Optimizing Update Frequencies for Decaying Information

Simon Razniewski
Free University of Bozen-Bolzano
razniewski@inf.unibz.it

ABSTRACT

Many kinds of information, e.g., addresses, crawls of webpages, or academic affiliations, are prone to becoming outdated over time. Therefore, if data quality shall be maintained over time, often periodical refreshing is done. As refreshing data usually has a cost, for instance computation time, network bandwidth or human work time, a problem is to find the right update frequency depending on the benefit gained from the information and on the speed with which the information is expected to get outdated.

This is especially important since often entities exhibit a different speed of getting outdated, e.g., addresses of students change more frequently than addresses of retirees, or news portals change more frequently than homepages. Consequently, there is no uniform best update frequency for all entities.

Previous work on data freshness has investigated how to best *distribute* a fixed number of updates among entities, in order to maximize average freshness. For businesses that are able to adapt their resources, another question is to determine the number of updates that *optimizes* the income derived from the data.

In this paper we present a model for describing the relationship between update frequency and income derived from data, present solutions for calculating the optimal update frequency for two common classes of functions for describing decay behaviour, and validate the benefits of our framework.

1. INTRODUCTION

In many applications such as address management or website crawling, information gets outdated over time and periodical refreshing is needed in order to ensure that the information remains useful. Refreshing information usually has a cost, e.g., computation time, network bandwidth or human work time. For instance, a company doing web indexing wants to revisit websites neither too seldom, as this leads to the stored information being outdated and thus to unsatisfied customers, nor too often, as this leads to too high computation and networking costs. The same holds for an advertising company that maintains a database of addresses: If the addresses are updated too seldom, much advertisement will not reach its destination. But if too much effort is spent on keeping the addresses up-to-date, this effort may outweigh the ad revenue. Previous work has studied how to *distribute* a fixed number of update resources best such that the (possibly weighted) freshness of the information is maximized ("Should I better use an update for entity A or entity B?"). While in the short-term, organizations may have to get along with the resources available, we believe that it is also crucial for them to know how many resources they should ideally provide, so that they can adjust their resources in the medium term. This is especially relevant today given the scalable availability of cloud resources and crowdsourcing.

In this paper we study the problem of *update frequency optimization,* which asks for the update frequency that maximizes the net income, that is, the benefit from correct information minus the cost for the updates ("How often should I entity website A?"). We next discuss two concrete scenarios where update distribution and optimization play a role.

Medical Advertisement. Consider an address reselling company in the medical domain, Spam Inc., whose business model consists of maintaining a high-quality database of specialist doctors and selling the contact information to suppliers of medical technology and to pharmaceutical companies. As medical equipment can be expensive, suppliers are willing to pay considerable sums for addresses of specialists that they intend to target with advertisement. The main activity of Spam Inc. is therefore the acquisition of information about new specialists, and the maintenance of existing information. For both tasks, it uses two techniques, web search and phone calls. Information about doctors shows different decay rates. Younger doctors are more likely to move than older doctors, and similar differences exist also between doctors in big cities versus in the countryside. If Spam Inc. has a fixed set of employees, its problem is one of resource distribution. Given the amount of updates that the employees can perform, the company has to decide how many of these should be used for each doctor. In the medium-term however, Spam Inc. may be interested in adjusting its resources in order to optimize its income, thus facing the problem of determining the optimal update frequency.

Web Crawling. In web crawling, companies are providing service based on information extracted from the web. Example services are search, price comparison sites or news aggregators. A commonality is that the quality of the provided service is highly dependent on the crawling frequency. A recent (2015) estimate for the cost of a crawling a single web document is 0.003 Cents, based on the pricing of $299 per 10 Million crawls of the crawl provider 80legs[1][2]. Webpages may exhibit very different frequencies with

CIKM'16 , October 24 - 28, 2016, Indianapolis, IN, USA

ISBN 978-1-4503-4073-1/16/10. . . $15.00

DOI: http://dx.doi.org/10.1145/2983323.2983719

[1]http://80legs.com

[2]A DIY solution confirms this, arriving at a slightly lower cost of 0.0023 ct/crawl at http://www.michaelnielsen.org/ddi/how-to-crawl-a-quarter-billion-webpages-in-40-hours/

which they get outdated. While e.g. newspages can change minutely, personal homepages often change even less frequently than once per month. From past crawls, website attributes and website content, crawlers can estimate the likelihood of change of a given webpage [8, 10], and given such estimated change frequencies, the crawlers have to decide how often to recrawl a given webpage. If their resources are fixed (i.e., they have a fixed number of servers and no or flat network fees), they face the problem of update distribution. If their resources are not fixed however, e.g. if they are able to scale their computing power using cloud services, are planning to buy servers for future use, or are charged depending on their network usage, they face the problem of determining the optimal update frequency for each website.

Previous work has investigated the resource distribution problem [7, 9, 12, 23, 29] in depth, focusing on distributions that are optimal wrt. metrics such as average freshness, staleness, or embarrassement. The straightforward question "How often should I update an entity", however, has not been adressed at all so far. A possible reason might be that previous work uses incomparable metrics for cost and benefit.

Contribution. Our contribution is to present a *cost-benefit-framework for determining the optimal update frequencies* for decaying entities. In particular, we

1. Introduce a framework to describe the net income derived from an update frequency of an entity subject to decay,
2. Show how one can compute the optimal update frequency under linear and exponential decay, which are two common classes of functions used to describe decay behaviour,
3. Extend our model to bulk updates, different costs between currency checks and updates, and costs for outdated entities,
4. Illustrate the benefit of our approach on two use cases, address data maintenance [26] and web crawling [7].

Our solution is independent of the actual decay function and applicable to a range of scenarios such as address data maintenance, web crawling or view maintenance.

Outline. In Section 2, we present a detailed motivating scenario. Section 3 discusses related work, Section 4 introduces the mathematical framework for information decay, and Sections 5 introduces the core framework for determining optimal update frequencies. In Section 6 we show how to deal with bulk updates, and further refine the model in Section 7. In Section 8 we validate the benefits of our framework, and conclude with a discussion in Section 9.

2. MOTIVATING SCENARIO

To illustrate the problems of update distribution and optimization, consider again the medical advertisement company, Spam Inc., and suppose that it currently has addresses of two doctors, Dr. Smith and Dr. Jones. Suppose that both addresses have a linear probability of getting outdated within 0 and 10, and within 0 and 20 years, respectively, as shown in Fig. 1 at the top. This implies for instance that after 5 years, the chance for Dr. Smith's address to still be correct is 50% and for Dr. Jones's 75%, and after 8 years, 20% and 60%, and so on).

Regarding the problem of distribution, suppose that Spam Inc. has resources to perform 3 updates per 10 years. Then the best distribution of these updates is to update Dr. Smith twice (every 5 years), and Dr. Jones once (every 10 years), because this implies that each of the addresses has an average probability of 0.75 to be up-to date, which also gives the best average for both (0.75). The probability of correctness as function of time for each address is shown in Fig. 1 (bottom). In contrast, if Spam Inc. would invert the frequencies and update Dr. Smith only once but Dr. Jones twice, this would give an average correctness of 0.5 and 0.875, respectively, and thus a lower average of 0.6875.

