+1}]$, the mean of its values $E[D_3]$ is not necessarily larger than that of D_4 . For example, with Pareto updates and $\Delta = 1$, we get $E[D_4] = 1.63$ and $E[D_3] = 1.33$. This can be explained by our previous discussion showing that under NWU update intervals the tail $\bar{G}_3(x)$ is upper-bounded by $\bar{G}_4(x)$, which implies $E[D_4] \geq E[D_3]$. Note that if U is NBU, this relationship is again reversed.

D. Method M_5

From the last two subsections, we learned that M_4 is always Δ -consistent with respect to $G_U(x)$, while M_3 is biased unless U is exponential or Δ is infinitely small. One advantage that M_3 may have is that it operates with significantly fewer samples. This raises the question of whether one can achieve Δ -consistency using the same number of samples as M_3 .

To this end, and define:

$$G_5(x_n, T) := \frac{1}{T} \sum_{j=1}^{T/\Delta} \min(x_n, A_U^+(s_j)) Q_{j,j+1} \quad (44)$$

to be an estimator that takes samples of M_3 , passes them through the min function, and normalizes the resulting sum by window size T . Note that the number of terms in the summation is $K(\infty, T)$, i.e., the number of detected updates.

Theorem 9: Estimator M_5 is Δ -consistent with respect to the age distribution:

$$G_5(x_n) := \lim_{T \rightarrow \infty} G_5(x_n, T) = G_U(x_n). \quad (45)$$

Proof: We start with an auxiliary result:

$$\begin{aligned} \sum_{k=0}^{n-1} \mathbf{1}_{A_U(s_j) > x_k} &= \sum_{k=0}^{n-1} \mathbf{1}_{A_U^+(s_j) > x_k} \\ &= \sum_{k=0}^{n-1} \mathbf{1}_{\lceil A_U(s_j) \Delta \rceil > k} \end{aligned}$$

TABLE I
CONVERGENCE OF BOTH Δ -CONSISTENT
METHODS UNDER PARETO U ($\mu = 2, \lambda = 1$)

T	M_4		M_5	
	$w(T)$	$\kappa(T)$	$w(T)$	$\kappa(T)$
10^2	3.5×10^{-2}	6.4×10^{-2}	3.7×10^{-2}	6.7×10^{-2}
10^3	1.4×10^{-2}	2.2×10^{-2}	1.4×10^{-2}	2.2×10^{-2}
10^4	4.7×10^{-3}	7.2×10^{-3}	4.7×10^{-3}	7.3×10^{-3}
10^5	1.5×10^{-3}	2.4×10^{-3}	1.5×10^{-3}	2.4×10^{-3}
10^6	4.1×10^{-4}	5.8×10^{-4}	4.1×10^{-4}	5.8×10^{-4}
10^7	2.2×10^{-4}	2.6×10^{-4}	2.2×10^{-4}	2.6×10^{-4}

$$\begin{aligned}
&= \min(n, \lceil A_U(s_j) \Delta \rceil) \\
&= \frac{\min(x_n, A_U^+(s_j))}{\Delta}.
\end{aligned} \quad (46)$$

Next, applying this to expansion of (44):

$$\begin{aligned}
G_5(x_n, T) &= \frac{\Delta}{T} \sum_{j=1}^{T/\Delta} Q_{j,j+1} \sum_{k=0}^{n-1} \mathbf{1}_{A_U(s_j) > x_n} \\
&= \frac{\Delta}{T} \sum_{k=0}^{n-1} \sum_{j=1}^{T/\Delta} \mathbf{1}_{A_U(s_j) > x_n} Q_{j,j+1} \\
&= \frac{K(\infty, T)}{N_S(T)} \sum_{k=0}^{n-1} \tilde{G}_3(x_n, T),
\end{aligned} \quad (47)$$

where $K(x, T)$ is given by (6) and $\tilde{G}_3(x_n, T)$ by (37). Since $K(\infty, T)/N_S(T)$ converges to p , we get after applying (36) to the expansion of $\tilde{G}_3(x_n, T)$:

$$G_5(x_n) = p \frac{G_U(x_n)}{G_U(\Delta)} = G_U(x_n),$$

where we use the fact that $p = G_U(\Delta)$. ■

Fig. 12 shows that M_5 accurately obtains the tail of $G_U(x)$, even for Δ bounded away from zero. We next compare M_5 with M_4 to see if the reduction in the number of samples has a noticeable impact on accuracy. The first metric under consideration is the Weighted Mean Relative Difference (WMRD), often used in networking [12]. Assuming $H(x, T)$ is some empirical CDF computed in $[0, T]$, then the WMRD between $H(x, T)$ and $G_U(x)$ is:

$$w(T) := \frac{\sum_n |H(x_n, T) - G_U(x_n)|}{\sum_n (H(x_n, T) + G_U(x_n))/2}. \quad (48)$$

