Inf Retrieval (2010) 13:22–45
DOI 10.1007/s10791-009-9098-8

A framework to compute page importance based on user behaviors

Yuting Liu · Tie-Yan Liu · Bin Gao · Zhiming Ma · Hang Li

Received: 3 December 2008 / Accepted: 26 May 2009 / Published online: 19 June 2009

Abstract This paper is concerned with a framework to compute the importance of webpages by using real browsing behaviors of Web users. In contrast, many previous approaches like PageRank compute page importance through the use of the hyperlink graph of the Web. Recently, people have realized that the hyperlink graph is incomplete and inaccurate as a data source for determining page importance, and proposed using the real behaviors of Web users instead. In this paper, we propose a formal framework to compute page importance from user behavior data (which covers some previous works as special cases). First, we use a stochastic process to model the browsing behaviors of Web users. According to the analysis on hundreds of millions of real records of user behaviors, we justify that the process is actually a continuous-time time-homogeneous Markov process, and its stationary probability distribution can be used as the measure of page importance. Second, we propose a number of ways to estimate parameters of the stochastic process from real data, which result in a group of algorithms for page importance computation (all referred to as BrowseRank). Our experimental results have shown that the proposed algorithms can outperform the baseline methods such as PageRank and TrustRank in several tasks, demonstrating the advantage of using our proposed framework.

Y. Liu (✉)
School of Science, Beijing Jiaotong University, Beijing, China
e-mail: liuyt_njtu@hotmail.com

T.-Y. Liu · B. Gao · H. Li
Microsoft Research Asia, Beijing, China

T.-Y. Liu
e-mail: tyliu@microsoft.com

B. Gao
e-mail: bingao@microsoft.com

H. Li
e-mail: hangli@microsoft.com

Z. Ma
Academy of Mathematical and Systems Science, CAS, Beijing, China
e-mail: mazm@amt.ac.cn

Keywords User browsing process · Continuous-time time-homogeneous Markov process · Staying time · BrowseRank

1 Introduction

Page importance is a key factor for Web search, because for contemporary search engines, crawling, indexing, and ranking are usually guided by this measure. Conventionally, the link analysis approaches have been used to calculate the page importance from the hyperlink graph of the Web. These approaches take the link from one webpage to another as an endorsement of the linking page, and assume that the more links pointed to a page, the more likely it is important. Two typical examples are PageRank (Brin et al. 1998, Page et al. 1999) and HITS (Kleinberg 1998). The link analysis algorithms have been successfully applied to Web search.

Recently people (Eirinaki and Vazirgiannis 2005, Liu et al. 2008, Oztekin et al. 2003) have realized the limitations of the link analysis approaches. First, the hyperlink graph is an incomplete and inaccurate data source for page importance calculation, because the hyperlinks can be easily added or deleted by the Web content creators. For example, it is a common technique to spam search engines by purposely creating a large number of hyperlinks to the target webpage (e.g., link farm and link exchange) (Gyongyi et al. 2005). Second, there are some unrealistic assumptions on the user behaviors in these link analysis methods. For example, in HITS and PageRank, it is assumed that users select the next page to visit from the outlinks of the current page in a uniformly random manner. For another example, in PageRank, it is assumed that the same period of time is spent on different pages. These are, however, clearly inconsistent with the browsing behaviors of real users.

To tackle these problems with the link analysis methods, a fundamental approach is to leverage the browsing behaviors of real users. This is because it is the users who determine which pages to visit, which hyperlinks to click, and which pages to spend much time on. This is exactly the motivation of this paper and some previous works (Liu et al. 2008). In this work, we propose a formal framework to compute page importance from user behavior data.

First, we use a stochastic process to model the browsing behaviors of Web users, which we call the user browsing process. From the investigation and analysis on hundreds of millions of real records of user behaviors, we find that the real user behaviors can be classified into two categories: reading (or staying on) a webpage or jumping from one page to another. The stochastic process thus contains the staying on a state and the transition from one state to another. We then show that such a process is actually a continuous-time time-homogeneous Markov process, and its stationary probability distribution can be used as the measure of page importance. Furthermore, we prove that this stationary distribution is determined by two factors, the visiting frequency and the length of staying time. In other words, the more visits to a page made by the users and the longer time spent by the users on it, the more likely it is important. This is consistent with our intuition.

Second, we propose a number of ways to estimate the parameters of the stochastic process. The parameter estimation contains two steps. The first step is to estimate the distribution of the staying time, and the second step is to estimate the stationary distribution of an associated embedded Markov chain of the original process, which encodes the transition information. For each step, we adopt various estimators. In total eight different algorithms are developed by combining different estimators in the two steps. Note that the algorithm proposed in Liu et al. 2008 is one of the eight algorithms. In this regard, we say our proposed framework can contain the work in Liu et al. 2008 as its special case.

Then, we conduct comprehensive experiments to study the effectiveness of different estimators, and thus different algorithms we proposed. We also compare the proposed algorithms with baselines like PageRank (Brin et al. 1998, Page et al. 1999) and TrustRank (Gyöngyi et al. 2004). We performed the experiments at both page level and website level. For the page-level experiment, we mainly tested the contributions of different page important algorithms to relevance ranking. For the website-level experiments, we mainly tested the use of page importance in authoritative page finding and web spam filtering. Our experimental results have shown that the proposed algorithms are very effective in these tasks, and significantly outperform the baseline algorithms. We also discussed the following issues based on the detailed analysis on the experimental results: (1) the contributions of the two steps in our proposed framework to the final page important computation; (2) the comparison between user behavior data and the conventional hyperlink graph; (3) the reason why the proposed algorithms outperform the baselines.

The rest of the paper is organized as follows. Section 2 describes the user browsing process. In Section 3, many different ways of estimating the parameters are discussed. Experimental results are reported in Section 4. Section 5 introduces some related works. Conclusion and future work are given in Section 6.

2 User browsing process

In order to precisely model the behaviors of real Web users, we first perform some analysis on the behavior data, and then give a mathematical formulation of the so-called user browsing process. After that we will show that the stationary distribution of this process can be an effective measure of the page importance.

2.1 Motivation

The Web might be the richest information source in the world, and people are fulfilling their information needs by browsing the Web. According to our observations on a large amount of user behavior data, users' browsing behavior can be classified into two types: reading a webpage, or jumping from one page to another.

Reading is an important means for users to get the information contained in a webpage, and it usually takes a certain period of time. When reading, some users may quickly scan the headlines, others may read the content very carefully. As a result, the time spent on the same page by different users can have a large variance.

When a user cannot find the desired information on the current page, or she/he wants to find some other interesting things, she/he may choose to jump to another page by clicking a hyperlink on the current page. She/he can also choose to directly input a new URL in the address bar of the browser.[1] The jump from one page to another by hyperlink click can be regarded as a kind of endorsement to the destination page.[2] The jump by means of direct URL input can be regarded as a kind of preferential reset (as opposite to the random reset in conventional link analysis algorithms).

When the user comes to a new page, she/he will repeat the loop of reading and jumping, until her/his information need is satisfied or she/he gives up.

[1] We also categorize the browsing behaviors of selecting an URL from the bookmark of the browser and typing a query in some search toolbar to jump to the ranking result webpage as a kind of direct input.

[2] This assumption has also been made in many conventional link analysis algorithms, although they are talking about a virtual Web surfer but not real Web users.

Based on the above observations, we can make the following assumptions, which are intuitively easy to understand.

1. **Assumption 1**
   The next page that a user will visit by means of hyperlink click only depends on the current page, regardless of other pages she/he visited previously. This is also a basic assumption in many previous works such as PageRank. However, PageRank further assumes the surfer to choose the next page in a uniformly random manner from the outlinks of the current page, which is not sound according to our observations.
2. **Assumption 2**
   The behavior of a user is independent of the specific time point at which the browsing occurs. For example, the period of time it takes for a user to go through a page is roughly fixed no matter when she/he reads the page. For another example, the probability that a user jumps from one page to another does not change either, no matter when she/he conducts the jump. Note that this is also a basic assumption in previous works like PageRank.

2.2 Mathematical formulation for user browsing process

We use the following stochastic process to describe the browsing behaviors of the Web users. Suppose there is a Web surfer browsing the Web. We use X_s to denote the webpage that the surfer visits at time $s \geq 0$. Then the user browsing process can be represented as a continuous-time discrete-state stochastic process $X = \{X_s, s \geq 0\}$ (Berger 1993), where the state space of X is the set of all webpages (we suppose there are N webpages in total).