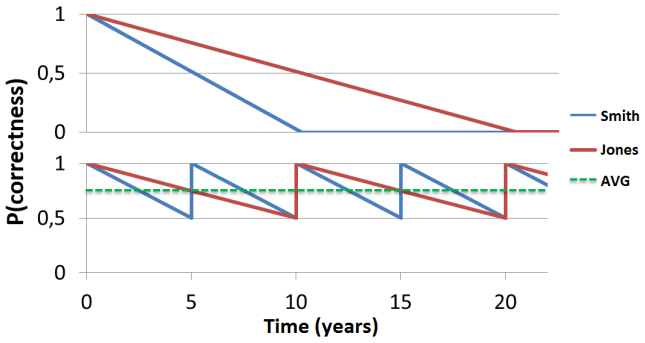

Figure 1: Correctness probability as a function of time, upper graph without updates, lower graph with updates for A every 5 years, for B every 10 years.

On the other hand, suppose that each update has a cost of \$1, and that the yearly gain from a correct address is \$1, too. Now we can ask for the optimal update resource allocation for Dr. Smith and Dr. Jones, that is, the number of updates that maximizes the gain minus the update costs. It turns out that for Dr. Smith, the optimal update interval is 4.5 years (2.2 updates per 10 years), which yields an average correctness of 0.78, an update cost of \$2.2, a gain of \$7.8 and a net income of $\$7.8 - \$2.2 = \$5.6$ per 10 years. Making available more resources, e.g. 3 updates per 10 years, leads to a higher average correctness of 0.83, and thus also to a higher gain of \$8.3, however, the higher update cost of \$3 outweighs this, leading to an overall income of $\$8.3 - \$3 = \$5.3$. Similarly, fewer resources, e.g., 2 updates per 10 years, are not optimal, as this leads to a net income of only \$5.5 per 10 years.

Using the same analysis, we find that the optimal update interval for Dr. Jones is every 6.3 years (1.6 updates per 10 years), which leads to an average correctness of 0.83 and a net income of \$6.8, compared with a net income of \$6.5 when updating once every 10 years. Summing the optimal frequencies for Dr. Smith and Dr. Jones, we conclude that in the medium term, in order to maximize its income, Spam Inc. should adjust its resources to be able to do $1.6 + 2.2 = 3.8$ updates instead of 3 per 10 years, as this improves its net income by 3.3% from \$12 to \$12.4.

3. RELATED WORK

Related work can be grouped into four topics: *Data currency* measures the quality of data, while *decay* describes how it gets outdated. In *web crawling*, update distribution has been extensively studied, while related problems on *determining optimal resources* have appeared in data integration.

Data Currency. Data currency, freshness or timeliness is, together with correctness and completeness, one of the core dimensions of data quality. Varying definitions state that timeliness measures the fraction of data that is up-to-date, the fraction of the current state of the world that is covered, or the time passed since data was acquired or was last known to be up-to-date [28, 27, 2].

Research on data currency has investigated how to measure and quantify freshness [4, 16], how to infer most recent values based on implicit orderings [13], or how to balance freshness and performance in view maintenance [6, 3, 19] (see below).

Decay. Decay is a well-known concept in physics and chemistry, for instance as radioactive decay or in chemical reactions, where it describes the quantitative degradation of substances over time. In many domains, also information is subject to change over time. An overview of relevant classes of decay functions for information is contained in [16], where the authors discuss linear, exponential, geometric, Weibull, and Gamma distributions for describing decay rates, and describe a methodology to measure data currency.

Information decay is also known in other domains such as recommender systems [18], where it is called drift in user interest, and describes the observation that the longer ago a user was found to have an interest in a certain record, the higher the chance that interest may have changed.

Also in entity matching and data cleaning [20, 14] it is known that decay plays a role, as correspondences between frequently changed attribute values are better indicators for the equivalence between entities than older correspondences.

Finally, decay is also a frequent issue for storage media. There decay is often called data degradation or data rot and refers to the physical processes that make data unreadable [25, 1][3]. Since different storage media show different decay rates, our analysis may also be relevant for refresh policies for degrading storage media.

Update Distribution in Crawling. One of the earliest works on optimal resource distribution in web crawling was done by Coffman et al. [9]. Cho and Garcia-Molina [7] showed that an optimal distribution of updates among a set of entities decaying with varying decay rates can significantly outperform random, uniform and proportional update distribution, and described a methodology to compute the optimal update distribution. Similarly, Edwards et al. [12] studied the ordering of webpages in the queue of a crawler in order to optimize metrics related to freshness.

Several subsequent works have focussed on resource distribution, such as [21], which discusses optimal resource distribution when webpages have different popularities, [17], which is centered on RSS feeds, [23], which investigates minimizing the divergence between crawled version and the true state of gradually changing websites, or [29], which centers on minimizing average staleness and the embarassement from dead links in search results.

Various other topics regarding web crawling have received attention: Cho and Garcia-Molina also studied how to determine the change frequency of web pages [8]. Denev et al. [11] discuss stochastic techniques for getting sharp crawls of large webpages, where the problem is that avoid inconsistencies and dead links due to differences in the exact crawl times of individual pages.

Similar problems with decaying information occur also in data delivery on the web, where based on freshness, a choice has to be made which views to materialize [19], or where user preferences between performance and freshness may guide a server in deciding whether to deliver possibly stale but cached data, or whether to deliver more costly fresh data [5]. Also in stream data warehouses, updates have to be scheduled so as to minimize staleness while taking into account view dependencies and priorities [15], and similar problems arise also in distributed databases [6].

[3]Data losses at NASA that were featured in popular media happened for different, but related reasons: While the data itself was preserved well, the knowledge how to decode it had gone lost in the 40 years since the Viking mission.

Determining Optimal Resources. A characteristic of the existing work on crawling is that it assumes that update resources are fixed, and focusses on the best distribution of these resources [7, 12, 21, 17, 23, 29]. Crucially, the units in which the cost of updates and the benefit of the data are measured are not comparable, i.e., the former is usually described by the number of updates, while the latter is described for instance in terms of freshness, staleness, divergence or embarassement. Consequently, optimization of a single factor is not possible.

Work where we find that cost and benefit are measured in the same unit can be found in the data integration domain: In [24], Rekatsinas et al. study the selection of data sources, when each source has a monetary cost, and a diminishing impact on the monetary benefit of the integrated data, thus allowing to study optimization problems regarding how many and which sources to integrate.

4. BACKGROUND

Many aspects of reality are subject to change, for instance affiliations and addresses of persons, webpages, or geographical features. Consequently, information describing reality may become outdated. We label this phenomenon as *information decay*, because, while the data persists, it is the information value of the data that is decaying.

Modeling Decay. To describe decay of information mathematically, a time-variant function describing the probability that an entity is up-to-date is used. We call this function the *decay function* z. The value of $z(t)$ describes the probability that a piece of information is correct after time t. Decay functions must satisfy $z(0) = 1$, and it is plausible to assume that $z(t)$ should be monotonically decreasing with $\lim_{t\to\infty} z(t) = 0$.

