Stochastic Models for Tabbed Browsing

Flavio Chierichetti[*]
Dipartimento di Informatica
Sapienza Università di Roma
Roma 00198, Italy.
chierichetti@di.uniroma1.it

Ravi Kumar
Yahoo! Research
701 First Avenue
Sunnyvale, CA 94089.
ravikumar@yahoo-inc.com

Andrew Tomkins[*]
Google, Inc.
1600 Amphitheater Parkway
Mountain View, CA 94043.
atomkins@gmail.com

ABSTRACT

We present a model of tabbed browsing that represents a hybrid between a Markov process capturing the graph of hyperlinks, and a branching process capturing the birth and death of tabs. We present a mathematical criterion to characterize whether the process has a steady state independent of initial conditions, and we show how to characterize the limiting behavior in both cases. We perform a series of experiments to compare our tabbed browsing model with pagerank, and show that tabbed browsing is able to explain 15–25% of the deviation between actual measured browsing behavior and the behavior predicted by the simple pagerank model. We find this to be a surprising result, as the tabbed browsing model does not make use of any notion of site popularity, but simply captures deviations in user likelihood to open and close tabs from a particular node in the graph.

Categories and Subject Descriptors. H.3.m [**Information Storage and Retrieval**]: Miscellaneous

General Terms. Algorithms, Experimentation, Theory

Keywords. Tabbed browsing, random walks, branching process, stationary distribution, convergence

1. INTRODUCTION

The Opera web browser version 4, released in 2000, was the first to popularize browsing with multiple tabs available within a single window. This *tabbed browsing* paradigm became popular among the technical community, and appeared in Firefox by the following year, and in Safari by 2003. By 2006, all major browsers offered a tabbed browsing capability. Today, as Meiss et al. [22] and Viermetz et al. [27] show, it is increasingly unusual for an online user to access the web through a single tab.

User models like pagerank [23] provide simple, well-known, and mathematically well-understood approaches to thinking about browsing with a single tab in a single window. However, the situation becomes more complex when multiple tabs are introduced. The most salient distinction is that a user no longer traces a single path through a graph, and so may no longer be modeled accurately as a state in a Markov chain corresponding to the underlying graph of web pages or hosts. Rather, the user's browsing may be viewed as a series of "pebbles" that move from node to node in a graph. New pebbles may come into being, and existing ones may split or die out. The operation of moving from one configuration of pebbles to another is not naturally represented as a stochastic matrix, and it is no longer obvious how to think about the "steady state" of such a system.

In this paper, we present a model of tabbed browsing that represents a hybrid between a Markov process capturing the graph of web pages, and a branching process capturing the creation, splitting, and dying of tabs. We present a mathematical criterion to characterize whether the process has a steady state independent of initial conditions, and we show how to characterize the limiting behavior in both cases. We perform a series of experiments to compare our tabbed browsing model with pagerank, and show that tabbed browsing is able to explain 15–25% of the deviation between actual measured browsing behavior and the behavior predicted by the simple pagerank model. We find this to be a surprising result, as the tabbed browsing model does not make use of any notion of site popularity, but simply captures deviations in user likelihood to open and close tabs from a particular node in the graph.

Models of stateful browsing. Browser developers are actively engaged in providing users with new capabilities to improve online navigation. Bookmarks, back buttons (and the corresponding forward buttons), tabs, multiple windows, toolbars, URL bars, auto-completion, search, and many other mechanisms may be seen as offering users a way to move through the web graph using more contextual and stateful information than a naive browsing model would assume. Here we touch briefly on a few of these mechanisms.

Tabbed browsing has two primary manifestations. In the first, a user visiting a page opens a link on the page in a new tab. This is commonly accomplished by right-clicking the link and selecting "Open in new tab," or by holding down the control key while clicking the link. We will refer to this behavior as "control-clicking." In the second manifestation, a user explicitly requests a new tab without specifying the contents, and this tab then opens to the user's (possibly empty) homepage. From there, the user may enter a URL, perform a search through a toolbar or browser chrome search box, and so forth. These two behaviors must be handled differently, as the first depends on the current page and represents a form of transition, while the second is more akin to a restart.

In addition to tabbed browsing, users may also click the "back button" to revisit the previous page of the current

[*] Part of this work was done while the author was at Yahoo! Research.

WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA.
ACM 978-1-60558-799-8/10/04.

tab. Importantly, a second click on the back button does not return the user to the original page, but instead continues backwards to the page loaded before the old page. Hence, implementing or modeling the back button is considerably more complex than allowing a user to follow links backwards in the graph — the process must maintain awareness of the entire chain of links that brought the user to the current page, as the user may choose to sequentially unwind the chain arbitrarily far back.[1] See [14] for a detailed analysis of the process of browsing with back buttons.

The focus of this paper is tabbed browsing, so we do not explicitly model any of the other stateful browsing mechanisms described above. However, the data we employ in evaluating our models is based on anonymized and aggregated browsing trails from Yahoo! toolbar logs, and makes available for each page visited by a user the time of the visit, and the source node (referrer) from which the user arrived at the current page. This information does not allow us to distinguish between a user who opens page A then control-clicks on pages B and C, versus a user who opens A, clicks to B, uses the back button to revisit A, and then clicks to C. As we describe below, however, tabbed browsing is the dominant mechanism by which users follow multiple links from a page, and as new data becomes available, improved estimates of our parameters can be incorporated without changes to the model.

Overview of the model. We will describe how our model captures the behavior of control-clicking to open a link in a new tab. Given this, it is straightforward to incorporate the behavior of opening a new empty tab.

In our model, each node i of the graph has a spawn probability $s_i > 0$ and a death probability $d_i > 0$. We assume that a user begins by opening a browser, which contains a single tab for a page chosen according to some initial distribution. As the process proceeds, the user may open more tabs in the browser, and the process ends when the user closes the last tab, which may never happen.

During each round, the user will follow one edge in one tab, and may optionally kill some tabs and spawn others. The order in which this happens is important for the analysis. Each round contains three phases:

(1) In the first phase, the user must select a tab of interest. She considers each tab in a round-robin fashion, and either kills the tab with probability equal to the death probability of the page in that tab, or selects the tab.

(2) In the second phase, the user control-clicks zero or more edges. She repeatedly flips a coin whose bias is the spawn probability of the given page. Once the coin comes up tails, the phase ends. As long as the coin comes up heads, she control-clicks on an outlink from the page, chosen from the outdegree distribution, and the destination page opens in a new tab.

(3) Finally, in the third page, the user picks a link from the outdegree distribution of the current page and clicks it, loading the destination into the current tab.

The steady-state measure is given by the expected fraction of times, in the limit, a particular page is viewed (or more accurately, loaded in a tab); we call this the *tabrank* of the process. We observe first that if there are constants s and d such that $s_i = s$ and $d_i = d$ for all i then the model may be reduced to a simple Markov process. In fact, if $s = 0$ then tabrank is exactly the steady state of the pagerank process with restart probability equal to d and with reset distribution equal to the initial distribution of the tabbed browsing process.

If the s_i's and d_i's are different, then this is no longer true and the steady state is more complicated. In fact, our analysis shows that depending on the eigenvalue of a certain matrix, the tabbing browsing process may eventually terminate or with positive probability, run forever. In either case, one can define a version of tabbed browsing with restart (mimicking the scenario when the user restarts the browser with a single tab). The steady-state measure remains well-defined.