Considering the two assumptions that we obtained from the observations on real data, we can attain the following properties of process X.

2.2.1 Markov property

According to Assumption 1 in Section 2.1, the process X is a continuous-time Markov process (Anderson 1991): given a previous state X_s, the transitions occurring afterwards are independent of the states earlier than X_s. Mathematically, for any time series $+\infty > t \geq s \geq u \geq 0$, we have

$$P(X_t = j | X_s = i, X_u = k) = P(X_t = j | X_s = i)$$

We use $P(s,t) \triangleq \left(p_{ij}(s,t)\right)_{N\times N}, t \geq s$ to denote the transition probability matrix from time s to time t of process X, where $p_{ij}(s,t) \triangleq P\{X_t = j | X_s = i\}$ is the probability of transition from page i at time s to page j at time t. We can find that this transition probability is dependent on not only the source and destination states, but also the starting and ending time of the transition.

2.2.2 Time-homogeneity

According to Assumption 2 in Section 2.1, the browsing behavior is independent of the specific time point. In stochastic process, this property is formally referred to as time-homogeneity (Berger 1993). Mathematically, for any time points $+\infty > t \geq s \geq 0$,

$$p_{ij}(s,t) = P\{X_t = j | X_s = i\} = P\{X_{t-s} = j | X_0 = i\} = p_{ij}(0, t-s)$$

The above equation indicates that the transition probability is only determined by the length of the transition period, but not by the starting and ending time of the transition. Hereafter, for simplicity, we use $p_{ij}(t)$ to denote the transition probability from page i to page j with the length of the transition period being equal to t. Thereby, the transition probability matrix can be written as $P(t) = \left(p_{ij}(t)\right)_{N\times N}$.

It is clear that for any t, we have

$$\sum_{j=1}^{N} p_{ij}(t) = 1, \tag{1}$$

which means that after a Web surfer spends a period of time t on page i, she/he can only do the following things: transits to another page $j \neq i$, or keeps staying at page i. In other words, the user browsing process is conservative (Anderson 1991).

Based on the above analysis, the user browsing process X is a *continuous-time time-homogeneous Markov process*. As we have discussed in Section 2.1, there are two types of user browsing behaviors: reading a page and jumping from one page to another. Now let us discuss whether these two types of behaviors can be well described by process X, and whether effective page importance can be computed from this process.

First, in order to relate the properties of process X to the two types of user behaviors, we need to introduce the concept of Q-Process. Actually, it has been a common practice to study a continuous-time time-homogeneous Markov process with its (one-to-one) corresponding Q-Process, when the state space is finite (Wang et al. 1992).

Definition 1 Suppose X is a continuous-time time-homogeneous Markov process as defined above. If it is conservative and its state space is finite, then $p_{ij}(t)$ is differentiable with respect to any time point t. Define $q_{ij} \triangleq p'_{ij}(0)$, and $Q \triangleq (q_{ij})_{N\times N}$, where $p_{ij}{}'(0)$ is the derivative of $p_{ij}(t)$ at $t = 0$. The matrix $(q_{ij})_{N\times N}$ is called a Q-matrix.

The Q-matrix is also referred to as the transition rate matrix, because $q_{ij}, i \neq j$ can be explained as the rate at which process X enters state j from state i, and $-q_{ii}$ as the rate at which process X leaves state i. Considering that the process must enter a state (denoted as j) when it leaves state i, we have $-q_{ii} = \sum_{j\neq i} q_{ij}$. This also justifies that process X is conservative.

Owing to the conservativity of the process, it is easy to prove that $-\infty < q_{ii} < 0$, $0 < q_{ij} < \infty$, and $\sum_j q_{ij} = 0$. That is, in the Q-matrix, the diagonal elements are negative; the non-diagonal elements are positive; the sum of all elements in each row is zero. Furthermore, according to the one-to-one correspondence between Q and $P(t)$, the original process X is sometimes referred to as a Q-Process.

Based on the definition of Q-matrix, it has been proved that process X satisfies the following two properties (Wang et al. 1992).

Property 1 *Suppose X is a Q-process. Then for any state i and any time interval r > 0, we have*

$$P(X_s = i, s \preceq r | X_0 = i) = 1 - e^{q_{ii}r} \tag{2}$$

This property indicates that the staying time on page i is governed by an exponential distribution with parameter $-q_{ii}$, where $-q_{ii}$ denotes the leaving rate of page i according to the definition of Q-Process. Due to this distribution, it is easy to obtain that the mean staying time on page i is $-\frac{1}{q_{ii}}$. Therefore, the larger $-q_{ii}$ is, the shorter the staying time on

page i is. Note that this property distinguishes the Q-Process from a discrete-time Markov process, in which the length of the staying time is a constant for all the states.

Property 2 *Suppose X is a Q-process. Then for any two states i, j and* $i \neq j$,

$$P(X_s = j, X_t = i, \forall t \leq s) | X_0 = i) = -\frac{q_{ij}}{q_{ii}} \tag{3}$$

This property indicates that the probability of the transition from state i to state j directly is determined solely by these two states, regardless of the staying time on page i.

Based on the above discussions, we can see that it is quite reasonable to define the user browsing process as a continuous-time time-homogeneous Markov process. First, the Q-Process can be decomposed into two parts, the staying on a state and the transition from one state to another, which just correspond to the two types of user behaviors. Second, in the Q-Process, the length of the staying time is governed by an exponential distribution, whose parameter is only determined by the leaving rate of the page (i.e., $-q_{ii}$). Accordingly, the period of time that a real user spends on a page is mainly determined by the page itself, e.g., by the size and the content of the page. Third, in the Q-Process, the transition is independent of the staying time. Also, when a user wants to jump from the current page to another page, the choice of the target page is usually determined by the importance of the pages, but irrelevant to the length of the staying time on the current page.

2.3 Page importance computation based on user browsing process

In this subsection, we discuss how to compute page importance based on the aforementioned user browsing process.

At first glance, it is easy to perform the task due to the following reason. As we know, for a continuous-time time-homogeneous Markov process $X = \{X_s, s \geq 0\}$, if it is irreducible[3] and aperiodic,[4] there exists a unique stationary probability distribution $\pi = (\pi_1, \pi_2, ..., \pi_N)$ (Berger 1993), which is independent of time interval t:

$$\begin{cases} \pi = \pi P(t) \\ \sum_i \pi_i = 1 \end{cases} \tag{4}$$

The ith element π_i stands for the ratio of the time spent on page i over the whole period of time spent on all pages when t approaches infinity. In other words, $\pi_i = \lim_{t \to \infty} \frac{\textit{time spent on page } i}{t}$. In this regard, π reflects the importance of a page, and we can use it as a measure of page importance.

However, in practice it is very difficult to directly compute this distribution π, because all elements in the transition probability matrix $P(t)$ are functions of variable t. That is, one needs to estimate these time-dependent functions in order to compute the distribution π.

One feasible (and widely-used) solution to the above challenge is to leverage the correspondence between $P(t)$ and matrix Q. In this solution, the embedded Markov chain of the Q-matrix, defined as below (Stewart 1994), is used.

[3] See Section 3.3 for detailed proof.

[4] It is usually assumed that the user browsing process is aperiodic in many related works (Eirinaki and Vazirgiannis 2005, Langville et al. 2004, Oztekin et al. 2003), we also regard it as true.

Definition 2 Suppose X is a Q-Process with transition matrix $P(t)$. Then a discrete-time Markov chain $Y = \{Y_n, n \geq 0\}$ with transition matrix $\tilde{P}$ is called an embedded Markov chain (EMC) of X, if process Y has the same state space as process X, and satisfies the following condition:

$$\tilde{p}_{ij} = \begin{cases} -\frac{q_{ij}}{q_{ii}}, & i \neq j \\ 0, & i = j \end{cases} \tag{5}$$

where q_{ij}, $i, j = 1,...,N$ are the elements in the Q-matrix.

By comparing Eq. 5 with 3, we find that process Y just records the direct transition information in process X. Then the stationary probability distribution of process Y, denoted as $\tilde{\pi}$, corresponds to the mean visiting frequencies of all the webpages.

With the help of EMC, we can compute the page importance π in the following way.