Various classes of decay functions such as linear, exponential and geometric decay are discussed in [16]. Linear decay, which is mathematically easy to describe, was already used in the introductory example. In Sec. 8.1 we show that the affiliation information of soccer players are subject to exponential decay, while in [7] the same was shown for the currency of webpages. We therefore in the following instantiate our abstract framework for linear and exponential decay.

Both linear and exponential decay functions are parameterized with a parameter λ, the *decay rate*, which describes the velocity with which entities get outdated.

Linear Decay. Under linear decay, the probability of correctness decreases by a constant amount per time until it reaches zero. Formally, the probability of correctness for an entity under linear decay is

$$z_{lin}(t) = max(1 - \lambda_{lin} t, 0). \quad (1)$$

Instead of using the parameter λ_{lin}, linear decay can also be characterized by the maximal lifetime $t_{\max}$ of an entity, which is calculated as $t_{\max} = \frac{1}{\lambda_{lin}}$.

Exponential Decay. Under exponential decay, the probability of correctness decreases by a constant percentage per time, which implies that it reaches zero only asymptotically. Formally, under exponential decay, the probability of correctness for an entity is

$$z_{exp}(t) = e^{-\lambda_{exp} t}. \quad (2)$$

Exponential decay behaviour is often described using the so-called *half-time* $t_{\frac{1}{2}}$. The half-time is the time after which the probability of an entity to still be correct is exactly 0.5. Given λ_{exp}, the half-time $t_{\frac{1}{2}}$ is calculated as $\frac{ln(2)}{\lambda_{exp}}$. Furthermore, the mean lifetime of an entity can be computed as $\frac{1}{\lambda_{exp}}$.

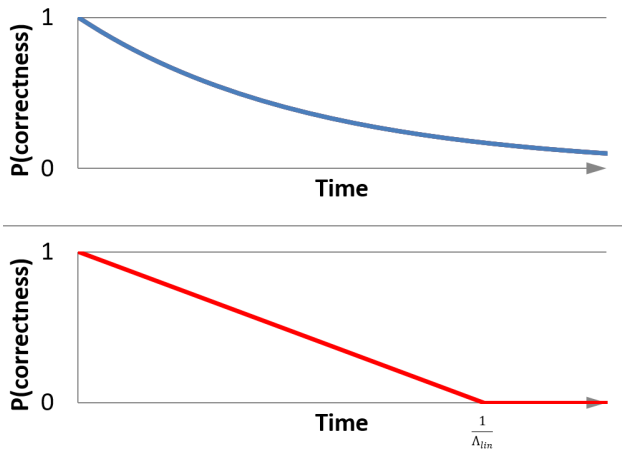

Figure 2: Probability of correctness as a function of time, top under exponential decay, bottom under linear decay.

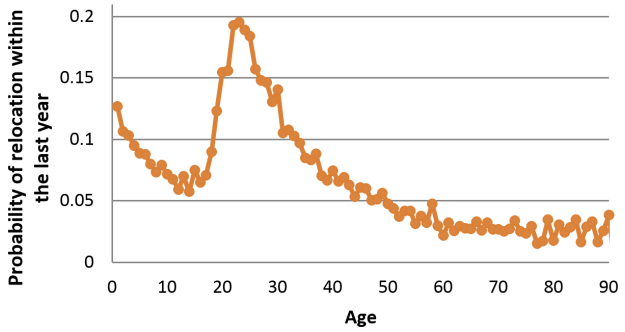

Figure 3: Illustration of varying decay rates. The relocation probability of persons in the taxpayer dataset varies by age.

Both classes of decay functions are illustrated in Fig. 2.

EXAMPLE 1 (DECAY RATES). *We use here the UCI dataset about Californian taxpayers [26] to illustrate how one can compute decay rates, and that variances in decay rates naturally occur. The dataset contains 200k records of Californian residents, each with 42 attributes describing socioeconomic features of the taxpayers. In particular, one of the attributes,* "lived in this house 1 year ago"*, describes whether the person changed residence within the last year. We aggregated this attribute by the age of the taxpayers, obtaining relocation probabilities per age group as shown in Fig. 3. Interesting to see is that newborns are more likely to move than older children, with a possible explanation being that their parents are likely to relocate to places accommodating the bigger family. Less surprising is that the relocation probability peaks at an age of 25, after which it continuously decreases.*

While we do not know the true decay behaviour of address data, assuming that it follows linear or exponential decay, we can compute the corresponding decay coefficients from the relocation probability. The results for persons of age 25, 30, 40 and 50 are shown in Table 1. For instance, observing that 36.7% of all 25-year olds relocate within a year, assuming that addresses are subject to exponential decay, we can compute the decay rate λ_{exp} as 0.457.

Since without interventions, the probability of correctness would steadily decrease towards zero and render data useless, a possible action is to refresh/update the data. The abstract *update operation*

Age	P(Move within last year)	P(No move within last year)	λ_{lin}	λ_{exp}	$t_{\frac{1}{2}}$ (years)
25	36.7%	63.3%	0.37	0.457	1.52
30	27.1%	72.9%	0.27	0.316	2.19
40	15.0%	85.0%	0.15	0.163	4.25
50	9.3%	90.7%	0.09	0.098	7.07

Table 1: Decay rates for linear decay and exponential decay, and half-time for exponential decay for persons of age 25, 30, 40 and 50 in the taxpayer dataset.

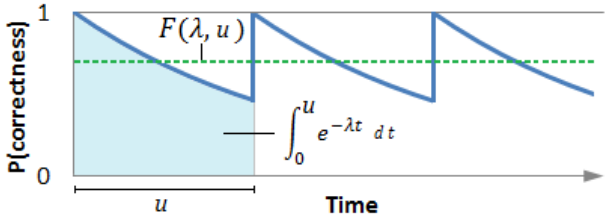

Figure 4: Calculating the average freshness.

that we consider in the following does just that: it retrieves the current correct value for a given entity, thus resetting its correctness probability to 100% for the updated entity. Subsequently, the entity is subject to decay as before, and the correctness probability decreases.

Frequency and Interval. Update frequency is the inverse of update interval, e.g. an update frequency of twice a year corresponds to an interval of half a year between updates, thus, either can be used to describe an update policy. Because for low frequencies intervals are more natural to understand, we mostly use intervals in the following.

Relation of Update Interval and Average Freshness [7]. Given a decay function z, a decay rate λ and an update interval u, the average correctness $F(\lambda,u)$ of any entity with decay rate λ wrt. the update interval u corresponds to the area under the curve z from zero to u, divided by the update interval u (see Fig. 4 for an illustration, where the average freshness is the average height of the blue shaded area):

$$F(\lambda,u) = \frac{\int_0^u z(t)dt}{u}. \tag{3}$$

Instantiating this formula with linear decay (Eq. 1), for $u \leq \frac{1}{\lambda}$, the average freshness under linear decay can be calculated as:

$$F_{lin}(\lambda,u) = 1 - \frac{\lambda \cdot u}{2}. \tag{4}$$

Similarly, if instantiated with exponential decay (Eq. 2), the average freshness is:

$$F_{exp}(\lambda,u) = \frac{1-e^{-\lambda u}}{\lambda \cdot u}. \tag{5}$$

EXAMPLE 2 (UPDATING ADDRESSES). *In Table 2, we show the average freshness of address information of persons of various age depending on the update frequency, assuming exponential*

Age	λ	Update interval U (in years)					
		0.25	0.5	1	2	5	10
25	0.457	94%	89%	80%	66%	39%	22%
30	0.316	96%	93%	86%	74%	50%	30%
40	0.163	98%	96%	92%	85%	68%	49%
50	0.098	99%	98%	95%	91%	79%	64%

Table 2: Average correctness of information for persons of age 25, 30, 40 and 50 in the taxpayer dataset for various update intervals, assuming exponential decay.

decay. As we can see, an update frequency of once a year for 25-year olds yields nearly the same average freshness as an update frequency of once every five years for 50-year olds, and an update frequency of once every two years for the former nearly the same average freshness as a frequency of once every ten years for the latter.