Organization. The remainder of the paper proceeds as follows. Section 2 describes the related work. In Section 3 we describe our model formally, and in Section 4 we present our theoretical results on the convergence conditions. Section 5 contains our experimental analysis. Finally, Section 6 presents some concluding remarks.

2. RELATED WORK

The random surfer model, first popularized in the pagerank paper [10, 23], has been studied extensively. In this model, there are two tunable parameters: the teleportation probability and the restart distribution. There has been a long line of research trying to understand the effect of these parameters on the steady state [3, 7]; we refer to the surveys on pagerank and other link-based analysis [6, 8, 19, 26].

The basic pagerank model has been extended in a variety of ways. Haveliwala [18] proposed topic-sensitive pagerank where the restart process happens from a topically-focused subset of webpages. Lempel and Moran [20] proposed a different random walk model and studied its robustness to topic drift and spam. Pagerank-like methods have also been to solve other problems arising in web search: estimating trust of websites [17], combating spam [5], identifying webpage decay [4], and many more.

There have been several attempts to make the browsing model of pagerank more realistic and closer to the actual user behavior, especially taking into accounts of the features of the (ever-changing) web browser. The eventual goal of many of these modifications is to obtain a query-independent quality-based ordering of the web pages. Gonçalves et al. [16] observe that the diversity of sites visited by individual users is smaller and more broadly distributed than what is predicted by the classical pagerank model. They introduce the *bookrank model*, where a list of web pages ranked by the number of previous visit (bookmarks), is the state of the Markov chain. At each step, the user chooses a bookmark according to some probability distribution and visits the bookmark. With the remaining probability, the user navigates locally or hits the browser back button. In principle, their work can encapsulate tabbed browsing; however, they only provide simulations and no formal analysis. Liu et al. [21] proposed *BrowseRank*, where the user browsing graph is used in conjunction with webpage dwell times in order to estimate page importance. They employ continuous-time Markov processes to incorporate the length of time spent on a page. They do not address the tabbed browsing process. Sydow [25] and Bouklit and Mathieu [9] considered the effect of "back buttons" in the browser. They augment

[1] Browsers may place some limit on how much history is accessible through the back button.

the browsing activity with a bounded history stack — this blows up the state space of the underlying chain.

On the theoretical front, Fagin et al. [14] analyzed standard random walks when there are "back buttons" available. Their analysis does not seem easily extensible to the tabbed browsing setting, which has a branching process component. Alon et al. [1], Efremenko and Reingold [11], and Elsässer and Sauerwald [12] considered parallel random walks. Parallel random walks can model simultaneously open tabs, but do not capture the birth-death aspect of tabbed browsing. Etessami and Yannakakis [13] formulated recursive Markov chains that can invoke one another (called *multi-type branching processes*) and studied their dying probabilities and the associated computational issues; while our model is a extreme special case of theirs (each node can recursively invoke itself), the simplicity of our model makes it amenable to a self-contained analysis of both the dying probability and even the steady state. For a detailed account of branching processes, see the book by Athreya and Ney [2].

3. THE TABBED BROWSING MODEL

Let P be an n-state Markov chain where $P_{i,j}$ denotes the *transition probability* from state i to state j. We have $\forall i, j\ P_{i,j} \geq 0$ and $\forall i, \sum_{j=1}^{n} P_{i,j} = 1$. Here, the states correspond to webpages and $P_{i,j}$ denotes the probability of navigating from webpage i to webpage j. We use $i \to j$ to denote the hyperlink from i to j.

Our goal is to study the effect of adding tabs to a basic browsing model such as the pagerank model [23]. As in pagerank, we assume that P is ergodic (i.e., the graph underlying the Markov chain is strongly connected and that P is aperiodic).

Further, we assume that for each page i, there exists two probabilities $s_i, d_i \in (0, 1)$ that represent the *tab spawn* probability and and the *tab death* probability respectively. Denote $\Sigma = (s_1, \ldots, s_n)$ and $\Delta = (d_1, \ldots, d_n)$.

We now describe the stochastic process of *tabbed browsing.* We assume that each user maintains a queue of open tabs that she visits in order. The browser starts with a single tab showing a webpage (chosen according to some distribution). When looking at the current tab with page i, the user acts as follows.

(1) She flips a coin with probability d_i. If it comes up tails, she goes to the next step. If it comes up heads, she closes the current tab, and starts all over again with the page on the next tab in the queue, if it is non-empty. If the queue becomes empty, she stops browsing.

(2) She flips a coin with probability s_i. If the coin comes up heads, she control-clicks on a hyperlink $i \to j$ on the page i, chosen randomly according to $P_{i,*}$; this is interpreted as the user loading the page j in a new tab and adding it to the end of the queue of open tabs. She then goes back to the current step. If the coin comes up tails, she goes to the next step.

(3) She choose a link $i \to j$ according to $P_{i,*}$; this is interpreted as the page in the current tab changing from i to j. Then she starts all over again with page j.

4. ANALYSIS

The goal in this section is to analyze the stochastic process proposed in Section 3. In particular, we are interested in the steady-state characteristics of the process and its dependence on the Markov chain and the tab spawn and tab death probabilities.

We say that the random-tab process *eventually ends* (or just *ends*) if, at some point, the user closes the last tab in the queue. We consider the following question.

> For which P, Σ, and Δ, does the process eventually end with probability 1?

We address this problem in a series of steps. First of all, let $S_f(j)$ be the random variable counting the number of visits to page j by a random tab process starting with page f in the initial tab. The first view of page f is counted and therefore $S_f(f) \geq 1$ with probability 1. Let $e_f = (0, \ldots, 0, 1, 0, \ldots, 0)$ be the unit vector with a 1 in the fth coordinate (corresponding to the page loaded in the initial tab of the process). Then, we have the following system.

$$\begin{aligned} E[S_f] &= e_f + d_f \cdot \vec{0} + (1 - d_f) \cdot \left(\sum_{\ell=0}^{\infty} (\ell \cdot s_f^{\ell} \cdot (1 - s_f) \right. \\ &\qquad \left. \cdot \sum_{j=1}^{n} (P_{f,j} \cdot E[S_j])) + \sum_{j=1}^{n} (P_{f,j} \cdot E[S_j]) \right) \\ &= e_f + (1 - d_f) \cdot \left(1 + \sum_{\ell=0}^{\infty} \left(\ell \cdot s_f^{\ell} \cdot (1 - s_f) \right) \right) \\ &\qquad \cdot \sum_{j=1}^{n} (P_{f,j} \cdot E[S_j]) \\ &= e_f + \frac{1 - d_f}{1 - s_f} \cdot \sum_{j=1}^{n} (P_{f,j} \cdot E[S_j]) \,. \end{aligned}$$

It is easy to see the following.

LEMMA 1. *A page is visited infinitely often in expectation if and only if all the pages are visited infinitely often in expectation. That is, $E[S_f(j)] = \infty$ for some f, j iff $E[S_{f'}(j')] = \infty$ for all f', j'.*

PROOF. By our assumptions on P, Σ, and Δ, with positive probability, one can get from any page to any other page in at most finitely many steps. The statement follows. □

The following observation will be used by the main result of the section, to distinguish between the cases where the process eventually ends with probability 1 or survives with positive probability.