Proposition 1 *Suppose X is an irreducible Q-process, and Y is the EMC derived from its Q-matrix. Let $\pi = (\pi_1,...,\pi_N)$ and $\tilde{\pi} = (\tilde{\pi}_1, \ldots, \tilde{\pi}_N)$ denote the stationary probability distributions of processes X and Y respectively. Then we have*

$$\pi_i = \frac{\frac{\tilde{\pi}_i}{q_{ii}}}{\sum_{j=1}^{N} \frac{\tilde{\pi}_j}{q_{jj}}} \tag{6}$$

The above proposition shows that we obtain the stationary probability distribution π of process X by normalizing the product of $\tilde{\pi}$ and the reciprocal of the diagonal elements of matrix Q. More importantly, $\tilde{\pi}$ and the Q-matrix are much easier to estimate than the original transition probability matrix $P(t)$.

As discussed before, $\tilde{\pi}$ corresponds to the mean visiting frequencies of all the webpages, and $-\frac{1}{q_{ii}}$ encodes the mean staying time on webpage i. Therefore, we can come to the conclusion that the page importance π is determined by two factors: visiting frequency and staying time. The larger the visiting frequency of a page is, the more important it is. Similarly, the longer the staying time on a page is, the more important it is. Therefore, in order to compute effective page importance, it turns out that we should develop effective methods to estimate $\tilde{\pi}$ and $-\frac{1}{q_{ii}}$ from the user behavior data.

3 Parameter estimation

In this section we discuss how to estimate the parameters of the user browsing process, so as to effectively compute page importance.

As analyzed in the previous section, parameter q_{ii} is related to the length of the staying time spent by users on page i, and parameter $\tilde{\pi}_i$ evaluates the visiting frequency of page i by users. In order to estimate these two quantities, we need to extract the corresponding information from the user behavior data. After that, some estimation models can be applied.

3.1 Information extraction from user behavior data

Many Web service applications can log the user browsing behaviors under agreements with the users, at the same time of assisting users in their accesses to the Web. Usually, the Web service log takes the following form. First a user ID is used to represent the user. The user ID is randomly created, and does not contain any privacy information of the user. Second, all the behaviors of the user in a period of time are stored as records, each represented by a

Table 1 User behavior records

User ID (after removing all privacy information)		
http://aaa.bbb.com/	2009-01-01, 21:33:05	INPUT
http://aaa.bbb.com/1.htm	2009-01-01, 21:34:11	CLICK
http://ccc.ddd.org/index.htm	2009-01-01, 21:34:52	INPUT
http://eee.fff.edu/	2009-01-01, 21:39:03	CLICK
–	–	–
http://eee.fff.edu/	2009-01-01, 22:00:45	CLICK

triple of <URL, TIME, TYPE> and sorted in the chronological order (see Table 1 for examples). Here, URL denotes the URL of the webpage visited by the user, TIME denotes the time stamp of the visit, and TYPE indicates whether the visit is by a direct URL input (INPUT) or by a hyperlink click (CLICK).

Since the amount of the user behavior data collected by Web service application is usually very large, one needs to perform some pre-processing to extract useful information from it. We list some important pre-processing in Table 2. Since some of them have been explained in (Liu et al. 2008) in detail, we just highlight the differences here. We actually modify the session segmentation and staying time extraction, and add a new step named counting. The reason for introducing the counting process is as follows. For each session, the last URL corresponds to the page whose hyperlinks will not be clicked by the user any more. In other words, the user will perform preferential reset on such a page. The resetting number may be useful for the estimation of the visiting frequency.

By aggregating the information extracted from the records, we are able to build a user browsing graph. Each vertex in the graph represents a URL, associated with a visiting number, a resetting number, a preferential reset probability, and a set of observations of staying time as its metadata. Each directed edge in the graph represents the transition between

Table 2 Pre-processing the user behavior data

Step no.	Step name	Process description
1	Session segmentation	If the type of a record is 'INPUT', then we will regard the current record as the start point of a new session. Then in the same session, all webpages are visited by means of hyperlink click
2	Reset probability estimation	Same as in (Liu et al. 2008)
3	Counting	We count the number of sessions whose last page is a given URL to get the resetting numbers of the corresponding webpage. In addition, we also count the total number of occurrences of a URL in allthe sessions as its visiting number
4	URL pair construction	Same as in (Liu et al. 2008)
5	Staying time extraction	For each URL pair, we regard the difference between the time stamp of the second page and that of the first page as the observed staying time on the first page. And for the last page in a session, we use the difference between the time of the last page and that of the first page in the next session as its observed staying time. If this difference is larger than 30 min (White et al. 2007), we randomly sample a value to be the observed staying time from the distribution of the observed staying time obtained from all pages in the entire data collection

two vertices along hyperlink, associated with the number of transitions as its weight. In other words, the user browsing graph is a weighted directed graph with vertices containing metadata and edges containing weights. We denoted it as $G = , W, C, R, Z, \gamma >$, where $V = \{i\}, W = \{w_{ij}\}, C = \{c_i\}, R = \{r_i\}, Z = \{Z_i | Z_i = \{z_i^1, ..., z_i^{m_i}\}\}, \gamma = \{\gamma_i\}, (i,j = 1, \ldots, N)$ denote the vertices, the numbers of transitions, the visiting numbers, the resetting numbers, the observations of staying time, and the preferential reset probabilities respectively. We still use N to denote the total number of webpages in the user browsing graph, and m_i is the number of observations of staying time for page i.

3.2 Estimation of q_{ii}

According to Property 2 in Section 2, the length of staying time T_i of a Q-Process is theoretically governed by an exponential distribution parameterized by $-q_{ii}$. When given a large number of observations $z_i^1, \ldots, z_i^{m_i}$ of the staying time on page i, the task is to estimate the unknown parameter q_{ii} of the random variable T_i based on these observations. We can choose to perform the task by using many methods like maximum likelihood estimation (MLE). According to our experiments, however, the estimation is non-trivial. This is because the observed staying time do not strictly follow the exponential distribution.

Here is an illustration. For each page in the user behavior data, we have a number of observations of the staying time. Some pages may have a large number of observations, while some other pages may have just a few, depending on the number of visits by users. We randomly select a webpage from those pages with sufficient observations (i.e., with more than one million observations of the staying time), and plot the distribution of these observations in a log-linear coordinate (See Fig. 1a[5]).

From the figure, we can see that the curve does not correspond to an exact exponential distribution, because the log-linear plot of an exponential distribution should be a straight line. The major difference is that there are significantly smaller numbers of short staying time in the observations. As for this phenomenon, we hypothesize that the observations of staying time from the user behavior data are noisy, due to the following reasons. (1) The speed of Internet connection, the length of the page, the layout of the page, and other factors will affect the staying time. (2) The recorded staying time will be longer than the actual staying time, if the user does some other things (e.g., answers a phone call) but leaves the browser window open.

Therefore, it would be necessary to consider the noise in the parameter estimation. To better demonstrate this necessity, we investigate two models to estimate q_{ii}. In the first model, we ignore the noise, and directly use the classical MLE method for the estimation. In the second model, we use a de-noise method which cleans the observed samples before estimation.

3.2.1 MLE model

Maximum likelihood estimation is a popular statistical method for parameter estimation.

From the user browsing graph G, we collect m_i observations $\{z_i^1, \ldots, z_i^{m_i}\}$ of the random variable T_i, and according to Eq. 2, we get the corresponding logarithmic likelihood function as follows,

[5] We only plot the curve for [0, 100] seconds in order to get a clear view.