As one can clearly see in the above example, smaller update intervals imply a higher average freshness. But one can also see that the relation is not linear: Doubling the update frequency for 50 year olds updated every 10 years (adding 0.2 updates per year) yields a 15% increase in average freshness, while doubling the frequency if they are already updated yearly (adding 1 update per year) yields only a 3% increase in average freshness. In the next section we focus on determining the optimal update frequency when each entity is updated independently.

5. INDEPENDENT UPDATES

In this section we show how assigning the same unit to the cost of updates and the benefit derived from fresh entities allows to define an update interval optimization problem. We then present solutions for this problem for linear and exponential decay.

Update Cost. Just as there is no free lunch, update operations are seldom without a cost. Updating data may require sending emails, making phone calls, or inspecting websites and data, or may require computating time and bandwidth to extract and process data from the web, and all these factors can be translated into a monetary cost. In the following, we use the letter C to refer to the cost of an individual update.

Benefit. Data is usually updated for a specific purpose, which is related to the business model of the organization maintaining the data. The business model may directly assign a value to fresh entities (i.e., address resellers charge by the number of addresses that they sell), or it may assign a value indirectly, for instance via the increase in revenue that a search engine experiences if it provides more current search results. In the following, we use the letter B to describe the monetary benefit per time that is derived from a correct entity.

Net Income. We can now put cost and benefit in relation, arriving at a *net income* (*NI*) calculated wrt. a fixed update interval u and decay coefficient λ as follows:

$$NI(u) = B \cdot F(\lambda, u) - \frac{C}{u}. \tag{6}$$

By plugging Eq. 4 in for $F(\lambda, u)$, we can calculate the net income under linear decay with $u \leq \frac{1}{\lambda}$ as:

$$NI_{lin}(u) = B - \frac{B \cdot \lambda \cdot u}{2} - \frac{C}{u}. \tag{7}$$

EXAMPLE 3. *Consider that an up-to-date doctor's address gives Spam Inc. a benefit of \$1 pear year, and that each update costs \$1, too. Suppose now we update a 40-years-old doctor once every two years, which gives an average correctness of 85% (see Table 2). Thus, we would gain \$1·0.85 per year, and would spend $\frac{\$1}{2}$ per year for the updates, thus, the yearly net income of an update interval of 2 years is \$0.35.*

Similarly, under exponential decay we can calculate $NI(u)$ by plugging Eq. 5 into Eq. 6, obtaining:

$$NI_{exp}(u) = B\frac{1 - e^{-\lambda u}}{\lambda \cdot u} - \frac{C}{u}. \tag{8}$$

We can now formulate the core problem, which is to maximize the net income $NI(u)$.

Problem: Optimal Update Interval

Input

- Decay function z
- Benefit B per time for up-to-date entity
- Cost C of update operations

Output

- Update interval u that maximizes the net income $NI(u)$

In the following, we show how one can use common algebra to find the maximum under linear and exponential decay.

To simplify the derivation, we note that one can eliminate either B or C by replacing the other constant by the ratio between the two. We choose here to eliminate C, so from now on B describes the ratio between benefit per time unit and update cost. Note also that under linear decay, the optimal update interval u will always be either less or equal $\frac{1}{\lambda}$, or infinity: If there is a way to derive a positive net income, then it must make sense to update the entity before its freshness probability reaches zero. If there is no way to derive a positive net income, the optimal update interval is ∞. Thus, we can safely concentrate on the case where $u \leq \frac{1}{\lambda}$ in the following, while we treat the case where $u_{\text{opt}} = \infty$ in the paragraph "Futile Entities".

To find the maximum of $NI_{lin}(u)$ and $NI_{exp}(u)$, by common calculus we take the derivative and find its zero. For linear decay (Eq. 7), the derivative is:

$$NI'_{lin}(u) = -\frac{B \cdot \lambda}{2} + \frac{1}{u^2}. \tag{9}$$

We can find the zero of this quadratic equation, which is also the maximum of $NI_{lin}(u)$, at $u_{\text{opt}} = \sqrt{\frac{2}{B \cdot \lambda}}$.

Similarly, for exponential decay, the derivative of Eq. 8 is

$$NI'_{exp}(u) = \frac{B \cdot \lambda \cdot u \cdot e^{-\lambda u} + B \cdot (e^{-\lambda u} - 1) + \lambda}{\lambda \cdot u^2}. \tag{10}$$

We find the zero of this equation at the following position:

$$u_{\text{opt}} = \frac{-W(\frac{\lambda - B}{B \cdot e}) - 1}{\lambda}, \tag{11}$$

where W is the Lambert W or product log function. While there is no symbolic way to find u_{opt}, the value can be found with simple numeric optimization techniques (we used bisection).

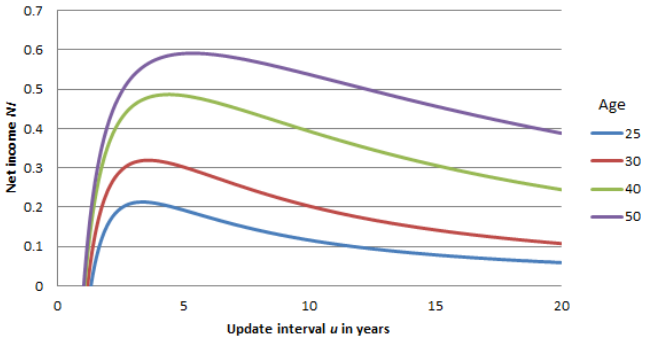

Figure 5: Net income *NI* per year for various person groups wrt. different update frequencies. Yearly benefit *B* and update cost *C* are both 1.

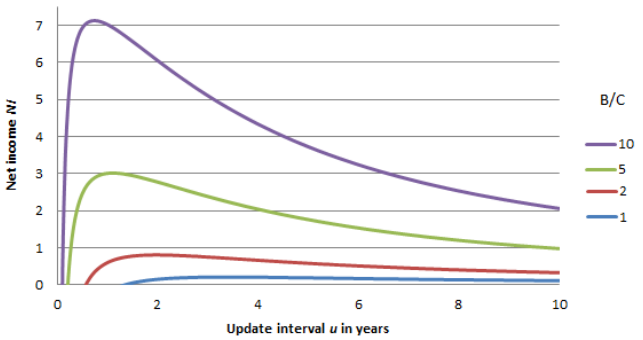

Figure 6: Net income *NI* per year for various benefit/cost ratios calculated for 25-year old doctors.