LEMMA 2. *If the expected number of visits to some page (and thus to all pages) is finite, then the process eventually ends with probability 1.*

PROOF. Let $\epsilon > 0$ be fixed. Indeed, by the Markov inequality, the probability that $S_f(j) \geq (n/\epsilon) \cdot E[S_f(j)]$ is at most ϵ/n. By a union bound over the n pages, we have that with probability at least $1 - \epsilon$, the process will end after at most $(n/\epsilon) \cdot \sum_{j=1}^{n} E[S_f(j)]$ steps. □

Before continuing with the analysis, we note that a statement such as "the process eventually ends if and only if the expected number of visits to each page is finite" does not hold in general.

LEMMA 3. *There are P, Σ, Δ such that the process will eventually end with probability 1 even if the expected number of visits to some page (and thus to all pages) is infinite.*

Proof. Suppose that P is a Markov chain with a single state and $P_{1,1} = 1$. Further, suppose that $0 < s_1 = d_1 < 1$. Then, $E[S_1(1)] = 1 + E[S_1(1)]$, so $E[S_1(1)] = \infty$. On the other hand, the expected number of "children" per tab is 1. Therefore, by the general branching process theorem [15], the process ends with probability 1. □

To state and prove our main result, we assign *epochs* to each tab. The initial tab will have epoch 0. Given a tab x of epoch t, all the tabs obtained by clicking, or control-clicking on x will have epoch $t+1$ (they are the *children* of x). Let $k_i^{(t)}$ $(1 \le i \le n)$ represent the number of tabs of epoch t showing page i. We will have $k_f^{(0)} = 1$ and $k_i^{(0)} = 0$ for each $i \neq f$.

Since the tabs are loaded in a queue-like manner, and since each single tab will eventually die with probability 1, we have that each opened tab (regardless of its epoch) will be visited. Thus, $S_f = \left(\sum_{t=0}^{\infty} k_1^{(t)}, \ldots, \sum_{t=0}^{\infty} k_n^{(t)}\right)$, where S_f (as defined before) counts the number of times each page will be loaded by our process, assuming to start from a single tab with page f.

Observe that the expected number of tabs of epoch $t+1$ open on page j $(1 \le j \le n)$ is equal to

$$E\left[k_j^{(t+1)} \mid \left(k_1^{(t)}, \quad \ldots, \quad k_n^{(t)}\right)\right] = \sum_{i=1}^{n} \left(P_{i,j} \cdot \frac{1-d_i}{1-s_i} \cdot k_i^{(t)}\right).$$

By the linearity of expectation, we have

$$E\left[k_j^{(t+1)}\right] = \sum_{i=1}^{n} \left(P_{i,j} \cdot \frac{1-d_i}{1-s_i} \cdot E\left[k_i^{(t)}\right]\right).$$

Let

$$A = \begin{pmatrix} P_{1,1} \cdot \frac{1-d_1}{1-s_1} & \cdots & P_{1,n} \cdot \frac{1-d_1}{1-s_1} \\ \vdots & \ddots & \vdots \\ P_{n,1} \cdot \frac{1-d_n}{1-s_n} & \cdots & P_{n,n} \cdot \frac{1-d_n}{1-s_n} \end{pmatrix},$$

and write

$$\begin{aligned} &\left(E\left[k_1^{(t+1)}\right], \quad \ldots, \quad E\left[k_n^{(t+1)}\right]\right) \\ &\qquad = \left(E\left[k_1^{(t)}\right], \quad \ldots, \quad E\left[k_n^{(t)}\right]\right) \cdot A. \end{aligned}$$

By induction, we get

$$\left(E\left[k_1^{(t)}\right], \quad \ldots, \quad E\left[k_n^{(t)}\right]\right) = e_f \cdot A^t \quad \text{for } t \ge 0.$$

Thus,

Lemma 4. $E[S_f] = e_f \cdot \sum_{t \ge 0} A^t$.

Let us now consider the matrix A. We know that, for each i, $\sum_j P_{i,j} = 1$, that for each i, j, $P_{i,j} \ge 0$, and that for each i, $0 < s_i, d_i < 1$. Note that A is irreducible, aperiodic, and non-negative, thanks to the irreducibility and aperiodicity of P.

Our analysis will make use of the irreducible-aperiodic Perron–Frobenius Theorem (see [24, Theorem 1.1]) for non-negative[2] matrices. In the following, we say that x is a non-negative (resp., positive) vector if $x(i) \ge 0$ (resp., $x(i) > 0$), for each i. We say that x is a unit vector if $\sum_i x(i) = 1$. The null vector is the vector having 0 in each coordinate.

Theorem 5 (Perron–Frobenius). *Suppose M is a non-negative, aperiodic, and irreducible $n \times n$ matrix. Then, (i) M has a real eigenvalue $\rho > 0$ of multiplicity 1. The eigenvalue ρ is such that (ii) for each other eigenvalue $\lambda \neq \rho$ of M, we have $|\lambda| < \rho$. Further, (iii) the eigenvalue ρ admits exactly one positive unit left eigenvector, and (iv) exactly one positive unit right eigenvector.*

We also use the following theorem from the theory of matrix powers approximation (see [24, Theorem 1.2]).

Theorem 6. *Suppose M is a non-negative, aperiodic, and irreducible $n \times n$ matrix. Let ρ be its eigenvalue of maximum norm, and $\Lambda < \rho$ be the maximum norm of the other eigenvalues of M. Further, let x^* be the positive unit left eigenvector of ρ. Then, if v is a non-null, non-negative vector, there exists $c > 0$ such that*

$$v \cdot M^t = \left(c \cdot \rho^t \pm O_t\left(t^{n-1} \cdot \Lambda^t\right)\right) \cdot x^*.$$

Let ρ be the spectral radius of A, i.e., ρ is the largest eigenvalue of A. Let τ be the unique non-negative unit-norm left eigenvector associated with ρ, i.e., $\sum_i \tau(i) = 1$ and $\tau A = \rho \tau$. We say that ρ is the *tabrate* of the process, and if $\rho > 1$, call τ to be its *tabrank* vector. We now state and proving our main theorem.

Theorem 7 (Tabbed Browsing). *If $\rho < 1$, then the tabbed browsing process eventually ends with probability 1. If $\rho > 1$, then with positive probability, the tabbed browsing never ends.*

Proof. We start with the case $\rho < 1$. Let f be the page loaded on the original tab. Recall that the vector (indexed by the pages) counting the expected number of epoch t tabs is $e_f \cdot A^t$. By Theorem 6, we can upper bound this expectation by

$$\left(e_f \cdot A^t\right) \le O_t(\rho^t) \cdot \sum_i x^*(i) = O_t(\rho^t).$$

Since $E[S_f] = \sum_{t=0}^{\infty} e_f \cdot A^t$, we have

$$\begin{aligned} E[S_f] &\le \left(O_t\left(\sum_{t=0}^{\infty} \rho^t\right), \ldots, O_t\left(\sum_{t=0}^{\infty} \rho^t\right)\right) \le \\ &\le \left(O_t\left(\frac{1}{1-\rho}\right), \ldots, O_t\left(\frac{1}{1-\rho}\right)\right). \end{aligned}$$

Thus, $E[S_f]$ is finite coordinate-wise and therefore we can apply Lemma 2 to conclude that the process eventually ends with probability 1.