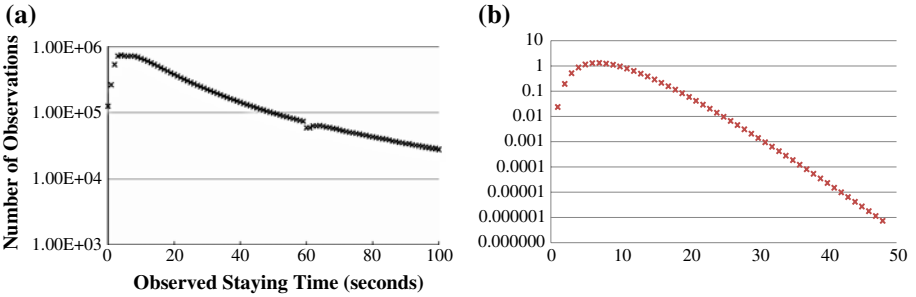

Fig. 1 Figures of two distributions **a** is the plot of real observed staying time, and **b** is the plot of a synthetical staying time with $k = 4$ and $q_{ii} = -1.5$

$$L(q_{ii}) = \sum_{l=1}^{m_i} (log(-q_{ii}) - q_{ii} z_i^l) \tag{7}$$

Then the maximum likelihood estimator of q_{ii} is computed through the following optimization.

$$\begin{gathered} \hat{q}_{ii} = \arg\max_{q_{ii}} L(q_{ii}) = -\frac{1}{\frac{1}{m_i}\sum_{l=1}^{m_i} z_i^l} = -\frac{1}{\bar{Z}_i} \\ \Downarrow \\ -\frac{1}{\hat{q}_{ii}} = \bar{Z}_i \end{gathered} \tag{8}$$

Here, $\bar{Z}_i = \frac{1}{m_i}\sum_{l=1}^{m_i} z_i^l$ is the *sample mean* of variable T_i. That is, we use the sample mean to approximate the real expectation of T_i which is equal to $-\frac{1}{q_{ii}}$.[6]

3.2.2 Additive noise model

Considering that the observed staying time is noisy, an additive noise model (Liu et al. 2008) can be used to represent the observations and to conduct an unbiased and consistent estimation of parameter q_{ii}.

We regard the observed staying time Z_i as being composed of the real staying time T_i and the noise U.

$$Z_i = U + T_i. \tag{9}$$

Here T_i is governed by an exponential distribution, and U is governed by a Chi-square distribution with k degrees of freedom, denoted as $Chi(k)$. The reasons of using the Chi-square distribution are as follows. First, the Chi-square distribution is supported within $[0, +\infty]$, and has been widely used to model the noise with positive value (Lowry 2000). Second, we plot the curve for the combination of the Chi-square distribution with $k = 4$ and the exponential distribution with $q_{ii} = -1.5$ in Fig. 1b. This figure can very well reproduce the observed data distribution in Fig. 1a, indicating that the model is appropriate.

[6] This is a property of the exponential distribution.

With the additive noise model, we estimate the staying time as below. Let the mean and variance of Z_i be μ_i and σ_i^2 respectively. According to the Chi-square distribution, the mean and variance of variable U are k and $2k$. According to Property 2, the mean and variance of variable T_i are $-\frac{1}{q_{ii}}$ and $\frac{1}{q_{ii}^2}$. Owing to the independence between U and T_i, we have (Rice 1995)

$$\begin{aligned} \mu_i &\triangleq E(Z_i) = E(U + T_i) = k - \frac{1}{q_{ii}} \\ \sigma_i^2 &\triangleq Var(Z_i) = Var(U + T_i) = 2k + \frac{1}{q_{ii}^2} \end{aligned} \tag{10}$$

If we can estimate the parameters μ_i and σ_i^2, there are many ways to compute the parameter q_{ii}, such as solving the set of Eqs. 10 directly. Actually, the estimation of μ_i and σ_i^2 can be straightforward, since the *sample mean* $\bar{Z}_i$ and *sample variance* S_i^2 are just unbiased and consistent estimators for μ_i and σ_i^2 (Sorenson 1980). Given m_i observations on Z_i denoted as $z_i^1, z_i^2, \ldots, z_i^{m_i}$, we have

$$\begin{aligned} \bar{Z}_i &= \frac{1}{m_i} \sum_{l=1}^{m_i} z_i^l \\ S_i^2 &= \frac{1}{m_i - 1} \sum_{l=1}^{m_i} (z_i^l - \bar{Z}_i)^2 \end{aligned} \tag{11}$$

As mentioned above, one can solve the group of Eqs. 10 so as to get the estimates of parameters q_{ii} and k as follows:

$$\begin{cases} \bar{Z}_i = k - \frac{1}{q_{ii}} \\ S_i^2 = 2k + \frac{1}{q_{ii}^2} \end{cases} \Rightarrow \begin{cases} q_{ii} = \frac{1}{-1 \pm \sqrt{S_i^2 - 2\bar{Z}_i + 1}}, & q_{ii} < 0 \\ k = (\bar{Z}_i - 1) \pm \sqrt{S_i^2 - 2\bar{Z}_i + 1}, & k > 0 \end{cases} \tag{12}$$

In practice, however, due to data sparsity, the observed samples do not necessarily satisfy the above equations in a strict manner. To tackle this challenge, one can relax the equation group 12 to the following optimization problem, in which the objective function represents the difference between parameter k calculated from the two equations in 12,

$$\begin{aligned} &\min_{q_{ii}} \left(\left(\bar{Z}_i + \frac{1}{q_{ii}} \right) - \frac{1}{2} \left(S_i^2 - \frac{1}{q_{ii}^2} \right) \right)^2 \\ &s.t. \qquad\qquad q_{ii} < 0 \end{aligned} \tag{13}$$

Optimize problem (13) can be solved by many optimization methods such as gradient descent (Boyd et al. 2003). In this way, we can get the estimate of q_{ii} efficiently.

3.3 Estimation of $\tilde{\pi}$

There are two ways of estimating the stationary distribution of a discrete-time Markov process. First, the stationary distribution of a discrete-time Markov process characterizes the mean visiting time of each state among that of all states when the time approaches infinity. Therefore we can *directly* estimate this distribution based on the visiting frequency. Second, as we know, the stationary distribution $\tilde{\pi}$ of a discrete-time Markov process is the principal eigenvector of its transition matrix. Therefore we also can estimate all the elements in the transition matrix first and then compute the distribution by the power method (Golub et al. 1996). We call the second approach an *indirect* approach.

In this subsection, we will introduce four models to estimate $\tilde{\pi}$ of the EMC. The first model belongs to the direct approach, and the other three models belong to the indirect approach.

3.3.1 Direct model

In the user browsing graph G, the set C contains the number of visits of each vertex. By the law of large number (Rice 1995), we can take the visiting frequency as a cursory estimator for $\tilde{\pi}$.

$$\tilde{\pi}_i = \frac{c_i}{\sum_k c_k} \tag{14}$$

3.3.2 Indirect model 1

From the user browsing graph, we can obtain the real transition information between any two pages. Then, based on the law of large number, we can use the frequency of the transitions between two pages i and j to estimate the corresponding transition probability $\tilde{p}_{ij}$.

$$\tilde{p}_{ij} \approx \frac{w_{ij}}{\sum_k w_{ik}}$$

To ensure that the EMC Y is irreducible (i.e., the transition probability matrix $\tilde{P}$ is primitive), we adopt the same smoothing trick as in the PageRank algorithm. Accordingly, the transition probability can be updated as follows,

$$\tilde{p}_{ij} = \begin{cases} \alpha \frac{w_{ij}}{\sum_k w_{ik}} + (1-\alpha)\frac{1}{N} & \sum_k w_{ik} \neq 0 \\ \frac{1}{N} & \sum_k w_{ik} = 0. \end{cases} \tag{15}$$

With the smoothing,[7] process Y will have a unique stationary probability distribution, and one can safely use power method to calculate it.

3.3.3 Indirect model 2

In Indirect Model 1, we adopt the same smoothing method as that in the PageRank algorithm, which assumes that users randomly choose any webpage to restart when they do not want to browse along hyperlinks. Actually, in Section 3.1 we have discussed that the preferential reset probability in the user browsing graph can better describe the real users' behaviors than the uniform reset probability. This encourages us to use the preferential reset probability to replace the uniform reset probability in Indirect Model 1. Accordingly, we obtain the following new estimator for the transition probability,

$$\tilde{p}_{ij} = \begin{cases} \alpha \frac{w_{ij}}{\sum_k w_{ik}} + (1-\alpha)\gamma_j & \sum_k w_{ik} \neq 0 \\ \gamma_j & \sum_k w_{ik} = 0. \end{cases} \tag{16}$$

One may have the concern whether process Y is still irreducible with the use of the preferential reset probability. The answer is positive, and the proof can be found in

[7] In our experiments, we set $\alpha = 0.85$.

Theorem 1. Because of this, we can still safely apply the power method to calculate the corresponding stationary distribution $\tilde{\pi}$.

3.3.4 Indirect model 3

Indirect Models 1 and 2 estimate the transition based on w_{ij}. In this way, only the number of transitions by clicking hyperlinks is considered, while the number of resets is not included. This may lead to some inaccurate estimation, as shown below.