EXAMPLE 4 (OPTIMAL UPDATE INTERVALS). *Consider again the person groups of various age as shown in Table 1. In Fig. 5, we show the yearly net income for various person groups depending on the update interval, assuming that yearly benefit B and update cost C are both 1. As we can see, the net income is negative for very small update intervals, reaches a maximum at roughly 3 to 5 years, and then drops gradually. We also see that the maximal net income for the 50-year olds (~0.6) is much higher than the maximal net income for the 25-year olds (~0.2). Furthermore, we see that the position of the maximum for the former is at around 5 years, while for the latter it is at around 3 years. Below we report the numeric values of the maxima:*

Group	*Optimal update interval u_{opt} (in years)*	*Maximal yearly net income NI*
25-year olds	*3.38*	*0.21*
30-year olds	*3.61*	*0.32*
40-year olds	*4.42*	*0.49*
50-year olds	*5.36*	*0.59*

To compare the individual optimal update intervals with a uniform update strategy, observe the following:

If we take the average of the yearly update intervals reported above, we would update all entities uniformly every 4.06 years, which would yield an average gain of 0.3973. Using the optimal update intervals for each group instead, the average gain becomes 0.4026, which is an increase of 1.3% over the uniform update intervals.

We can make two observations. First, the maximal possible gain for entities whose freshness is decaying slower can be considerably higher than that for entities decaying faster. Second, in order to achieve the maximal possible gain over all entities, the update interval should be determined separately for each group of entities with a different decay rate.

In Fig. 6 we also show the influence of the benefit-cost ratio $\frac{B}{C}$ onto the net income. As one can see, the higher the benefit-cost ratio, the higher the net income, and also the smaller the optimal update interval.

Futile Entities. Entities may decay so fast that, no matter which update frequency is chosen, the update cost exceeds the benefit derived from the entity. For instance, if a trainee doctor relocates every few months, big update intervals will lead to information that is outdated most of the time, while small update intervals that guarantee a sufficient correctness will be too expensive.

To see that mathematically, observe that for linear decay, the maximal benefit that can be achieved by one update is the area under the decay curve $z(t) = max(1-\lambda_{lin}t, 0)$, which reaches zero at $t = \frac{1}{\lambda}$ and thus covers an area of $\frac{1}{2\lambda}$. If the benefit $\frac{B}{2\lambda}$ is outweighed by the update cost C, that is, if $B \leq 2\lambda C$, then updating the entity does not make sense as the benefit from the update is outweighed by the update cost. Similarly, for exponential decay, the maximal benefit is the area under $z(t) = e^{-\lambda t}$, which is $\frac{1}{\lambda}$, and thus, for values of B and C such that $\frac{B}{\lambda} \leq C$, the benefit derived from any update interval is outweighed by the update cost of the updates.

EXAMPLE 5 (FUTILE ENTITIES). *If benefit per year B and update cost C are fixed to* \$1, *entities under linear decay with a decay rate* ≥ 0.5 *(maximum lifetime 2 years) or under exponential decay with a decay rate* ≥ 1 *satisfy the above conditions, and thus it is more efficient to not update them at all. In our use case, that would be doctors with a maximum lifespan less or equal to 2 years (linear decay), or a half-time less or equal than* $ln(2) \approx 0.69$ *years (exponential decay), respectively.*

Concretely, consider a doctor's address subject to linear decay with a maximum lifetime of 1 year ($\lambda_{lin} = 1$). Updating this address yearly leads to an average freshness of 50%, thus, the yearly cost would be \$1 and the yearly benefit would be \$0.5, giving a net income of \$-0.5 per year. Since the correctness probability after a year is zero, bigger update intervals clearly will not increase the net income. Similarly, consider a smaller update interval, every half a year. This would imply a yearly update cost of \$2, and imply an average freshness of 75%, thus, the net income will be \$-1.25.

6. BULK UPDATES

In some scenarios, data can be acquired in bulks with diminishing costs per entity. For instance, an employee Spam Inc. might call a doctor's office, and whether 3 or 5 doctors are employed there might have little impact on the duration (and hence the cost) of the phone call, compared with the initial effort to set up the call and to get the call recipient to disclose information at all. Similarly, when extracting addresses from web crawls, the cost of local compute time to extract addresses from a crawled document may be small or even neglectible compared with the time and bandwidth cost of loading a webpage.

EXAMPLE 6. *Consider that we want to periodically recrawl doctor affiliations from four webpages that contain 200, 50, 5 and 1 doctor, respectively. The expensive part of web crawling is the*

loading of the documents[4]*, while parsing documents is cheap, so there is no difference between the cost of refreshing the page that contains 200 doctors and the page that contains a single doctor.*

We call update operations that allow for updating multiple entities at once *bulk updates*. Given the availability of bulk updates, the question is how this changes the optimal update interval wrt. the previous analysis for independent updates. Two extremes would be to perform bulk updates the same often as the independent updates, or to increase the frequency of bulk updates proportional to the bulk size, i.e., update a page containing 200 entities 200 times as often as a page containing 1 entity. To analytically find the optimal bulk update frequency, let us look again at Eq. 6 for calculating the net income, which is $NI(u) = B \cdot F(\lambda,u) - \frac{C}{u}$.

So far, $F(\lambda,u)$ described the average freshness of a single entity. We can replace this with the average number of fresh entities from some given set. This value can be calculated simply as sum of the individual F-functions, i.e., for a bulk operation that can refresh n doctors whose freshness is characterized by functions $F_1(\lambda_1,t)$ to $F_n(\lambda_n,t)$, respectively, we can replace $F(\lambda,u)$ by $\sum_{i=1}^{n} F_i(\lambda_i,u)$, obtaining the following formula to describe the net income derived from a bulk update interval u for n entities:

$$NI_{bulk}(u) = B \cdot \sum_{i=1}^{n} F(\lambda_i,u) - \frac{C}{u}. \quad (12)$$

Having this formula, we can determine the update interval u_{opt} that maximizes the net income with techniques similarly as seen before. If all entities in the bulk show the same decay behaviour, the solution is especially easy, as then, we can simply multiply the benefit B with the number of entities in the bulk, and reuse the old model.

EXAMPLE 7. *Let us continue Example 6. Let us assume for simplicity that all doctors are 40 years old and subject to linear decay ($\lambda_{lin} = 0.15$ or a maximum life time of 7 years), and that the benefit per year is* \$1 *and that the cost per page update is \$1. The optimal update interval for a bulk of n doctors can then be calculated as $u_{\text{opt}} = \sqrt{\frac{2}{n\lambda}}$, yielding:*

#Doctors on webpage	$u_{\max}$ (in years)
200	0.26
50	0.52
5	1.63
1	3.66

We can clearly see that group size and $u_{\max}$ are not linearly related, which may be surprising. After a quarter year, a record has decayed with probability $\frac{1}{28}$, which means that out of 200 records, around 7 are expected to be decayed. The reason that lower values of u, e.g. every eighth of a year, do not yield a higher net income is that the time period saved is not sufficient (4 records would be detected earlier for an eighth of a year, yielding an increased net income of $\frac{1}{2}$, but an increased cost of 1 update.