The second metric is the Kolmogorov-Smirnov (KS) statistic, which is the maximum distance between two distributions:

$$\kappa(T) := \sup_x |H(x, T) - G_U(x)|. \quad (49)$$

Simulations results are shown in Table I. Observe that M_4 performs slightly better for $T \leq 10^3$, but then the two methods become identical and their error decays as $1/\sqrt{T}$. Even if T is small, the minor loss of accuracy in M_5 may well be worth a 20% reduction in the number of samples. As given in Fig. 4(a), larger λ leads to even higher savings, e.g., 80% for $\lambda = 10$.

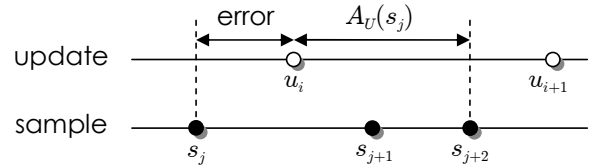


Fig. 13. Illustration of G- M_4 .

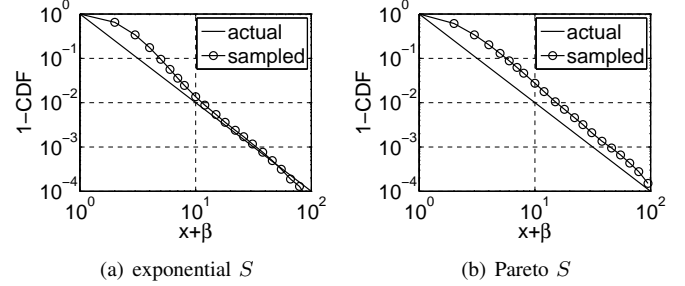


Fig. 14. Bias of G- M_4 with Pareto updates ($\mu = 2, \lambda = 1$).

VI. COMPARISON SAMPLING: RANDOM INTERVALS

Although M_4 and M_5 are consistent estimators of $G_U(x)$, they do not generally guarantee recovery of $F_U(x)$. Furthermore, constant S may not always be achievable in practice. For instance, search engine juggle trillions of pages, whose download rate is dynamically adjusted based on real-time ranking and budgeting. It may thus be difficult to ensure constant return delays to each page. Additional problems stem from lattice update processes, where constant S fails to satisfy Assumption 1, rendering measurements arbitrarily inaccurate.

In this section, we consider comparison sampling with random intervals. We first show that extending M_4 to this scenario delivers surprisingly biased results. Then, we present our new method M_6 and verify its correctness using simulations.

A. Straightforward Approaches

Our first attempt is to generalize M_4 to random S , which we call G- M_4 . For a given s_j , define the most-recent sample point after which an update has been detected as:

$$s_j^* := \max_{i < j} \{s_i : Q_{ij} = 1\}. \quad (50)$$

Then, G- M_4 rounds age $A_U(s_j)$ up to $s_j - s_j^*$. An example is shown in Fig. 13, where the measured value is $s_{j+2} - s_j$. For constant S , this method is identical to M_4 , which we know is consistent. The main difference with random S is that the amount of round-off error in G- M_4 varies from interval to interval. This issues has a profound impact on the result, as shown in Fig. 14. Observe that the exponential case becomes somewhat consistent only for $x_n \gg 0$ and the Pareto case produces a tail that is completely different from the actual $G_U(x)$. This motivates us to search for another approach.

B. Method M_6

Our rationale for this technique stems from the fact that $Q_{ij} = 1$ if and only if $A_U(s_j) < s_j - s_i$. Therefore, counting the fraction of pairs (i, j) that sustain an update may lead to

$G_U(x)$. Define $y^\circ = h\lceil y/h \rceil$ to be the rounded-up value of y with respect to a user-defined constant h . Let $y_n = nh$ and:

$$W_{ij}(y_n) := \begin{cases} 1 & (s_j - s_i)^\circ = y_n \\ 0 & \text{otherwise} \end{cases}. \quad (51)$$

Then, the number of inter-sample distances $s_j - s_i$ in $[0, T]$ that round up to y_n is given by:

$$W(y_n, T) := \sum_{i=1}^{N_S(T)} \sum_{j=i+1}^{N_S(T)} W_{ij}(y_n) \quad (52)$$

and the number of them with an update is:

$$Z(y_n, T) := \sum_{i=1}^{N_S(T)} \sum_{j=i+1}^{N_S(T)} Q_{ij} W_{ij}(y_n). \quad (53)$$