Suppose we have the following observations for page i: it has been visited for 100 times, among which users only transit once from it to page j, and once to page l. For all the other 98 times, users conduct resets (i.e., users terminate their browsing sessions at page i and start new sessions). If we only use the observed transitions for the estimation, without smoothing, we will get $\tilde{p}_{ij} = \frac{1}{2}$, and $\tilde{p}_{il} = \frac{1}{2}$. This indicates that after the user visits page i, she/he must visit either page j or page l, without any other choices. This is significantly different from the real data. Note that this issue cannot be resolved by the smoothing factor, because there are actually 98% users conducting resets, while the smoothing factor can at most introduce a probability of $1 - \alpha$ to the reset.

To tackle the problem, we propose using two quantities, $\frac{\sum_k w_{ik}}{c_i}$ and $\frac{r_i}{c_i}$, in the estimation of the transition probability. The first quantity is the frequency of real transitions from page i to other pages by hyperlink click, and the second one is the frequency of performing reset at page i. Furthermore, we assume that when performing reset, users will follow the preferential reset probability. As a result, we can get the estimator for the transition probabilities as below.

$$\begin{aligned}\tilde{p}_{ij} &= \alpha\left(\frac{\sum_k w_{ik}}{c_i} \times \frac{w_{ik}}{\sum_k w_{ik}} + \frac{r_i}{c_i} \times \gamma_j\right) + (1-\alpha)\gamma_j \\ &= \alpha\frac{w_{ik} + r_i\gamma_j}{c_i} + (1-\alpha)\gamma_j\end{aligned} \tag{17}$$

It can be proved that with Eq. 17, the corresponding EMC is irreducible (see Theorem 1), and thus a unique stationary probability distribution π exists. Therefore, one can safely use the power method (Golub et al. 1996) to calculate it in an efficient manner.

Theorem 1 *Suppose X is a Q-process and Y is its EMC. If the entries in the transition probability matrix $\tilde{P} = (\tilde{p}_{ij})$ of Y are defined as in Eq. 16 or 17, then such matrix is primitive, accordingly, process Y and X are irreducible.*

Proof Please refer to the proof in Liu et al. 2008.

3.4 BrowseRank algorithms

With the aforementioned models for parameter estimations, we can construct eight algorithms by different combinations of the estimators. For ease of reference, we call all of these algorithms BrowseRank. To differentiate them, we use the notation BR(x, y) to denote the BrowseRank algorithm with model x to estimate q_{ii} and model y to estimate $\tilde{\pi}$. We summarize these algorithms in Table 3. It is easy to find that BR(A,InD3) is exactly the algorithm proposed in Liu et al. 2008. In this sense, we say that our proposed framework can cover the work in Liu et al. 2008 as its special case.

Table 3 BrowseRank algorithms

Algorithm name	q_{ii} Estimation model	$\tilde{\pi}$ Estimation model
BR(M,D)	MLE model	Direct model
BR(M,InD1)	MLE model	Indirect model 1
BR(M,InD2)	MLE model	Indirect model 2
BR(M,InD3)	MLE model	Indirect model 3
BR(A,D)	Additive noise model	Direct model
BR(A,InD1)	Additive noise model	Indirect model 1
BR(A,InD2)	Additive noise model	Indirect model 2
BR(A,InD3)	Additive noise model	Indirect model 3

4 Experimental results

We conducted two types of experiments to verify the effectiveness of the BrowseRank algorithms and thus the proposed framework. The first type of experiment was conducted at the webpage-level, in order to test the contribution of an algorithm to relevance ranking. The second type was conducted at the website-level, to test the performance of an algorithm in finding important websites and depressing spam sites.

4.1 Webpage-level experiments

4.1.1 Ranking in web search

In Web search engines, the retrieved webpages for a given query are often ranked based on two factors: content relevance and page importance. A linear aggregation of the ranked lists given by these two factors can be used to generate the final ranked list (Baeza-Yates et al. 1999):

$$\theta \times rank_{relevance} + (1 - \theta) \times rank_{importance}$$

where $0 \leq \theta \leq 1$ is the combination coefficient.

We used this method in our experiments, and used BM25 (Robertson 1997) as the relevance model for ranking.

4.1.2 Dataset and baselines

We used a user behavior dataset, obtained from a commercial search engine. All possible privacy information was rigorously filtered out and the data was pre-processed according to the steps listed in Section 3.1. There are in total over three-billion records, and among them there are about one-billion unique URLs. We also obtained a hyperlink graph containing these one-billion webpages from the same commercial search engine, and computed baselines based on it.

In addition, we obtained a dataset for relevance ranking also from the search engine. The dataset contains 7500 queries and their associated webpages. The queries were randomly sampled from the query log, and the pages were randomly selected from the top 1000 results in the click-through log for each query. The BM25 scores of these webpages were also provided (when counting the inverse document frequency, the

entire index of the search engines was used). For each query-page pair, three labelers were asked to independently judge whether the page is relevant to the query. Then these judgments were aggregated by simple voting to determine the final relevance of the page to the query.

We implemented two baselines for comparison. One is **PageRank**, and the other is **UPR**, which runs the PageRank algorithm on the user behavior data as described in Eirinaki et al. 2005. The first baseline was selected to show the effectiveness of the user behavior data, and the second was used to show the effectiveness of the proposed models.

4.1.3 Evaluation measures

We used three evaluation measures in our experiments, Precision (Baeza-Yates et al. 1999), Mean Average Precision (MAP) (Baeza-Yates et al. 1999), and Normalized Discount Cumulative Gain (NDCG) (Järvelin and Kekäläinen 2000, 2002). Their definitions are as follows.

Suppose the query collection is S, and $|S|$ means the total number of queries. For some query $q \in S$, there are N_q webpages associated with it. Among these webpages, N_{rel} are relevant, and N_{irrel} are irrelevant.

4.1.3.1 Precision

$$Precision@n = \frac{N_{rel}(n)}{n}, \quad n = 1, 2, \ldots, N_q$$

where $N_{rel}(n)$ denotes the number of relevant webpages ranked in the top n positions for a query.

4.1.3.2 MAP

$$MAP = \frac{\sum_{q \in S} AvgP_q}{|S|}$$

$$AvgP_q = \sum_{n=1}^{N_q} \frac{Precision@n \times pos(n)}{N_q}$$

where $pos(n)$ is an indicator. If the webpage ranked at position n is relevant, then $pos(n) = 1$; otherwise, $pos(n) = 0$.

4.1.3.3 NDCG There are four steps to compute NDCG for a ranked list of webpages: (1) Compute the gain of each webpage; (2) Discount the gain of each webpage by its position; (3) Cumulate the discounted gain of the list; (4) Normalize the discounted cumulative gain of the list.

Specifically in our experiments, we compute the NDCG value at position n as follows,

$$NDCG@n = M_n \sum_{i=1}^{n} \frac{2^{r(i)} - 1}{\log_2(1 + i)}, \quad n = 1, 2, \ldots, N_q$$

where $r(i)$ is the rating of the ith page in the list (one for irrelevant pages, three for relevant pages), and the normalization constant M_n is chosen so that the perfect list gets a NDCG score of 1.

4.1.4 Results and discussions

We compared the performances on relevance ranking for different BrowseRank algorithms, and also for PageRank and UPR. The experimental results are shown as below.

4.1.4.1 Comparison among different BrowseRank algorithms The experimental results about the comparison among the eight BrowseRank algorithms are presented in Figs. 2, 3, and 4.

In these figures, when $\theta = 1$, the page importance will not affect the final ranking result, and therefore all the algorithms have the same performance. For other values of θ, we can see that BR(A,InD3) has the best performance in terms of all the evaluation measures. In addition, we also find that the comparisons among different BrowseRank algorithms are robust to various measures. Roughly speaking, their performances can be ordered as follows:

$$\begin{aligned} &BR(A, InD3) > BR(A, InD2) \geq BR(A, D) \\ &> BR(M, InD3) > BR(M, InD2) > BR(M, D) \\ &> BR(A, InD1) > BR(M, InD1) \end{aligned} \tag{18}$$

Based on these results, we can come to the following conclusions.