We now consider the case $\rho > 1$. Let τ be the left eigenvector with eigenvalue ρ, with $\sum_i \tau(i) = 1$.

We now describe what we will call phase 1. We start from some page i^* with one tab. Since $0 < s_i, d_i < 1$, for each i, in epoch 0 we will spawn N tabs from this initial tab with positive probability, for some large enough N to be fixed later. Since the Markov chain P is irreducible and aperiodic, there will exists some finite (albeit possibly exponential) epoch t_0 such that with positive probability, (i) no dying/spawning events happened in any of the epochs $1, \ldots, t_0 - 1$ and (ii) for each $1 \le i \le n$, the fraction of epoch t_0's tabs in state i is within a $(1 \pm \epsilon)$ multiplicative factor of $N \cdot \tau(i)$. We condition on this event.

We now move to phase 2. Let $X_{i,t}$ be the number of tabs on state i at epoch t. Let $X_{\vec{i,t}}$ be the sum of the number of

[2] Throughout the paper, we use the term "non-negative" to mean "non-negative and real".

tabs of epoch t that transitioned from state i to some other state, and the number of the children they spawned in epoch t. Let $X^{\rightarrow}_{i,j,t}$ be the number of tabs and of their children transitioning from state i to state j at the end of epoch t. Then, $X^{\rightarrow}_{i,t} = \sum_{j=1}^{n} X^{\rightarrow}_{i,j,t}$. Also, $X_{j,t+1} = \sum_{i=1}^{n} X^{\rightarrow}_{i,j,t}$.

Then,

$$E[X^{\rightarrow}_{i,j,t} | X_{i,t}] = P_{i,j} \cdot \frac{1-d_i}{1-s_i} \cdot X_{i,t}.$$

We want to get a bound on $X^{\rightarrow}_{i,t}$. We do so in three steps. Let $X^{+}_{i,t}$ be the number of tabs on state i at the beginning of epoch t that do not die at epoch t. Then $E[X^{+}_{i,t}] = (1-d_i) \cdot X_{i,t}$. By the Chernoff bound,

$$\Pr[X^{+}_{i,t} < E[X^{+}_{i,t}] - \sqrt{3 \cdot X_{i,t} \cdot \ln X_{i,t}}] < e^{-4 \ln X_{i,t}} = (X_{i,t})^{-6}.$$

For each of the surviving $X^{+}_{i,t}$ tabs, a geometric random variable Y_{1-s_i} (taking values in $\mathbf{Z}^{+}$) of stopping probability $1-s_i$ (and thus mean $(1-s_i)^{-1}$) is sampled; the corresponding surviving tab will have Y_{1-s_i} children (including itself) in epoch $t+1$. We thus want to bound a sum of $X^{+}_{i,t}$ independent geometric random variables Y_{1-s_i}.

By the Chernoff bound, the probability of getting more than $(1-s_i) \cdot d + \sqrt{2d \ln d}$ heads in d coin flips with head probability $1-s_i$ is less than d^{-6}. Thus, the probability of the sum of $(1-s_i) \cdot d + \sqrt{2d \ln d}$ independent geometric random variables Y_{1-s_i} is less than d is $< d^{-6}$. Choosing $d = \frac{1}{1-s_i} \cdot k - O(\sqrt{k \log k})$, we have

$$\Pr\left[\sum_{i=1}^{k} Y_{1-s_i} < \frac{1}{1-s_i} \cdot k - O(\sqrt{k \log k})\right] < O\left(k^{-6}\right).$$

Thus, by implicitly conditioning on $X^{+}_{i,t} \geq (1-d_i) \cdot X_{i,t} - \sqrt{3 \cdot X_{i,t} \cdot \ln X_{i,t}}$, we have

$$\Pr[X^{\rightarrow}_{i,t} < E[X^{\rightarrow}_{i,t}] - O(\sqrt{X_{i,t} \log X_{i,t}})] < O(X_{i,t}^{-6}).$$

Thus, with probability at least $1 - O(X_{i,t}^{-6})$, we have $X^{\rightarrow}_{i,t} \geq E[X^{\rightarrow}_{i,t}] - O(\sqrt{X_{i,t} \log X_{i,t}}) = \frac{1-d_i}{1-s_i} \cdot X_{i,t} - O(\sqrt{X_{i,t} \log X_{i,t}})$. Applying the Chernoff bound again, we get that the number of tabs that actually transition to state j is such that

$$\Pr[X^{\rightarrow}_{i,j,t} < E[X^{\rightarrow}_{i,j,t}] - O(\sqrt{X_{i,t} \log X_{i,t}})] < O(X_{i,t}^{-6}).$$

That is, with probability at least $1 - O(X_{i,t}^{-6})$, we have that $X^{\rightarrow}_{i,j,t} \geq P_{i,j} \cdot \frac{1-d_i}{1-s_i} \cdot X_{i,t} - O(\sqrt{X_{i,t} \log X_{i,t}})$.

Let $X_t = \sum_i X_{i,t}$. By a union bound over all i, j's, we have that

$$\Pr\left[\exists j \mid X_{j,t+1} < \sum_i \left(P_{i,j} \cdot \frac{1-d_i}{1-s_i} \cdot X_{i,t}\right) - \right.$$
$$\left. -O\left(n \cdot \sqrt{\sum X_t}\right)\right] < O(n^2 \cdot \min_i X_{i,t}^{-6}).$$

If this bad event does not happen, we say that the corresponding step of phase 2 was successful.

We will choose the N of phase 1 at the end of phase 2, to guarantee that in each of the infinite steps t of phase 2, the $X_{i,t}$'s will increase; this will ensure that the error term of the previous probability inequality, and its error probability, are both negligible.

We will prove by induction that, with positive probability, the $X_{i,t}$'s, for each i and each t of phase 2, satisfy $X_{i,t} \geq (1-\epsilon) \cdot \tau(i) \cdot X_t$ (recall that τ was the tabrank vector), if a large enough N is chosen. This is true at the end of phase 1. Recall that $\tau A = \rho \tau$. Thus, if $\chi_t = (X_{1,t}, \ldots, X_{n,t})$, and if step t was successful, we have

$$\chi_{t+1} \geq (1-\epsilon)\chi_t A \geq X_t (1-\epsilon)^2 \tau A = X_t \rho (1-\epsilon)^2 \tau.$$

Choosing ϵ sufficiently small (and, consequently, N sufficiently large) such that $\rho(1-\epsilon)^2 > 1$, gives us a lower bound on χ_{t+1} that is multiplicatively larger than the lower bound we had on χ_t. Thus, we can reapply the analysis to step $t+1$. If step $t+1$ happens to be successful we can do it again, and so on. We upper bound the probability p of eventually ending, by upper bounding the probability that some step will be unsuccessful. If t_1 is the first step of phase 2, by the union bound, the upper bound is

$$\begin{aligned} p &\leq \sum_{t=t_1}^{\infty} O\left(n^2 \cdot \min_i X_{i,t_1}^{-6}\right) \\ &\leq \sum_{t=t_1}^{\infty} O\left(n^2 \cdot \left(((1-\epsilon)^2 \cdot \rho)^{t_1} \min_i X_{i,t_1}\right)^{-6}\right) \\ &\leq \frac{1}{1-((1-\epsilon)^2 \cdot \rho)^{-6}} \cdot O\left(n^2 \cdot \min_i X_{i,t_1}^{-6}\right) \\ &\leq O\left(n^2 \cdot \min_i X_{i,t_1}^{-6}\right). \end{aligned}$$

The latter happens to be $o(1)$, if a large enough N is chosen in phase 1.