Note that the model can also take into account bulk updates with a cost variable in the number of entities in a group, by adapting the value of C with the group size. We discuss such a case in Sec. 8.4, where we assume a cost of $\sqrt{n}$ for updating a set of n entities.

[4]The pricing of the crawler 80legs.com for instance depends only on the number of crawled URLs and is independent of the amount of extracted information.

7. EXTENSIONS

The model discussed so far still considerably simplifies realistic use cases. In reality, we may expect that outdated records actually incurr a cost, and that the cost of checking updating an entity are not the same. We discuss how to extend our model wrt. these two aspects next.

Cost for Outdated Records. So far we have assumed that all costs come from the update operation. However, outdated records may incur a cost too. For instance, in the use case of the medical advertisement, sending postal advertisement has a cost (work time, material, stamp), and this cost is definitely wasted if an address is outdated. Therefore, it may be plausible that not only the benefit of an outdated entity is zero, but there is also a cost incurred by outdated entities.

EXAMPLE 8. *Consider again the medical address scenario, but assume now that each outdated address incurs a cost of \$1 per year. Remember that in Ex. 4, the optimal update interval for 40-year old doctors was found to be 4.42 years, yielding a net income of \$0.42 per year. Updating every 4.42 years implies an average freshness of 72%, thus, on average, the expenses for outdated records would be \$1·(1-0.72) = \$0.28, thus, with a yearly cost of \$1 for oudated records, the net income would drop to \$0.20 per year.*

To observe that updating every 4.42 years would not be optimal anymore, note that updating every 3 years (without cost for outdated records) would give a lower net income of \$0.46 per year, but would imply an average freshness of 79% and hence, only \$1·(1-0.79)=\$0.21 would need to be deducted for outdated records, thus yielding then a net income of 0.24, which is 20% more than for the previously optimal update interval of 4.42 years.

To calculate the optimal update interval when there is a cost O for outdated records, we can extend Eq. 6 for the net income by incorporating the cost O multiplied by the average nonfreshness, i.e., $1 - F(\lambda,u)$

$$NI(u) = B \cdot F(\lambda,u) - O \cdot (1 - F(\lambda,u)) - \frac{C}{u}. \quad (13)$$

The impact of a cost for outdated records is twofold. First, we will find that the optimal update interval u_{opt} will be smaller. Second, the maximal net income will be lower than before. As a consequence of the second effect, entities for which a positive net income was possible before may now be futile entities, i.e., it may now be impossible to derive a positive net income at all.

For linear decay, the solution for u that maximizes the net income is presented at the end of this section.

Checking vs. Updating. So far we also assumed that the cost of checking the correctness of an entity is the same as the cost of updating an entity. In some situations this might not be the case, e.g., it might be easier to find out that a doctor has moved ("Can I please talk to Dr. Jones"), than to find out her new address ("Can you tell me where Dr. Jones is working now?"). Similarly, as the HTTP protocol provides the functionality to check whether a cached version of a webpage is still up to date, or as some webpages contain "last modified" fields in their body, it may be cheaper to check whether a webpage was modified, than to pull the new version.

EXAMPLE 9. *Assume now that for doctor's addresses, the cost of checking correctness is only \$0.10, while the cost of updating an (incorrect) address is still \$1. Remember that in Ex. 4, the optimal update interval for 40-year old doctors was 4.42 years, yielding a net income of \$0.42 per year. Given that updating every 4.42*

years implies an average freshness of 72%, in 72% of the cases the update would only require the check for 10 ct, while in 28% of the cases, the check would be followed by an actual update operation, thus requiring a total cost of $1.10. Due to the cheaper updates, the net income would rise to $0.60 per year. To observe that the previous optimal interval of 4.42 years would not be optimal any more, note that if we checked the address of a doctor once a year, we would obtain an average freshness of 92%, thus, we would earn $0.92 as benefit, and spend 0.92·$0.10+0.08·$1.10=$0.19 for the updates, obtaining an overall net income of $0.73, or 22% more than with the previously optimal interval.

To calculate the optimal update interval when C_C describes the cost of checking whether an item is still up to date, and C_U the cost for updating an item that is not, we can adapt Eq. 6 by replacing C by C_C (every update operation requires to spend C_C) and multiplying C_U with the probability $z(u,\lambda)$ of finding an outdated entity after time u (we only need to spend C_U if the entity is actually outdated):

$$NI(u) = B \cdot F(\lambda,u) - \frac{C_C}{u} - \frac{C_U * (1 - z(u,\lambda))}{u}. \quad (14)$$

The effect of a cost for refreshs depends on how big this cost is compared with the cost of updates. If C_C is small compared with C_U, we can expect that the optimal update interval gets smaller, and that the maximal net income increases. If however the value of C_C is close to or even larger as C_U, we can expect the optimal update interval to get larger, and the maximal net income to decrease.

If C_C is small compared to C_U, the optimal update frequency will be much higher than previously.

Combined Model. Putting the two extensions together, we arrive at the following formula

$$NI(u) = B \cdot F(\lambda,u) - O * (1 - F(\lambda,u)) - \frac{C_C}{u} - \frac{C_U * z(u,\lambda)}{u}. \quad (15)$$

Instantiating this with linear decay, we obtain

$$NI_{lin}(u) = B \cdot (1 - \frac{\lambda \cdot u}{2}) - \frac{O \cdot \lambda \cdot U}{2} - \frac{C_C}{u} - \frac{C_U \cdot (1 - \lambda u)}{u}. \quad (16)$$

We can now use basic algebra to find the optimal value for u again, and find that the optimum is at:

$$u_{\text{opt}} = \frac{\sqrt{2}\sqrt{C_C + C_U}}{\sqrt{B \cdot \lambda + \lambda \cdot O}}. \quad (17)$$

For exponential decay, there is no symbolic way to find the optimum, but one can use numeric optimization techniques as discussed in Sec. 5.

8. VALIDATION

We next validate four aspects of our framework. In Sec. 8.1 we show that affiliations of soccer players are subject to exponential decay, in Sec. 8.2 we establish variances in decay rates for different job roles publishing at VLDB, in Sec. 8.3 we show how our framework can be used to determine the optimal recrawling frequencies for webpages from a scenario in [7], and in Sec. 8.4 we calculate optimal bulk update frequencies for affiliations from CS department webpages.

8.1 Establishing Exponential Decay

Exponential decay behaviour naturally arises in processes, where intervals between events are distributed following a poisson-distribution. Exponential decay behaviour has already been established for webpages [7]. In this section we show that exponential decay also applies to the probability of a soccer player playing at the same club over time. For some prominent soccer clubs, Wikipedia has historical listings of players, which contain the year that players joined and left a club. For this analysis, we used Manchester United and Bayern München. In Fig. 7 we plot the observed decay behaviour for both clubs (solid lines), together with exponential approximations (dashed lines). We left out the players that stayed less than a year at their club. Note that the data for Manchester United (789 players) ranges back to 1884, while that for Bayern München (366 players) ranges back only till 1965, thus, the observed differences in decay rates need not stem from country-specific differences. We find that the decay behaviour of players of Manchester United can be well approximated by exponential decay with $\lambda = 0.26$ (relative error 0.9%), while that of Bayern München can be well approximated by exponential decay with $\lambda = 0.36$ (relative error 1.2%).