1. When comparing BR(A,y) with BR(M,y) for any $y \in \{D, InD1, InD2, InD3\}$, we can find that BR(A,y) always outperforms BR(M,y). This seems to indicate that the additive noise model outperforms the MLE model in the estimation of the distribution of staying time (i.e., parameter q_{ii}). This validates our discussions in Section 3.2, about the necessity of removing the noises in the estimation process.
2. From the comparison between Direct Model and the Indirect Models, we can obtain that Indirect Model 3 > Indirect Model 2 > Direct Model > Indirect Model 1. In other words, Indirect Model 3 has the best performance, indicating the benefit of using the resetting number and the preferential reset probability in the estimation of $\tilde{P}$. We also find that the performance of Direct Model is even better than Indirect Model 1. This somehow implies that the visiting frequency is more effective than the PageRank algorithm when transition probability matrix is not accurately estimated.
3. Note that BR(M,D) actually ranks pages according to the ratio of the time spent on each page over those on all the pages. According to the definition of the stationary

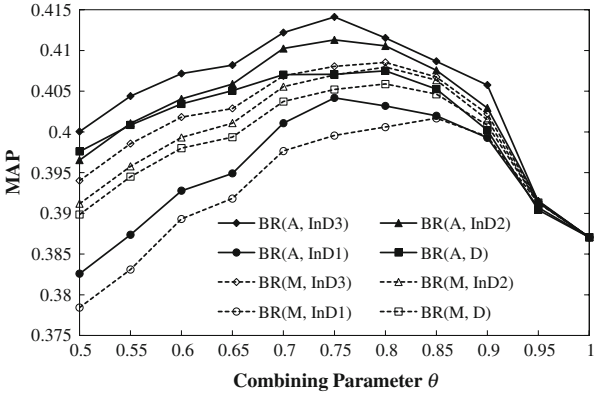

Fig. 2 Search performance in terms of MAP for BrowseRank algorithms

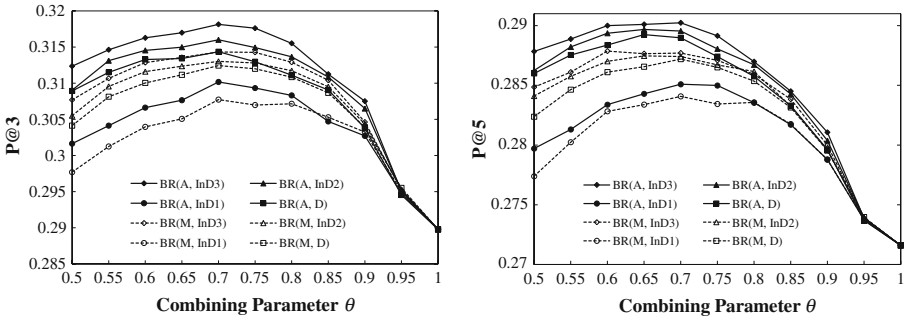

Fig. 3 Search performance in terms of P@3, 5 for BrowseRank algorithms

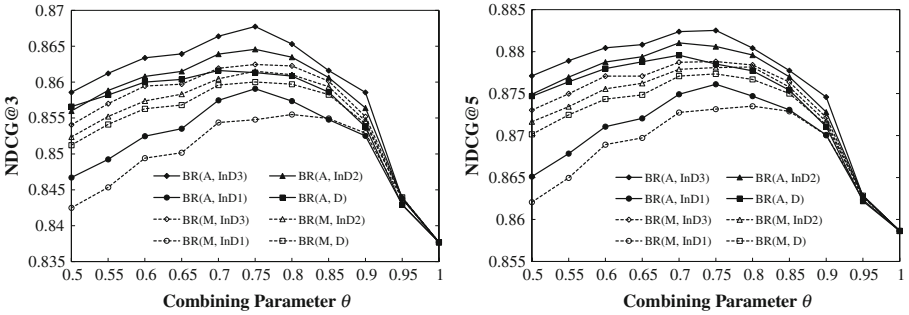

Fig. 4 Search performance in terms of NDCG@3, 5 for BrowseRank algorithms

distribution π of process X (see Section 2.3), π is just the expectation of the ratio given by BR(M,D). In this regard, BR(M,D) can be viewed as a reference estimator for π. As compared to this reference, we can find that BR(A,InD3), BR(A,InD2), BR(A,D), BR(M,InD3), and BR(M,InD2) are better estimators for page importance, while BR(A,InD1) and BR(M,InD1) are worse estimators.

4.1.4.2 Comparison between BrowseRank and baselines Based on the above discussions, we find that BR(A,InD3) is the best among the eight BrowseRank algorithms, and BR(M,D) is the reference to distinguish different estimators. Therefore, in the following experiments, we just use them as the representatives of the BrowseRank algorithms in the comparisons with the baselines. The corresponding results are presented in Figs. 5, 6, and 7.

From the figures, we can see that with the help of page importance, the performance of relevance ranking can be improved. Furthermore, the two BrowseRank algorithms consistently outperform PageRank and UPR in all settings and in terms of all evaluation measures. And the UPR algorithm has better performance than PageRank, which agrees with the observations in Eirinaki et al. 2005.

We also conducted *t*-tests at a confidence level of 95% for the above observations. In terms of MAP, the improvement of BR(A,InD3) over PageRank is statistically significant with a *p*-value of 0.00218. In terms of P@3, P@5, NDCG@3, and NDCG@5, the improvements are also statistically significant with *p*-values of 0.00026, 0.0123, 7.37×10^{-8} and 3.11×10^{-6}, respectively. In terms of MAP, P@3, NDCG@3, and

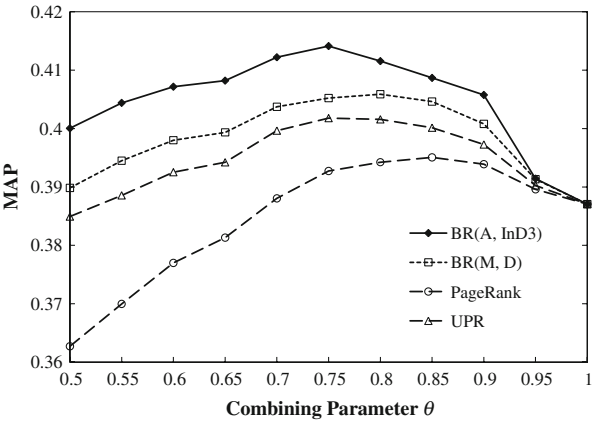

Fig. 5 Search performance in terms of MAP for BrowseRank algorithms and baselines

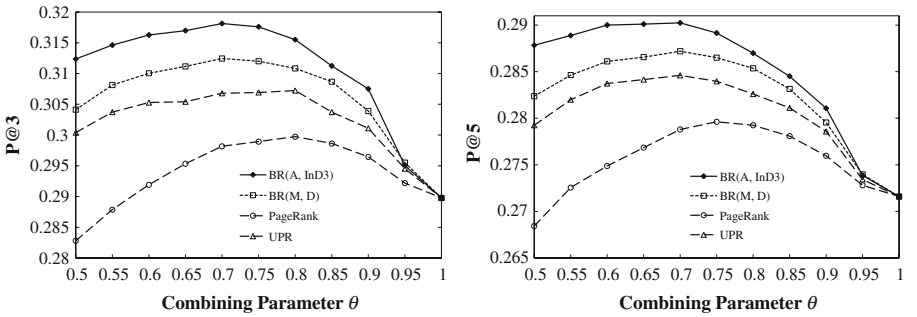

Fig. 6 Search performance in terms of P@3, 5 for BrowseRank algorithms and baselines

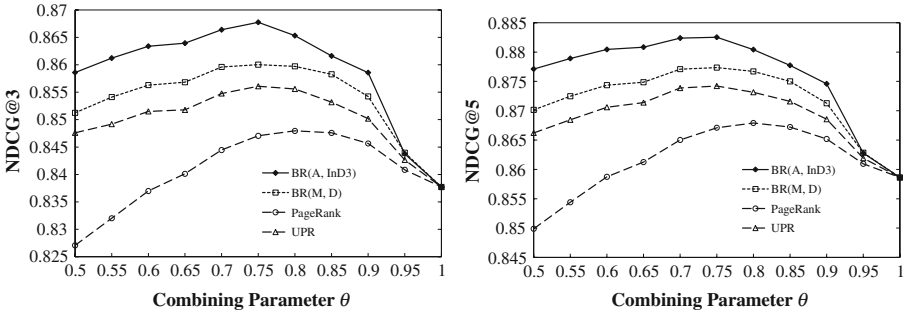

Fig. 7 Search performance in terms of NDCG@3, 5 for BrowseRank algorithms and baselines

NDCG@5, the improvements of BR(A,InD3) over UPR are also statistically significant with p-values of 0.0328, 0.0206, 8.72×10^{-4} and 0.00455, respectively. The only exception is P@5, which p-value is 0.1107 (>0.05) indicating that the improvement is not significant.