We can now define estimator M_6 by its CDF:

$$G_6(y_n, T) := \frac{Z(y_n, T)}{W(y_n, T)} \quad (54)$$

For a given λ , method M_6 has the same network overhead as the other methods; however, it utilizes $\Theta(n^2)$ pairwise comparisons, significantly more than the other methods, which are all linear in n . Despite a higher computational cost, M_6 gains significant accuracy advantages when distances $s_i - s_j$ are allowed to sweep all possible points $x \geq 0$. Combining this with bins of sufficiently small size creates a continuous CDF, which allows recovery of not only $G_U(x)$, but also $F_U(x)$.

Theorem 10: Assume $h \rightarrow 0$ and $F_S(x) > 0$ for all $x > 0$. Then, method M_6 is consistent with respect to the age distribution:

$$G_6(y) := \lim_{T \rightarrow \infty} G_6(y, T) = G_U(y). \quad (55)$$

Proof: First, it helps to observe that:

$$Q_{ij} = \mathbf{1}_{R_U(s_i) \leq (s_j - s_i)}. \quad (56)$$

Since the download process is renewal, it follows that:

$$s_j - s_i \sim F_S^{*(j-i)}, \quad (57)$$

where $F_S^{*k}(x)$ denotes a k -fold convolution of distribution $F_S(x)$. Furthermore, the renewal nature of N_S implies that variable $s_j - s_i$ is independent of s_i . Now, let

$$Y_k \sim F_S^{*k}(x) \quad (58)$$

be a random variable with the same distribution as $S_1 + \dots + S_k$ and define the renewal function driven by $F_S(x)$ as [41]:

$$M_S(t) = 1 + \sum_{k=1}^{\infty} F_S^{*k}(t). \quad (59)$$

Then, renewal theory shows for $x > h$ and $n \rightarrow \infty$ that:

$$\frac{1}{n} \sum_{i=1}^n \sum_{j=i+1}^n \mathbf{1}_{R_U(s_i) \leq (s_j - s_i)} \mathbf{1}_{s_j - s_i \in (x-h, x]} \quad (60)$$

converges to

$$\sum_{k=1}^{\infty} P(R_U \leq Y_k, Y_k \in (x-h, x])$$

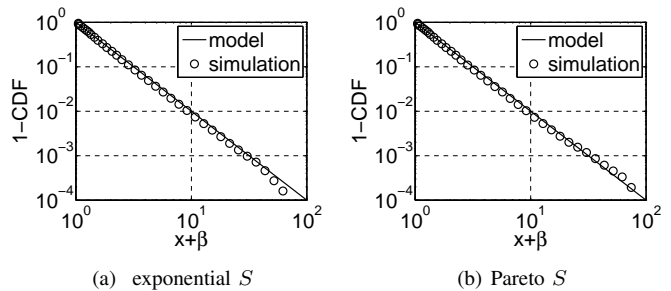


Fig. 15. Simulations of M_6 under Pareto updates ($h = 0.05, \mu = 2, \lambda = 1$).

$$= \sum_{k=1}^{\infty} \int_{x-h}^x G_U(y) dF_S^{*k}(y) = \int_{x-h}^x G_U(y) dM_S(y). \quad (61)$$

Let $n = N_S(T)$ and assume that $h(T) = T^{-\delta}$, where $\delta \in (0, 1)$ ensures that h diminishes to zero at some appropriate rate¹. Since $G_U(x)$ is continuous, it follows that:

$$\lim_{T \rightarrow \infty} \frac{Z(y_n, T)}{W(y_n, T)} = \lim_{h \rightarrow 0} \frac{\int_{x-h}^x G_U(y) dM_S(y)}{\int_{x-h}^x dM_S(y)} = G_U(x) \quad (62)$$

for each $x > 0$. ■

The assumption that $F_S(x)$ contains non-zero mass in the vicinity of zero is necessary for accurate estimation of $g_U(x)$ at $x = 0$, which then leads to $F_U(x)$. This can be accomplished by a number of continuous distributions, e.g., $\exp(\lambda)$ or uniform in $[0, 2\lambda]$. It should also be noted that M_6 works for lattice S , but in that case it offers no benefits over M_4 . Fig. 15 compares the M_6 estimator against $G_U(x)$ under two sampling distributions $F_S(x)$, both satisfying Theorem 10. Compared to Fig. 14, this result is overwhelmingly better.