Thus, with positive probability both phase 1 and phase 2 are successful, and the process never stops. □

We observe that the previous proof only gives an exponential lower bound on the probability of not eventually ending, even if $\rho > 1$. This is in fact unavoidable. Indeed, consider a directed cycle on $\{1, \ldots, n\}$ as the underlying Markov chain (the probability of transitioning from one node of the cycle to the next will be 1). Let $d_i = 1/2$, for $i = 1, \ldots, n$ and let $s_i = 1/n$ for $i = 1, \ldots, n-1$ and $s_n = 1 - \exp(-n)$. Then, an easy calculation shows that $\rho > 1$. But the probability of not eventually ending will be $2^{-\Omega(n)}$.

Observe that τ is in general different from the limiting distribution vector of P, unless $d_i = d, s_i = s$, for all i's, in which case τ *is* the limiting distribution vector of P.

4.1 Tabbed browsing with restart

By the tabbed browsing theorem (Theorem 7), if $\rho < 1$, then the tabbed browsing process eventually ends. To force the process to never end, we add the following simple modification to our tabbed browsing process: whenever the user closes the last open tab, she opens a new tab (with epoch one more than that of the last closed tab) on a page chosen according to some probability distribution π. In this way, the tabbing browsing process will never end. We now proceed to analyze this process.

Let $\ell_i^{(t)}$ be the random variable denoting the number of times page i has been loaded in the first t epochs of our process and let $\ell^{(t)} = \sum_{i=1}^{n} \ell_i^{(t)}$.

THEOREM 8. *Suppose $\rho > 1$. Then, for each $i = 1, \ldots, n$ we have*

$$\lim_{t \to \infty} \frac{E[\ell_i^{(t)}]}{E[\ell^{(t)}]} = \tau(i).$$

Proof. Observe that, with probability 1, the number of times the process is restarted will be finite (since each execution of our original process will go on forever with positive probability, independently). The expected number of page loads in the process executions that die is thus a constant number (in the sense that it does not grow with t, bu t it can still be exponential in n), between 0 and some $c = c(P)$.

Thus, we can restrict ourselves to look at the last (never-ending) execution of the original tab process. An application of Theorem 6, observing that the expected size of the epochs grows exponentially, completes the proof. □

Observe that the limiting theorem uses the index of the epoch as the growing variable. If we were to use another notion of "time", say, the number of iterations of our process (with the $\bar{\ell}_i^{(t)}$'s, and $\bar{\ell}^{(t)}$ defined accordingly), then we can show with the same proof that there is an infinite subsequence[3] of the $\frac{\bar{\ell}_i^{(t)}}{\bar{\ell}^{(t)}}$ sequence that has the limit $\tau(i)$.

We also notice that the reset distribution π does not play a role in determining tabrank, if $\rho > 1$.

On the other hand, if $\rho < 1$, then the following holds.

Theorem 9. *Suppose $\rho < 1$. Then, for each $i = 1, \ldots, n$ we have*

$$\lim_{t\to\infty} \frac{E[\ell_i^{(t)}]}{E[\ell^{(t)}]} = \frac{\sum_{f=1}^{n} \pi(f) \cdot E[S_f(i)]}{\sum_{f=1}^{n} \pi(f) \cdot \sum_{j=1}^{n} E[S_f(j)]}.$$

The proof directly follows from the fact that each execution of the original non-restarting process ends with probability 1 after finitely many steps.

5. EXPERIMENTS

5.1 Data

The results in this section are based on the click-stream data gathered from users of the Yahoo! toolbar. Data is included only for those users who voluntarily "opt in" to having their data collected for such purposes, and all personally identifying information was removed from the data before performing the experiments. The toolbar data contains for each user, the list of hosts and the edges visited. We assume that a URL with no referrer represents a *tab restart* event and a URL that does not refer to any other URL represents a *tab death* event.

We use toolbar data sampled from three time periods: January 20, 2008, Mar 18–25, 2009, and July 18–25, 2009. The latter period is the largest dataset, and contains approximately 5M nodes and 10M edges. We will refer to this dataset below as the *primary dataset.*

We first analyze the data in order to estimate spawn, death, and restart probabilities for each site. The analysis operates as follows. For a host h, we define the following:

degree(h) = the number of times a user traversed a hyperlink originating from page h,

leaf(h) = the number of times a user loaded a page on h but did not follow a link from the page, and

nonleaf(h) = the number of times a user loaded a page on h from which the user followed at least one link.

Based on these values, we estimate the death and spawn probabilities as follows:

$$d_h = \frac{\text{leaf}(h)}{\text{leaf}(h) + \text{nonleaf}(h)},$$

$$s_h = 1 - \frac{\text{nonleaf}(h)}{\text{degree}(h)}.$$

We now give some justification for these estimates. Recall that in the tabbing browsing model, the user first decides whether to close the tab by flipping a biased coin with probability d_h. If the user does not close the tab, then the model stipulates that there will be exactly one transition after zero or more spawn events. Thus, an unbiased estimator is provided by measuring the probability that a page load yields a transition from that page to another page: leaf$(h)/($leaf$(h) +$ nonleaf$(h))$. The spawn probability may then be estimated as follows. If the user does perform at least one transition from a node, then the model stipulates that all but one of the transitions will be spawn events. We may therefore simply estimate the spawn probability that is most likely to generate the observed number of spawn events for each instance of a nonleaf page load.

5.2 Sensitivity analysis of smoothing

We adopt a simple scheme to smooth these estimates. First, we take $\bar{d}$ and $\bar{s}$ to be the mean death and spawn probabilities over all hosts. We then smooth the death estimates by adding some smoothing parameter Δ to the denominator and $\bar{d}\Delta$ to the numerator. Likewise, for spawn estimates, we add Δ to the denominator and $\bar{s}\Delta$ to the numerator.

We perform an experiment on a small sample dataset to determine the sensitivity of the algorithm to smoothing. We compute pagerank and smoothed tabrank, then compare the steady-state results of each to the actual steady-state, using ℓ_1 distance. Let A be the actual distribution, PR be the pagerank distribution, and TR be the tabrank distribution. Figure 1 plots $||A - \text{PR}||_1/||A - \text{TR}||_1$, so larger scores represent a better approximation of the steady state. Based on these results, we set $\Delta = 50$ for all further experiments.

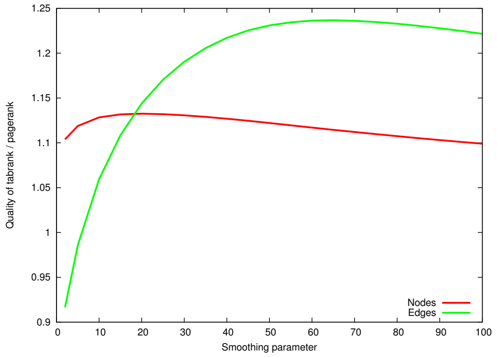

Figure 1: Sensitivity to the smoothing parameter.