As for the above experimental results, we have the following discussions.

1. The algorithms using the user behavior data (i.e., BrowseRank and UPR) outperform PageRank. This seems to indicate that the user behavior data is more reliable and effective than the hyperlink graph, as a data source to compute page importance.
2. The performance of PageRank and UPR are worse than that of BR(M,D). This seems to show that these two baselines are worse estimators of the page importance than BR(M,D), and thus also worse than BR(A,InD3), BR(A,InD2), BR(A,D), BR(M,InD3), and BR(M,InD2) according to our discussions in the previous subsection.

4.2 Website-level experiments

4.2.1 Dataset and baselines

In the second experiment, we still used the same user behavior data. The difference was that we did not distinguish webpages in the same website when running the BrowseRank algorithms. That is, we ignored the transitions between the pages within the same website and also aggregated the transitions from (or to) the pages in the same website. The website-level user browsing graph consists of about 6-million vertices and 60-million edges.

We implemented three baselines, PageRank, TrustRank and UPR, for the website-level experiments. As for these algorithms, we also aggregated the corresponding page importance to the website level. We chose BR(A,InD3) as the representative of the BrowseRank algorithms to compare with these baselines.

4.2.2 Top-20 websites

We listed the top-20 websites ranked by using the four algorithms in Table 4. From this table, we have the following observations.

First, BR(A,InD3) tends to give high ranks to Web 2.0 websites (marked in bold) such as *myspace.com*, *youtube.com*, and *facebook.com*. The main reasons are that Web users visit these websites with high frequencies and often spend long time on them, even if they do not have as many inlinks as Web 1.0 websites like *adobe.com* and *apple.com* do.

Second, some websites like *adobe.com* are ranked very high by PageRank. One reason is that *adobe.com* has a huge number of inlinks for Acrobat Reader and Flash Player downloads. However, Web users do not really visit such websites very frequently and they should not be regarded more important than the websites on which users spend much more time (like *myspace.com* and *facebook.com*).

Third, the ranking results produced by TrustRank are similar to PageRank. The difference is that the well-known websites are ranked higher by TrustRank, mainly because these websites are likely to be included or pointed to by websites in the seed set.

Fourth, UPR also gets better ranking results than PageRank and TrustRank. This also indicates that the user behavior data is more reliable and effective than hyperlink graph as a data source to calculate page importance.

In summary, BR(A,InD3) seems to better represent users' preferences than the baselines.[8]

[8] Note that the top results of the algorithms such as BR(A,InD3) and UPR are impacted by the browsing data we choose to compute the importance scores, which also indicate that the algorithms based on users' browsing data can reflect users' preference timely, while the results of baseline methods will not have large variations in different periods of time.

Table 4 Top 20 websites by four different algorithms

No.	PageRank	TrustRank	UPR	BR(A,InD3)
1	adobe.com	adobe.com	google.com	***myspace.com***
2	passport.com	yahoo.com	msn.com	msn.com
3	msn.com	google.com	yahoo.com	yahoo.com
4	microsoft.com	msn.com	live.com	***youtube.com***
5	yahoo.com	microsoft.com	***myspace.com***	live.com
6	google.com	passport.net	***youtube.com***	***facebook.com***
7	mapquest.com	ufindus.com	***facebook.com***	google.com
8	miibeian.gov.cn	***sourceforge.net***	google.co.th	ebay.com
9	w3.org	***myspace.com***	google.co.uk	***hi5.com***
10	godaddy.com	***wikipedia.org***	passport.com	***bebo.com***
11	statcounter.com	phpbb.com	***hi5.com***	***orkut.com***
12	apple.com	yahoo.co.jp	ebay.com	aol.com
13	live.com	ebay.com	microsoft.com	***friendster.com***
14	xbox.com	nifty.com	***wikipedia.org***	***craigslist.org***
15	passport.com	mapquest.com	aol.com	google.co.th
16	***sourceforge.net***	cafepress.com	ask.com	microsoft.com
17	amazon.com	apple.com	google.ca	***comcast.net***
18	paypal.com	infoseek.co.jp	mywebsearch.com	***wikipedia.org***
19	aol.com	miibeian.gov.cn	***friendster.com***	***pogo.com***
20	***blogger.com***	***youtube.com***	***photobucket.com***	***photobucket.com***

4.2.3 Spam fighting

We randomly sampled 10,000 websites from the 6-million websites and asked human experts to make spam judgments on them. As a result, 2,714 websites are labeled as spam and the rest are labeled as non-spam.

We used the spam bucket distribution to evaluate the performances of the algorithms. Given an algorithm, we sorted the 6-million websites in descending order of the scores that the algorithm produces. Then we put these sorted websites into 15 buckets. The experiment setting was similar to that in (Gyöngyi et al. 2004). The numbers of the labeled spam websites over buckets for PageRank, TrustRank, UPR and BR(A,InD3) are listed in Table 5.

We can see that the number of spam websites in the top buckets of BR(A,InD3) is smaller than those of the baselines. Take the top three buckets as examples. BR(A,InD3) returns five spam sites in its top 883 websites, while PageRank returns 11, TrustRank returns 12, and UPR returns 10. And we also found that BR(A,InD3) can successfully push a lot of spam websites to the tail buckets. UPR also pushes many spam websites to the tail buckets, however, the number is smaller than that of BR(A,InD3). In this regard, we say that BR(A,InD3) is more effective in spam fighting than PageRank, TrustRank and UPR. Furthermore, UPR is better than PageRank and TrustRank, and TrustRank outperforms PageRank. These results are consistent with the results obtained in previous works (Gyöngyi et al. 2004).

Table 5 The number of spam websites over buckets

Bucket no.	Number of websites	PageRank	TrustRank	UPR	BR(A,InD3)
1	15	0	0	0	0
2	148	2	1	1	1
3	720	9	11	9	4
4	2231	22	20	28	18
5	5610	30	34	61	39
6	12600	58	56	101	88
7	25620	90	112	121	87
8	48136	145	128	156	121
9	87086	172	177	188	156
10	154773	287	294	166	183
11	271340	369	320	195	198
12	471046	383	366	224	277
13	819449	434	443	319	323
14	1414172	407	424	484	463
15	2361420	306	328	661	756

The possible explanations to the above experimental findings are as follows:

1. Creating fraudulent hyperlinks, which can hurt PageRank, cannot hurt BR(A,InD3) so much, because the link information is not used in the BrowseRank algorithm.
2. The performance of TrustRank can be affected by the selection of the seed set. For BrowseRank algorithms, seed selection is not necessary.
3. Click fraud, which can hurt UPR, cannot hurt BR(A,InD3) so much, because BR(A,InD3) mainly utilizes transition information and staying time information. Pure clicks cannot really generate meaningful transitions and long staying time.
4. It should be noted that there are always unknown spam techniques that can spam a ranking algorithm. Owing to its nature, BrowseRank will be robust to unknown spams. For example, spammers need to not only click their target pages, but also simulate real transitions; not only just click pages, but also stay at the target pages for long periods of time. As a result, the cost of spamming BrowseRank will be high. In this sense, we say that BrowseRank algorithms can be good choices for resisting unknown Web spams.

5 Related work

Many algorithms have been developed to compute page importance, and roughly they can be classified into two categories: the first category is based on the structure of the link graph, which we call link analysis; the second introduces extra information into the calculation process, such as users' behaviors.

5.1 Methods based on link analysis

The link analysis algorithms have been extensively studied in the literature. Representative methods include PageRank (Brin et al. 1998, Page et al. 1999), and HITS (Kleinberg

1998). The basic idea of PageRank is as follows. If many important pages link to a page on the link graph, then the page is also likely to be important, and the importance information can be propagated along the hyperlinks. A discrete-time Markov chain which simulates a Web surfer's random walk on the hyperlink graph is defined and page importance is calculated as the stationary probability distribution of the Markov chain. HITS is based on the notions of hub and authority to model the two aspects of importance of a webpage. A hub page is the one from which many pages are linking to, while an authority page is the one to which many pages are linked from. In principle, good hubs tend to link to good authorities and vice versa. Previous study has shown that HITS performs comparably with PageRank (Amento et al. 2000).