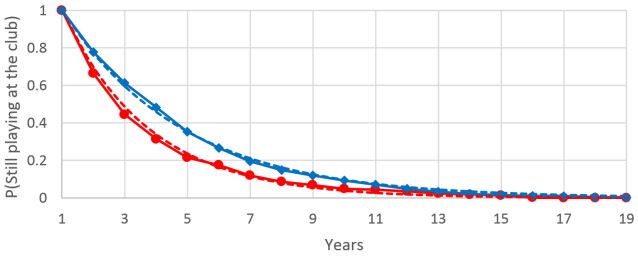

Figure 7: Decay behaviour of soccer players at Manchester United (blue) and Bayern München (red), observed (solid lines), and approximated by exponential decay curves with $\lambda = 0.26$ and 0.36, respectively (dashed lines).wuff

Group	Relocation probability (2010-2015)	n
Students	100%	10
Academic researchers	63%	8
Industrial researchers	23%	13
Professors	11%	19

Table 3: Relocation probabilities of academic personnel.

8.2 Establishing Variable Decay Behaviour

In [7] it has been shown that websites exhibit different decay behaviour, i.e., that some websites are updated daily, while others are updated weekly, monthly, or even more seldom. In this analysis we show that also affiliations of academic personnel exhibit such differences. We use the affiliations of persons that published at VLDB 2010. We chose the 51 first persons listed in DBLP, and grouped them into four categories based on their 2010 job title: PhD students, academic researchers, industrial researchers, professors. We then checked whether their affiliation in November 2015 was different from the one listed on the VLDB 2010 paper. The results are shown in Table 3. Not surprisingly, 10 out of 10 students in 2010 have a different affiliation in 2015. Also affiliations of academic researchers show a relatively high decay, with 5 out of 8 getting outdated in this period. Affiliations of industrial researchers and professors are fairly stable by contrast, with 3 out of 13 and 2 out of 19 getting outdated, respectively.

Benefit-cost ratio $\frac{B}{C}$	Optimal number of monthly updates	Net income gain of optimal frequency wrt. baseline (270 updates/month)
0.1	60	18.5%
0.3	242	0.1%
0.355	*270*	-
0.5	333	0.5%
1	1,033	6.0%
10	4,710	25.8%
100	15,882	31.8%

Table 4: Optimal number of updates in the web crawling scenario from [7].

8.3 Optimal Update Resources in Web Crawling

In this section we discuss a use case concerning the frequency with which a crawler recrawls webpages. In [7], Cho and Molina did measurements on the change frequency of 270 popular websites. Subsequently, they created a model where they grouped the sites into five categories, for which they assumed that 23% were updated on average daily, 15% weekly, 16% monthly, 16% quarterly and 30% yearly. They then assumed to have resources to update the crawled version of every web page once a month, and discussed how to best distribute these 270 updates over the whole set.

The number 270 was hereby completely arbitrary.

In this analysis, we investigate the optimal number of monthly updates for different ratios between monthly benefit and update cost. For ratios between 0.1 and 100, we calculate the number of monthly updates that maximizes the net income, and compare the obtained net income with the net income obtained from an optimal distribution of 270 updates. The results are shown in Table 4.

As we can see, 270 updates are only optimal if the benefit-cost ratio is 0.355. For lower ratios (first two rows), this 270 updates would imply that entities are updated too frequent, for higher ratios (last four rows) it would imply that entities are updated less often than optimal. For instance, if the true value of $\frac{B}{C}$ was 1, it would be better to almost triple the update resources, and to perform 1033 instead of 270 updates per month, obtaining a 6% higher net income. Note that the loss in net income for higher $\frac{B}{C}$ ratios does not grow proportional with the benefit-cost ratio. The reason is that independent of this ratio, 270 updates already achieve an average freshness of $\approx 60\%$, and no update frequency can achieve more than 100%.

As we can see, depending on the true value of $\frac{B}{C}$, a fixed provision of updates which is different from the optimum may lead to significant losses in the net income. For instance, if the true ratio is 100, using resources for monthly updates (ratio 0.355) leads to a reduction of the overall income by 31.8%. Note also that the benefit-cost ratio for search engines may be even much higher, as in the next section we compute an estimated $\frac{B}{C}$ ratio of 16k per day for the Google Search Engine.

8.4 Bulk Updates of Academic Personnel

Let us consider an ad company that targets academic personnel. By visiting university and department webpages, the company can get bulks of name, email addresses and position data. For instance, the website of the CSAIL group at the MIT lists 1024 persons, the website of the CS department at the Free University of Bozen-Bolzano lists 61, and the website of the Chair of Automata Theory at the Technische Universität Dresden lists 22.

Let us call these websites big, medium and small, and let us assume that there are 10 times as many medium and 20 times as many small websites than big websites. Let us also assume that the benefit from a correct record is \$1 per year, and that the cost for manually updating a bulk of persons is the square root of the number of persons in dollars, i.e., refreshing the big, medium and small websites has a cost of \$32, \$7.8 and \$4.7, respectively.

Furthermore we assume that at each university, the ratio between professors, postdocs and PhD students is 1:2:5, thus, if we average the decay rates observed in Sec. 8.2, we arrive at an average decay rate $\lambda = 0.24$.

Using Eq. 10, we find that the optimal update interval for big/-medium/small webpages is every 0.53, 1.13 and 1.50 years, respectively. With these update intervals, the total net income for 20 small, 10 medium and 1 big website would be \$1672 per year.

Let us now see the net income if we use other update periods, namely the optimal one for the big or small websites, or yearly updates:

Uniform update interval (in years)	Total yearly net income (in \$)	Loss wrt. using individually optimal intervals
1.5	1606	3.96%
0.53	1563	6.53%
1	1640	1.92%

As we can see, updating all websites with the same frequency leads to a noticable loss of 1.92% to 6.53% of net income, compared with using for each type of website the individually optimal update interval.

9. DISCUSSION

Computing the optimal update interval under exponential decay (Eq. 10) requires numerical methods, and cannot be calculated in standard Excel. We provide a Java implementation to compute it, which can be downloaded at <*anonymized*> . Executed with values for the two parameters λ and $\frac{B}{C}$, it returns the optimal update interval u_{opt} under exponential decay and the implied net income.

Finding the Correct Decay Function. Finding the correct function class and the right parameters requires either domain knowledge, or a statistical analysis such as the one in Sec. 8.1. The work in [16] shows that in different domains, different classes of decay functions may be applicable. In [7], it was verified that Poisson processes for data change lead to exponential decay behaviour. Even if the class of functions is known, the decay rate needs to be determined. In the example in this paper, we used only the age attribute, but it is likely that there are other attributes that allow more refined predictions.

Determining Cost and Benefit. As good estimates for cost and benefit are crucial for the calculation of the optimal update frequency, we briefly sketch here how these values could be obtained in different scenarios. In the *medical advertisement scenario*, the benefit of an up-to-date address is immediate, as addresses are traded by address resellers, with pricing proportional to the number and correctness of the traded addresses. To determine the cost, one would need to analyze the cost associated with

the various techniques that can be used to refresh addresses. In the *web crawling scenario*, the cost of an update is immediate, via the compute time or network bandwidth that is required for the update. On the other hand, the benefit highly depends on the use of the crawled data. For general web search, determining the benefit requires three complex translations: From information currency to the quality of search results, which in turn translate into user experience, and which in turn translates into monetary value [22].