VII. CONCLUSION

This paper studied the problem of estimating the update distribution at a remote source under blind sampling. We analyzed prior approaches in this area, showed them to be biased under general conditions, introduced novel modeling techniques for handling these types of problems, and proposed several unbiased algorithms that tackled network sampling under a variety of assumptions on the information provided by the server and conditions at the observer.

Future work includes analysis of convergence speed, investigation of non-parametric smoothing techniques for density estimation, and modeling of non-stationary update processes.

REFERENCES

- [1] E. Adar, J. Teevan, S. T. Dumais, and J. L. Elsas, "The Web Changes Everything: Understanding the Dynamics of Web Content," in *Proc. WSDM*, Jun. 2009, pp. 282–291.
- [2] B. E. Brewington and G. Cybenko, "How Dynamic is the Web," *Computer Networks*, no. 1-6, pp. 257–276, Jun. 2000.
- [3] F. E. Bustamante and Y. Qiao, "Friendships that Last: Peer Lifespan and its Role in P2P Protocols," in *Proc. Web Content Caching and Distribution*, Sep. 2003.
- [4] J. Cho and H. Garcia-Molina, "The Evolution of the Web and Implications for an Incremental Crawler," in *Proc. VLDB*, Sep. 2000, pp. 200–209.

¹From well-known results in non-parametric function estimation, $\delta = 1/5$ should be optimal for the mean squared error of the estimate.