In addition to PageRank and HITS, many other algorithms have also been proposed. Some of these methods focus on the speed up of the computation of PageRank and HITS (Haveliwala 1999, McSherry 2005), while some others focus on the refinement and enrichment of PageRank and HITS algorithms. Examples include Topic-sensitive PageRank (Haveliwala 2002) and query-dependent PageRank (Richardson et al. 2002). The basic idea of these two algorithms is to introduce topics into the page importance model, and to assume that the endorsement from a page with the same topic is larger than that from a page with a different topic. Some other example algorithms modify the 'personalized vector' (Haveliwala et al. 2003), change the 'damping factor' (Boldi et al. 2005), or introduce different weights to inter-domain and intra-domain links (Langville et al. 2004). Besides, there are also studies on theoretical issues of PageRank algorithm (Bianchini et al. 2005, Haveliwala et al. 2003). Langville et al. (2004) provide a good survey on these related works.

Link analysis algorithms that are robust against link spam have also been proposed recently. For example, TrustRank (Gyöngyi et al. 2004) is a link analysis technique which takes into consideration the reliability of webpages when calculating their importance. In TrustRank, a set of reliable pages are first identified as seed pages. Then the *trust* of the seed pages is propagated to other pages along hyperlinks. Since the propagation starts from reliable pages, TrustRank can be more spam-resistant than PageRank.

5.2 Methods using users behavior data

In recent years, several methods have been proposed to leverage user behavior data in calculation of page importance. Oztekin et al. (2003) combine the hyperlink graph with the usage graph which is obtained from Web logs to adjust the transition weights between two pages. The weights of existing hyperlinks may be adjusted, and in the mean time, some new edges may be created according to the usage data. After that they apply the PageRank algorithm to the new transition matrix to get the page importance. Similarly, Eirinaki and Vazirgiannis (2005) use the usage data, i.e., the visiting frequency by previous users when amending the transition probability. The difference lies in that they only trust the usage data, and wipe off the Web structure information. Their main contribution is that they point out that real user behavior data is very important for page importance calculation.

Liu et al. (2008) also propose an algorithm to compute page importance only based on the user behavior data. Differ from the previous works, they not only use the real transition information, but also use the staying time information. Furthermore, they build a continuous-time Markov process model to combine these two kinds of information. Their work can be regarded as a special case as what we have proposed in this paper.

6 Conclusion and future work

In this paper, we have pointed out that the user behavior data is a more reliable data source for computing page importance than hyperlink graph. We have then proposed a framework to compute effective page importance from user behavior data. In the framework, a stochastic process, named the user browsing process, is used to model the behavior data. We have shown that this process is actually continuous-time time-homogeneous Markov process. Then, a group of methods have been studied to estimate the parameters of the process. Consequently, we obtain a set of algorithms. Our experiments have shown that the proposed algorithm can outperform PageRank and other baselines in several Web search tasks, indicating the advantages of the proposed framework.

In the future, we plan to further investigate the following issues.

1. User behavior data tends to be very sparse. The use of user behavior data can lead to reliable page importance for the head webpages, but not for the tail webpages that have low frequency or even zero frequency in the user behavior data. We plan to find a principled way to deal with this problem.
2. We model the user behavior data using a continuous-time time-homogeneous Markov process. In the process, the selection of the next page is independent of the length of the staying time on the current page. In some cases, this assumption might not hold. A more reasonable way is to relate the estimation of the transition probability to the estimation of the staying time. We plan to use the semi-Markov process to perform this task. Actually, the continuous-time time-homogeneous Markov process is a special case of the semi-Markov process.
3. The content information and other metadata are not used in the proposed algorithms. We will take such information into consideration in the future work.
4. The user browsing process can be used for tasks even beyond the calculation of page importance. For example, this process can be used to detect Web spam, to detect events, and to perform collaborative page recommendation. We plan to investigate new applications of the user browsing process.

References

Amento, B., Terveen, L., & Hill, W. (2000). Does "authority" mean quality? Predicting expert quality ratings of web documents. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval* (pp. 296–303). New York, NY: ACM.

Anderson, W. J. (1991). *Continuous-time Markov chains: An applications-oriented approach*. New York: Springer-Verlag.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. USA: Addison Wesley.

Berger, M. A. (1993). *An introduction to probability and stochastic processes*. New York: Springer-Verlag.

Bianchini, M., Gori, M., & Scarselli, F. (2005). Inside PageRank. *ACM Transactions on Internet Technology, 5*(1), 92–128.

Boldi, P., Santini, M., & Vigna, S. (2005). PageRank as a function of the damping factor. In *WWW'05: Proceedings of the 14th international conference on world wide web* (pp. 557–566). New York, NY: ACM.

Boyd, S., & Vendenberghe, L. (2003). *Convex optimization*. Cambridge: Cambridge University Press.

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

Eirinaki, M., & Vazirgiannis, M. (2005). Usage-based PageRank for web personalization. In *ICDM'05: Proceedings of the fifth IEEE international conference on data mining* (pp. 130–137). Washington, DC: IEEE Computer Society.

Golub, G. H., & Loan, C. F. V. (1996). *Matrix computations*, 3rd ed. Baltimore, MD: Johns Hopkins University Press.

Gyongyi, Z., & Garcia-Molina, H. (2005). Web spam taxonomy. In *first international workshop on adversarial information retrieval on the web (AIRWeb)*.

Gyöngyi, Z., Garcia-Molina, H., & Pedersen, J. (2004). Combating web spam with TrustRank. In *VLDB'04: Proceedings of the thirtieth international conference on very large data bases* (pp. 576–587). VLDB Endowment.

Haveliwala, T. (1999). Efficient computation of PageRank. Technical report 1999-31, Stanford InfoLab.

Haveliwala, T., & Kamvar, S. (2003). The second eigenvalue of the google matrix. Technical report 2003-20, Stanford InfoLab.

Haveliwala, T., Kamvar, S., & Jeh, G. (2003, June). An analytical comparison of approaches to personalizing PageRank. Technical report 2003-35, Stanford InfoLab.

Haveliwala, T. H. (2002). Topic-sensitive PageRank. In *WWW'02: Proceedings of the 11th international conference on world wide web* (pp. 517–526). New York, NY: ACM.

Järvelin, K., & Kekäläinen, J. (2000). IR evaluation methods for retrieving highly relevant documents. In *SIGIR'00: Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval* (pp. 41–48). New York, NY: ACM.

Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Transcation on Information System 20*(4), 422–446.

Kleinberg, J. M. (1998). Authoritative sources in a hyperlinked environment. In *SODA'98: Proceedings of the ninth annual ACM-SIAM symposium on discrete algorithms* (pp. 668–677). Philadelphia, PA: Society for Industrial and Applied Mathematics.

Langville, A. N., & Meyer, C. D. (2004). Deeper inside PageRank. *Internet Mathematics, 1*(3), 335–400.

Liu, Y., Gao, B., Liu, T.-Y., Zhang, Y., Ma, Z., He, S., & Li, H. (2008). BrowseRank: Letting web users vote for page importance. In *SIGIR'08: Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 451–458). New York, NY: ACM.

Lowry, R. (2000) *Concepts and applications of inferential statistics*. Vassar College. http://faculty.vassar.edu/lowry/webtext.htm.

McSherry, F. (2005). A uniform approach to accelerated PageRank computation. In *WWW'05: Proceedings of the 14th international conference on world wide web* (pp. 575–582). New York, NY: ACM.

Oztekin, B. U., Ertoz, L., Kumar, V., & Srivastava, J. (2003). Usage aware PageRank. Technical report 03-010, University of Minnesota.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web. Technical report 1999-66, Stanford InfoLab. Previous number = SIDL-WP-1999-0120.

Rice, J. A. (1995). *Mathematical statistics and data analysis*, 2nd ed. USA: Duxbery Press.

Richardson, M., & Domingos, P. (2002). The intelligent surfer: Probabilistic combination of link and content information in PageRank. In *advances in neural information processing systems 14* (pp. 1441–1448). Cambridge, MA: MIT Press.

Robertson, S. E. (1997). Overview of the Okapi projects. *Journal of Documentatioin, 53*(1), 3–7.

Sorenson, H. W. (1980). *Parameter estimation: Principles and problems*. USA: Marcel Dekker.

Stewart, W. J. (1994). *Introduction to the numerical solution of Markov chains*. Princeton, NJ: Princeton University Press.

Wang, Z. K., & Yang, X. Q. (1992). *Birth and death processes and Markov chains*. New York: Springer-Verlag.

White, R. W., Bilenko, M., & Cucerzan, S. (2007). Studying the use of popular destinations to enhance web search interaction. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 159–166). New York, NY: ACM.