In addition to this smoothing, a further form of pre-processing must be performed. For actual users, the browsing process ultimately ends, as must all things in life. However, the measured spawn and death probabilities may lead to a process that does not ultimately end. We check for certain degenerate cases in which a host's self-loop may cause the process

[3] That is, the one containing all and only the time steps corresponding to the last tab of an epoch.

to run forever. If $P_{i,i}(1 - d_i)/(1 - s_i) > 0.95$, we normalize it to be 0.95 in order to keep these "factory" hosts from generating infinite on-host transitions.

5.3 Birth–death probabilities for some sites

The top picture of Figure 5.3 shows the spawn, death, and restart probabilities of the primary dataset. As the restart probabilities drop off so quickly, they are shown in log-scale on the right-hand axis. As the figure shows, most death probabilities are around 0.7, and most spawn probabilities are around 0.1. Restart probabilities cover a wide range, but the vast majority of nodes are below 10^{-7}, as we would expect given the size of the graph.

The bottom picture shows a scatter plot of spawn and death probabilities. Each host represents a host, and the x-axis shows the spawn probability for the host, with the y-axis showing the death probability. Scaling on the axes are linear, and while most mass has lower spawn probabilities, there is significant correlation between the two values.

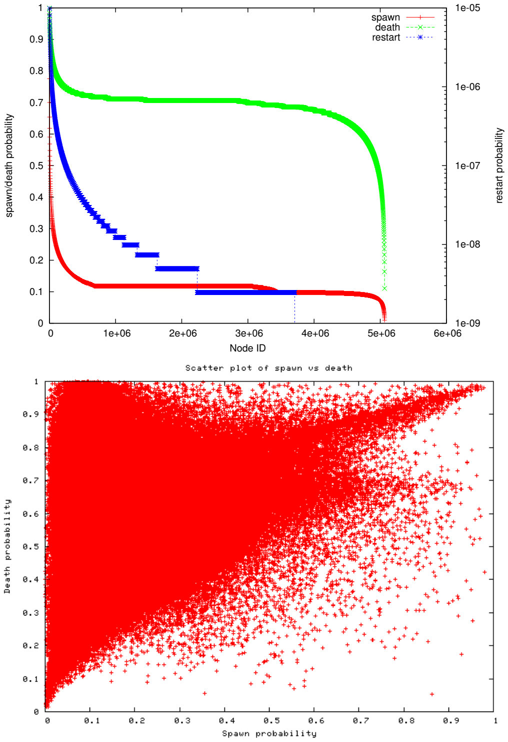

Figure 2: Basic properties of spawn and death probabilities.

Table 1 gives examples of the hosts with particularly low and high spawn and death probabilities. Most of the sites with high death probabilities are spam sites from which users are unlikely to initiate significant browsing activity. The sites with the lowest death probability are compelling online service providers like games, sports news, and social networks that tend to generate long-term engagement.

The hosts with high spawn probabilities are those from which it is common to open many links. Photo gallery sites and photoblogs and well represented here. Hosts with low spawn probabilities represent sites from which exploring many paths from a page is not a common behavior, either because the user is typically looking for a single resource (dictionary.com, for example) or because the user tends to engage heavily with a single page (games.msn.com or cartoonetwork.com, for example).

Host	Spawn	Death
apps.facebook.com	0.5245	0.468
facebook.com	0.6567	0.952
friendster.com	0.5237	0.773
google.com	0.2793	0.339
mail.yahoo.com	0.5271	0.918
search.yahoo.com	0.2851	0.326
tagged.com	0.6244	0.722
viewmorepics.myspace.com	0.4951	0.623
yahoo.com	0.5240	0.539
youtube.com	0.1393	0.318

Table 2: Spawn, death probabilities of top visited sites.

Notice that there are many reasons why a host may have extremal spawn and/or death probabilities. The model is agnostic to the particulars, but will faithfully reproduce the behavior.

Table 2 shows the spawn and death probabilities of ten frequently-visited visited hosts from our sample, in alphabetical order. The social networking sites like Facebook, Friendster, Tagged, and Myspace have higher spawn probabilities than the others, partly because the prevalence of photos, and partly because such sites offer users feeds containing multiple pieces of interesting content. Youtube, on the other hand, has a low spawn probability capturing the intuition that users do not open multiple windows or tabs with simultaneous videos, and prefer to consume a sequence of videos, rather than using a single page as a "hub" from which to explore.

The death probabilities are likewise interesting. The social network sites tend to have higher death probabilities, as does Yahoo! mail, which offers an inbox from which users can visit messages and then click back to the inbox for more.

Note that a site with a high spawn and death probability can be explained by sites that offer users "hub"-style pages that link to destinations that are consumed in-place, such as galleries, lists of products with links to product information for each, and so forth. The upper right quadrant of the scatter plot in Figure 5.3 shows hosts that offer the extreme of this behavior.

5.4 Site-level and traffic distribution

Table 3 gives the aggregate performance numbers for pagerank and tabrank. Results on other time periods are similar. The numbers in the table are the ℓ_1 distance between the model and the actual steady-state distribution over either nodes or edges.

The first finding is that tabrank performs similarly to pagerank, and often slightly worse, when the distribution over outlinks on a page is taken to be uniform. Tabrank is 6% better for uniform restart distribution over nodes, and 1-4% worse for the other cases.

The differences become more apparent in the models that have the greater fidelity. When comparing models that have the correct reset distribution and the correct outlink distribution, tabrank is able to remove 1/3 of the error of pagerank in estimating node steady states, and 70% of the error in estimating edge steady states.

Lowest spawn probabilities	Highest spawn probabilities	Lowest death probabilities	Highest death probabilities
169.70.240	ar.babel.com	acesolitaire.com	149.244.124
169.70.241	atlpics.net	asseenonpc.directtrack.com	168.102.254:1000
adv-adserver.com	carpediemphoto.smugmug.com	asycieniasy.pl	168.254.251:1000
bay.livefilestore.com	chat.tchatche.com	cartoonetwork.com	174.149.70
bearwww.com	fazendopose.multiply.com	cbs.sportsline.com	2.120.6
cartoonetwork.com	fr.babel.com	disneyxd.com	224.57.23
cr.naver.com	inmoeciu.ro	games.msn.com	3839dm.cn
dictionary.com	looklet.com	imediabiz.tv	ads.freedomltd.biz
games.msn.com	meiodomato.com.br	instantcertonline.com	alllikes.cn
goldresults.net	my-gor.com	lebestof.eu	c5.zedo.com
imediabiz.tv	nugaalmedia.com	secure.myembarq.com	heruholsvrshs03
jarmediatrack.com	pctrailruns.com	secure.studivz.net	img2.zamunda.net
maxsun.biz	photo.tiratron.com	securebank.regions.com	new.going.com
moreverde.com	photos.essence.com	submit-tools.com	premiercardoffers.com
recs.richrelevance.com	picasaweb.google.ro	tmobile.com	s.magentic.com
scraps.recadopop.com	playdatephotos.smugmug.com	ui.texasworkforce.org	server2.mediajmp.com
serve.socialcash.com	rallogotpicz.smugmug.com	web.charter.net	skyblueads.com
signup.live.com	simnet.is	ww.myspace.com	thefutoncritic.com
surveys.surveynetwork.com	sublime-stitching.blogspot.com	wwww.myspace.com	thexinxin.cn
te.kontera.com	voyagehotel.com	yourtube.ru	web1699.cn
wzpo1.ask.com		youtube.com.br	

Table 1: Sample sites with the lowest spawn probability.