- [5] J. Cho and H. Garcia-Molina, "Synchronizing a Database to Improve Freshness," in *Proc. ACM SIGMOD*, May 2000, pp. 117–128.
- [6] J. Cho and H. Garcia-Molina, "Effective Page Refresh Policies for Web Crawlers," *ACM Transactions on Database Systems*, no. 4, pp. 1–36, Dec. 2003.
- [7] J. Cho and H. Garcia-Molina, "Estimating Frequency of Change," *ACM Trans. Internet Technol.*, vol. 3, no. 3, pp. 256–290, Aug. 2003.
- [8] G. L. Ciampaglia and A. Vancheri, "Empirical Analysis of User Participation in Online Communities: the Case of Wikipedia," in *Proc. ICWSM*, Sep. 2010, pp. 219–222.
- [9] E. Cohen and H. Kaplan, "The Age Penalty and Its Effect on Cache Performance," in *Proc. USENIX USITS*, Mar. 2001, pp. 73–84.
- [10] E. Cohen and H. Kaplan, "Refreshment Policies for Web Content Caches," in *Proc. IEEE INFOCOM*, Apr. 2001, pp. 1398–1406.
- [11] D. Denev, A. Mazeika, M. Spaniol, and G. Weikum, "SHARC: Framework for Quality-Conscious Web Archiving," in *Proc. VLDB*, Aug. 2009, pp. 183–207.
- [12] N. Duffield, C. Lund, and M. Thorup, "Estimating Flow Distributions from Sampled Flow Statistics," in *Proc. ACM SIGCOMM*, Aug. 2003, pp. 325–336.
- [13] J. Edwards, K. McCurley, and J. Tomlin, "An Adaptive Model for Optimizing Performance of an Incremental Web Crawler," in *Proc. WWW*, May 2001, pp. 106–113.
- [14] D. Fetterly, M. Manasse, and M. Najork, "On the Evolution of Clusters of Near-Duplicate Web Pages," in *Proc. LA-WEB*, Nov. 2003, pp. 37–45.
- [15] C. Grimes and D. Ford, "Estimation of Web Page Change Rates," in *Proc. JSM*, 2008.
- [16] C. Grimes, D. Ford, and E. Tassone, "Keeping a Search Engine Index Fresh: Risk and Optimality in Estimating Refresh Rates for Web Pages," in *Proc. INTERFACE*, May 2008.
- [17] D. Gruhl, L. Chavet, D. Gibson, J. Meyer, P. Pattanayak, A. Tomkins, and J. Zien, "How to Build a WebFountain: An Architecture for Very Large-Scale Text Analytics," *IBM Systems Journal*, vol. 43, no. 1, pp. 64–77, 2004.
- [18] R. Horincar, B. Amann, and T. Artières, "Online Change Estimation Models for Dynamic Web Resources," in *Proc. ICWE*, Jul. 2012, pp. 395–410.
- [19] J.-J. Lee, K.-Y. Whang, B. S. Lee, and J.-W. Chang, "An Update-Risk Based Approach to TTL Estimation in Web Caching," in *Proc. IEEE WISE*, Dec. 2002, pp. 21–29.
- [20] X. Li, D. B. Cline, and D. Loguinov, "On Sample-Path Staleness in Lazy Data Replication," in *Proc. INFOCOM*, Apr. 2015.
- [21] X. Li and D. Loguinov, "Stochastic Models of Pull-Based Data Replication in P2P Systems," in *Proc. IEEE P2P*, Sep. 2014, pp. 1–10.
- [22] N. Matloff, "Estimation of Internet File-access/Modification Rates from Indirect Data," *ACM Trans. Model. Comput. Simul.*, vol. 15, pp. 233–253, Jul. 2005.
- [23] A. Ntoulas, J. Cho, and C. Olston, "What's New on the Web? The Evolution of the Web from a Search Engine Perspective," in *Proc. WWW*, May 2004, pp. 1–12.
- [24] M. Oita and P. Senellart, "Deriving Dynamics of Web Pages: A Survey," in *Proc. TAWW*, Mar. 2011, pp. 25–32.
- [25] C. Olston and S. Pandey, "Recrawl Scheduling Based on Information Longevity," in *Proc. WWW*, Apr. 2008, pp. 437–446.
- [26] V. N. Padmanabhan and L. Qiu, "The Content and Access Dynamics of a Busy Web Site: Findings and Implications," in *Proc. ACM SIGCOMM*, Aug. 2000, pp. 111–123.
- [27] S. Pandey, K. Dhamdhere, and C. Olston, "WIC: A General-Purpose Algorithm for Monitoring Web Information Sources," in *Proc. VLDB*, Aug. 2004, pp. 360–371.
- [28] S. Pandey and C. Olston, "User-Centric Web Crawling," in *Proc. WWW*, May 2005, pp. 1–12.
- [29] S. Pandey, K. Ramamritham, and S. Chakrabarti, "Monitoring the Dynamic Web to Respond to Continuous Queries," in *Proc. WWW*, May 2003, pp. 659–668.
- [30] K. Radinsky and P. N. Bennett, "Predicting Content Change on the Web," in *Proc. WSDM*, Feb. 2013, pp. 415–424.
- [31] D. Roselli, J. R. Lorch, and T. E. Anderson, "A Comparison of File System Workloads," in *Proc. USENIX Annual Technical Conference*, Jun. 2000, pp. 41–54.
- [32] S. Saroiu, P. K. Gummadi, and S. D. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems," in *Proc. SPIE/ACM Multimedia Computing and Networking*, vol. 4673, Jan. 2002, pp. 156–170.
- [33] K. C. Sia and J. Cho, "Efficient Monitoring Algorithm for Fast News Alerts," *IEEE Trans. Knowl. Data Eng.*, no. 7, pp. 950–961, Jul. 2007.
- [34] S. R. Singh, "Estimation of Web Page Change Rates," in *Proc. International Joint Conferences on Artificial Intelligence*, Jun. 2007, pp. 2874–2879.
- [35] M. Spaniol, D. Denev, A. Mazeika, G. Weikum, and P. Senellart, "Data Quality in Web Archiving," in *Proc. WICOW*, Apr. 2009.
- [36] D. Stutzbach and R. Rejaie, "Understanding Churn in Peer-to-Peer Networks," in *Proc. ACM IMC*, Oct. 2006, pp. 189–202.
- [37] Q. Tan and P. Mitra, "Clustering-based Incremental Web Crawling," *ACM Transactions on Information Systems*, no. 4, pp. 1–27, Nov. 2010.
- [38] C. Voglis, P. Hadjidakis, I. Lagaris, and D. Papageorgiou, "A Numerical Differentiation Library Exploiting Parallel Architectures," *Computer Physics Communications*, vol. 180, no. 8, pp. 1404–1415, Aug. 2009.
- [39] X. Wang, Z. Yao, and D. Loguinov, "Residual-Based Measurement of Peer and Link Lifetimes in Gnutella Networks," in *Proc. IEEE INFOCOM*, May 2007, pp. 391–399.
- [40] J. L. Wolf, M. S. Squillante, P. S. Yu, J. Sethuraman, and L. Ozsen, "Optimal Crawling Strategies for Web Search Engines," in *Proc. WWW*, May 2002, pp. 136–147.
- [41] R. W. Wolff, *Stochastic Modeling and the Theory of Queues*. Prentice Hall, 1989.
- [42] M. Yang, H. Wang, L. Lim, and M. Wang, "Optimizing Content Freshness of Relations Extracted From the Web Using Keyword Search," in *Proc. ACM SIGMOD*, Jun. 2010, pp. 819–830.