Applicability. The presented framework can be applied in any domain in which information is getting oudated over time, and update/refresh operations are used to restore the freshness of the data. Besides the scenarios already discussed (address maintenance, web crawling), we imagine this to be the case in view maintenance [30] and caching [3]. Note also that for web crawlers, it is plausible that the benefit derived from outdated webpages is not zero, but that it gradually decreases over time as the webpage evolves. This could be taken into account in our model by adapting the benefit function in the style of [23]. This is not an issue for addresses, as an address is indeed either correct and useful, or outdated and useless.

Search Engine Business. We can instantiate our framework with known values of the Google search engine. Note that this is a hypothetical calculation, the techniques Google uses for determining recrawl frequencies are not public and may be very different[5]. Equation 10 for calculating the optimal update interval has four parameters (B, u, λ and C). Thus, if we fix three of them we can calculate the fourth. We have seen that the cost C for one crawl is in the order of 0.003 Cents. It is reported that the website *(http://quietnightbeds.co.uk/)*, an online store, gets crawled by Google 75 times a day[6] which gives u. If we assume that this webpage changes on average daily (λ), we can compute the value 16,000 for the ratio $\frac{B}{C}$. Thus, given the known value for u, we can deduce that the crawling service (Google) obtains a benefit B of around \$0.48 per day from having the current version of this webpage. Analogously, if the webpage would change twice a day, the benefit would be ≈\$0.24 per day.

10. CONCLUSION

In this paper we have introduced the problem of determining the optimal update interval for information subject to decay, especially relevant given today's scalable cloud resources and crowdsourcing. We have illustrated the problem in two use cases, medical advertisement and web crawling, and have instantiated this framework for entities subject to linear and exponential decay. Our validation showed that the framework can lead to a significant increase of net income in crawling or advertisement, compared with strategies that use a fixed amount of updates or that uniformly distribute updates among entities. Future work might focus on techniques for better estimating the value of the parameter B in web search applications.

11. REFERENCES

[1] M. Baker, K. Keeton, and S. Martin. Why traditional storage systems don't help us save stuff forever. In *1st IEEE Workshop on Hot Topics in System Dependability*, 2005.

[2] C. Batini, C. Cappiello, C. Francalanci, and A. Maurino. Methodologies for data quality assessment and improvement. *ACM CSUR*, 41(3):16, 2009.

[3] C. Bornhövd, M. Altinel, C. Mohan, H. Pirahesh, and B. Reinwald. Adaptive database caching with DBCache. *IEEE Data Eng. Bull.*, 27(2):11–18, 2004.

[4] M. Bouzeghoub. A framework for analysis of data freshness. In *International workshop on Information quality in information systems*, pages 59–67, 2004.

[5] L. Bright and L. Raschid. Using latency-recency profiles for data delivery on the web. In *VLDB*, pages 550–561, 2002.

[6] D. Carney, S. Lee, and S. Zdonik. Scalable application-aware data freshening. In *ICDE*, pages 481–492, 2003.

[7] J. Cho and H. Garcia-Molina. Effective page refresh policies for web crawlers. In *ACM TODS*, pages 390–426, 2003.

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

[9] E. G. Coffman, Z. Liu, and R. R. Weber. Optimal robot scheduling for web search engines. 1997.

[10] N. Craswell, F. Crimmins, D. Hawking, and A. Moffat. Performance and cost tradeoffs in web search. In *ADC*, pages 161–169, 2004.

[11] D. Denev, A. Mazeika, M. Spaniol, and G. Weikum. SHARC: framework for quality-conscious web archiving. *VLDB*, 2(1):586–597, 2009.

[12] J. Edwards, K. McCurley, and J. Tomlin. An adaptive model for optimizing performance of an incremental web crawler. In *WWW*, pages 106–113, 2001.

[13] W. Fan, F. Geerts, and J. Wijsen. Determining the currency of data. *ACM TODS*, 37(4):25, 2012.

[14] F. Geerts, G. Mecca, P. Papotti, and D. Santoro. The LLUNATIC data-cleaning framework. *PVLDB*, 6(9):625–636, 2013.

[15] L. Golab, T. Johnson, and V. Shkapenyuk. Scalable scheduling of updates in streaming data warehouses. *IEEE TKDE*, 24(6):1092–1105, 2012.

[16] B. Heinrich, M. Klier, and M. Kaiser. A procedure to develop metrics for currency and its application in CRM. *JDIQ*, 1(1):5, 2009.

[17] H.-K. C. Ka Cheung Sia, Junghoo Cho. Efficient monitoring algorithm for fast news alert. *IEEE TKDE*, pages 950–961, 2007.

[18] Y. Koren. Collaborative filtering with temporal dynamics. *CACM*, 53(4):89–97, 2010.

[19] A. Labrinidis and N. Roussopoulos. Balancing performance and data freshness in web database servers. In *VLDB*, pages 393–404, 2003.

[20] P. Li, X. L. Dong, A. Maurino, and D. Srivastava. Linking temporal records. *PVLDB*, 4(11):956–967, 2011.

[21] J. McKeeth. Method and system for updating a search engine, July 13 2004.

[22] V. McKinney, K. Yoon, and F. Zahedi. The measurement of web-customer satisfaction: An expectation and disconfirmation approach. *Information systems research*, 13(3):296–315, 2002.

[23] C. Olston and S. Pandey. Recrawl scheduling based on information longevity. In *WWW*, pages 437–446, 2008.

[24] T. Rekatsinas, X. L. Dong, and D. Srivastava. Characterizing and selecting fresh data sources. In *SIGMOD*, pages 919–930, 2014.

[25] T. Schwarz, M. Baker, S. Bassi, B. Baumgart, W. Flagg, C. van Ingen, K. Joste, M. Manasse, and M. Shah. Disk failure investigations at the internet archive. In *Work-in-Progess session, NASA/IEEE Conference on Mass Storage Systems and Technologies (MSST2006)*, 2006.

[26] UCI Machine Learning Repository. California taxpayer dataset. http://archive.ics.uci.edu/ml/datasets/Census-Income+(KDD).

[27] Y. Wand and R. Y. Wang. Anchoring data quality dimensions in ontological foundations. *CACM*, 39(11):86–95, 1996.

[28] R. Y. Wang and D. M. Strong. Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, pages 5–33, 1996.

[29] J. L. Wolf, M. S. Squillante, P. Yu, J. Sethuraman, and L. Ozsen. Optimal crawling strategies for web search engines. In *WWW*, pages 136–147, 2002.

[30] Y. Zhuge, H. Garcia-Molina, J. Hammer, and J. Widom. View maintenance in a warehousing environment. *ACM SIGMOD Record*, 24(2):316–327, 1995.

[5] for a discussion of Google's recrawl frequency see http://blog.boostability.com/how-often-is-google-crawling-my-site/.

[6] http://www.sitepoint.com/increase-search-traffic-getting-site-recrawled-often/