	Uniform restart distribution				Measured restart distribution			
	Outlinks uniform		Outlinks measured		Outlinks uniform		Outlinks measured	
	PR	TR	PR	TR	PR	TR	PR	TR
Nodes	1.785898	1.691266	1.702787	1.600571	1.339382	1.381173	0.584982	0.386610
Edges	1.334243	1.435901	1.239302	1.190221	1.535682	1.614064	0.956315	0.276970

Table 3: ℓ_1 distance between model steady state and empirical ground truth distribution for 2009/07/18.

Our first observation is that pagerank is more likely to account correctly for node probabilities, as the reset distribution is seeded from these directly. If each node visit resulted in a single edge visit, then pagerank would be equally accurate at modeling edge transition probabilities. However, as we have seen, users follow varying numbers of edges out of a single node based on the types of behavior that tabrank models. Thus, we expect that tabrank will show better performance relative to pagerank in estimating edge probabilities.

With non-uniform restart distribution, tabrank is about 7% more accurate than pagerank at representing both node and edge distributions under non-uniform outlink distributions. This indicates that multiple outlinks have a small but non-negligible contribution to error rates even in the highly approximate uniform model.

Additionally, we revisit the same data over all three datasets. Figure 3 shows the results for non-uniform restart distributions in all three time periods. The results can be seen to be similar, although for earlier time periods the distinction between pagerank and tabrank is not as large. Again, without outdegree distributions, we see pagerank and tabrank behaving almost identically, with pagerank often slightly better.

Finally, we turn our attention to the nature of the distribution of node and edge steady state probabilities.

Figures 4 and 5 show these distributions in the uniform and non-uniform restart distribution cases respectively. The figures show for the actual and modeled distributions the probability of the xth most likely node under that distribution, on a log-log plot. It is possible that the forms of two distributions could be identical but the ids of the high-mass nodes could be entirely different; thus, these charts give a visual impress on of areas of agreement but should be interpreted in light of the more reliable global metric of Table 3. Figure 4 shows that employing a uniform restart distribution underestimates the head and overestimates the tail in all models. Figure 5 shows that models with access to the overall popularity of each host, as encoded in the restart distribution, are able to fit the actual data much more reliably for both nodes and edges. Adding the outdegree distributions yields a much more accurate fit, especially for modeling the probability assigned to each edge.

However, while the pagerank and tabrank curves for non-uniform restart distributions both appear quite accurate, the ℓ_1 distance between pagerank and the actual edge distribution is 3.45 times larger than that of tabrank, indicating that multiple edge traversals are the key “low hanging fruit” in providing a more accurate assessment of edge traversals in a simple browsing model.

5.5 Spawn, death, and spam

Figure 6 follows up on the observation from Section 5.3 that high death probabilities, and to a lesser extent low spawn probabilities, tend to correlate with spam sites. The top figure shows the probability that hosts with a given death rate are spam sites as determined by a spam classifier available at Yahoo! The expected correlation appears: as the death probability transitions from 0.5 to 0.65, the probability of spam increases dramatically.

The story for spawn probabilities is much less clear. If anything, hosts with intermediate spawn scores are more likely to be spam, but the relationship is weak.

This suggests that probability of death is a candidate feature for spam classification.

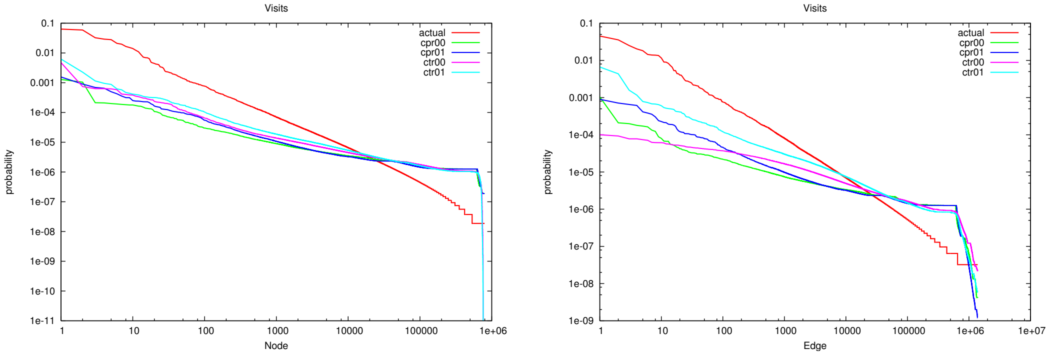

Figure 4: Distribution of pagerank and tabrank, uniform reset distribution, nodes(left) and edges(right).

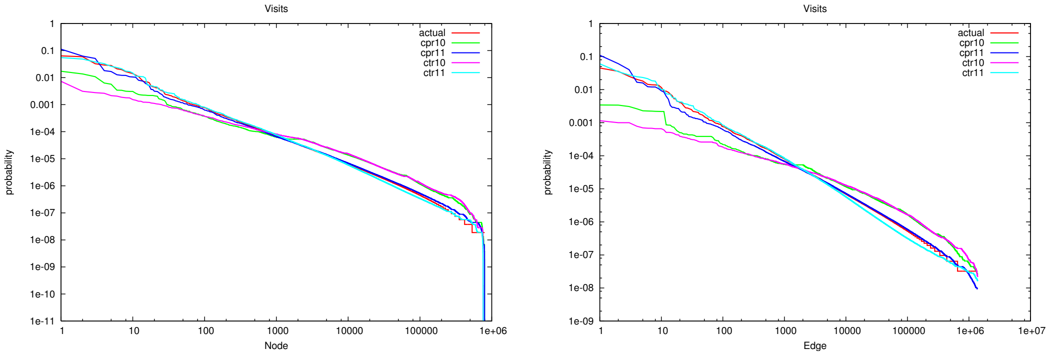

Figure 5: Distribution of pagerank and tabrank, non-uniform reset distribution, nodes(left) and edges(right).

5.6 Non-uniform restart probabilities

The experiments regarding non-uniform restart probabilities are compared against the actual measured distribution, and this distribution is used to build the restart probabilities in the first place. Thus, at least in the case of node steady states probabilities, the correct answer has been "planted" in the restart probabilities, and the naive algorithm that simply jumps with probability 1 to the restart distribution will perfect match the target. In fact for these experiments we do see that the distance from the target distribution increases with the number of iterations.

The situation for edges, however, is quite different: the restart distribution seeds the correct node steady states, but under pagerank, these may be incompatible with the observed edge transition probabilities. This disparity accounts for the significant improvements given by tabrank in Table 3 for non-uniform restart distributions and non-uniform outlink distributions.

6. CONCLUSIONS

We have developed a model for tabbed browsing, in which users may have multiple tabs open simultaneously. We formalize this model as a stochastic process, and analyze the conditions under which this process terminates. In both the terminating and non-terminating conditions, we characterize the steady state distribution of the process. If the process terminates, this steady state depends on the initial conditions; if the process fails to terminate with nonzero probability then the steady state does not depend on the initial conditions.

We then apply our algorithm and the pagerank algorithm to a dataset derived from anonymized Yahoo! toolbar data. We show that the tab process performs identically to a variant of pagerank (as predicted) if all hosts have the same spawn and death probability. Otherwise, tabrank gives a more accurate representation of the steady state as the processes are given more information. In particular, if pagerank and tabrank are given both non-uniform outlink distributions based on data, and a non-uniform restart distribution, then the steady state node distribution under pagerank has more than 1.5 times the error of tabrank, and the steady state edge traversal distribution under pagerank has more than 3.45 times the error of tabrank.

Acknowledgments

We thank the anonymous reviewers for their suggestions.

7. REFERENCES

[1] N. Alon, C. Avin, M. Koucký, G. Kozma, and Z. Lotker. Many random walks are faster than one. In *Proc. 20th SPAA*, pages 119–128, 2008.

[2] K. B. Athreya and P. E. Ney. *Branching Processes.* Dover Publications, Inc., New York, 2004.

[3] R. A. Baeza-Yates, P. Boldi, and C. Castillo. Generic damping functions for propagating importance in link-based ranking. *Internet Mathematics*, 3(4):445–478, 2007.

[4] Z. Bar-Yossef, A. Z. Broder, R. Kumar, and A. Tomkins. Sic transit gloria telae: Towards and understanding of the web's decay. In *Proc. 13th WWW*, pages 328–337, 2004.

[5] A. Benczur, K. Csalogany, T. Sarlos, and M. Uher. Spamrank - fully automatic link spam detection. *Proc. 1st AIRWeb*, pages 1–14, 2005.

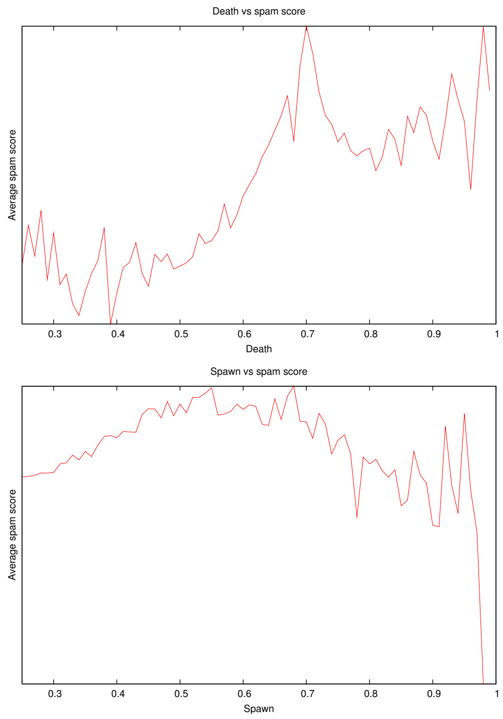

Figure 6: Relationship between the spawn and death probabilities of a host and its spam score.

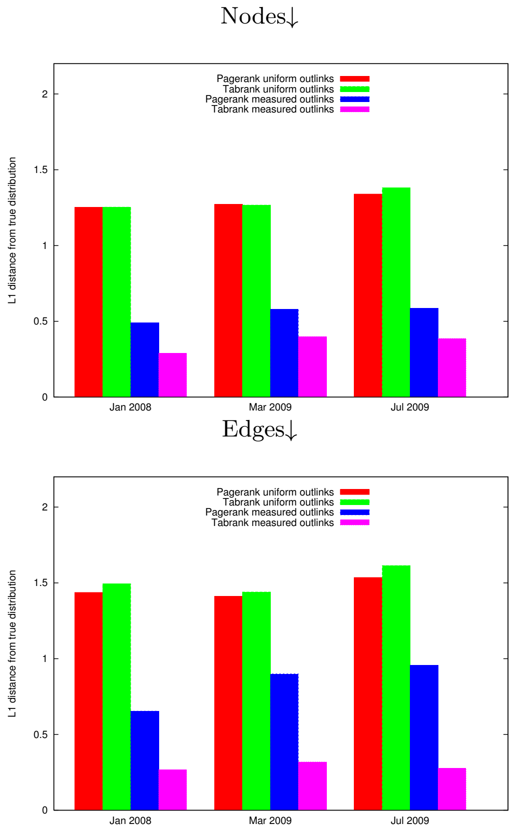

Figure 3: Performance of pagerank and tabrank with non-uniform reset distribution.

[6] P. Berkhin. A survey on pagerank computing. *Internet Mathematics*, 2(1):73–120, 2005.

[7] P. Boldi, M. Santini, and S. Vigna. A deeper investigation of pagerank as a function of the damping factor. In *Web Information Retrieval & Linear Algebra Algorithms*, 2007.

[8] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. T. P. Link analysis ranking: Algorithms, theory, and experiments. *ACM TOIT*, 5:231–297, 2005.

[9] M. Bouklit and F. Mathieu. BackRank: An alternative for PageRank? In *Proc. 14th WWW (Special interest tracks and posters)*, pages 1122–1123, 2005.

[10] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.

[11] K. Efremenko and O. Reingold. How well do random walks parallelize? In *Proc. 12th APPROX-RANDOM*, pages 479–489, 2009.

[12] R. Elsässer and T. Sauerwald. Tight bounds for the cover time of multiple random walks. In *Proc. 36th ICALP*, pages 415–426, 2009.

[13] K. Etessami and M. Yannakakis. Recursive Markov chains: Stochastic grammars, and monotone systems of nonlinear equations. *J. ACM*, 56(1), 2009.

[14] R. Fagin, A. R. Karlin, J. Kleinberg, P. Raghavan, S. Rajagopalan, R. Rubinfeld, M. Sudan, and A. Tomkins. Random walks with ”back buttons”. *Annals of Applied Probability*, 11(3):810–862, 2001.

[15] W. Feller. *An Introduction to Probability Theory and Its Applications, Volume 1*. Wiley, 1968.

[16] B. Gonçalves, M. R. Meiss, J. J. Ramas, A. Flammini, and F. Menczer. Remembering what we like: Toward an agent-based model of web traffic. In *Proc. 2nd WSDM (Late Breaking Results)*, 2009.

[17] Z. Gyongi, H. Garcia-Molina, and J. Pedersen. Combating web spam with TrustRank. In *Proc. 30th VLDB*, pages 576–587, 2004.

[18] T. H. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *TKDE*, 15(4):784–796, 2003.

[19] A. N. Langville and C. D. Meyer. Deeper inside PageRank. *Internet Mathematics*, 1(3):335–380, 2005.

[20] R. Lempel and S. Moran. SALSA: The stochastic approach for link-structure analysis. *ACM TOIS*, 19(2):131–160, 2001.

[21] Y. Liu, B. Gao, T.-Y. Liu, Y. Zhang, Z. Ma, S. He, and H. Li. Browserank: Letting web users vote for page importance. In *Proc. 31st SIGIR*, pages 451–458, 2008.

[22] M. Meiss, J. Duncan, B. Gonçalves, J. J. Ramasco, and F. Menczer. What's in a session: Tracking individual behavior on the web. In *Proc. 20th Hypertext*, pages 173–182, 2009.

[23] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[24] E. Seneta. *Non-negative Matrices and Markov Chains.* Springer, 1973.

[25] M. Sydow. Randoms surfer with back step. In *Proc. 13th WWW (Special interest tracks and posters)*, pages 352–353, 2004.

[26] P. Tsaparas. *Link Analysis Ranking Algorithms.* PhD thesis, University of Toronto, 2003.

[27] M. Viermetz, C. Stolz, V. Gedov, and M. Skubacz. Relevance and impact of tabbed browsing behavior on web usage mining. In *Proc. WI*, pages 262–269, 2006.