BrowseRank: Letting Web Users Vote for Page Importance

Yuting Liu*
School of Science
Beijing Jiaotong University
Beijing, 100044, P. R. China
liuyt_njtu@hotmail.com

Bin Gao, Tie-Yan Liu
Microsoft Research Asia
4F, Sigma Center,
No. 49, Zhichun Road.
Beijing, 100190, P. R. China
{bingao,tyliu}@microsoft.com

Ying Zhang*
Dept. of Computer Science
Nankai University
Tianjin, 300071, P. R. China
nkjaiden@hotmail.com

Zhiming Ma
Academy of Mathematical and Systems Science
Chinese Academy of Sciences
Beijing, 100190, P. R. China
mazm@amt.ac.cn

Shuyuan He
School of Mathematical Sciences
Peking University
Beijing, 100871, P. R. China
syhe@math.pku.edu.cn

Hang Li
Microsoft Research Asia
4F, Sigma Center,
No. 49, Zhichun Road.
Beijing, 100190, P. R. China
hangli@microsoft.com

ABSTRACT

This paper proposes a new method for computing page importance, referred to as BrowseRank. The conventional approach to compute page importance is to exploit the link graph of the web and to build a model based on that graph. For instance, PageRank is such an algorithm, which employs a discrete-time Markov process as the model. Unfortunately, the link graph might be incomplete and inaccurate with respect to data for determining page importance, because links can be easily added and deleted by web content creators. In this paper, we propose computing page importance by using a 'user browsing graph' created from user behavior data. In this graph, vertices represent pages and directed edges represent transitions between pages in the users' web browsing history. Furthermore, the lengths of staying time spent on the pages by users are also included. The user browsing graph is more reliable than the link graph for inferring page importance. This paper further proposes using the *continuous-time* Markov process on the user browsing graph as a model and computing the stationary probability distribution of the process as page importance. An efficient algorithm for this computation has also been devised. In this way, we can leverage hundreds of millions of users' *implicit voting* on page importance. Experimental results show that BrowseRank indeed outperforms the baseline methods such as PageRank and TrustRank in several tasks.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia.

General Terms

Algorithms, Experimentation, Theory

*This work was performed when the first and the fourth authors were interns at Microsoft Research Asia.

SIGIR'08, July 20–24, 2008, Singapore.

Keywords

PageRank, User behavior data, User browsing graph, Continuous-time Markov process, Q-matrix.

1. INTRODUCTION

Page importance, which represents the 'value' of an individual page on the web, is a key factor for web search, because for contemporary search engines, the crawling, indexing, and ranking are usually guided by this measure. Because the scale of the web is extremely large and the web evolves dynamically, accurately calculating the importance scores of web pages becomes critical, and also poses a great challenge to search engines. In this paper, we propose a new method for computing page importance, as our answer to the challenge.

Currently, page importance is calculated by using the link graph of the web and such a process is called link analysis. Well known link analysis algorithms include HITS [15], PageRank[5, 18], and others [4, 8, 9, 11, 12, 16, 17, 20]. Most of the algorithms assume that if many important pages link to a page on the link graph, then the page is also likely to be important, and they calculate the importance of the page on the basis of a model defined on the link graph. Link analysis algorithms have been successfully applied to web search.

For example, PageRank employs a *discrete-time* Markov process on the web link graph to compute page importance, which in fact simulates a random walk along the hyperlinks on the web of a web surfer. Although PageRank has its advantages, it also has certain limitations as a model for representing page importance.

1. The link graph, which PageRank relies on, is not a very reliable data source, because hyperlinks on the web can be easily added or deleted by web content creators. For example, purposely creating a large number of hyperlinks is a favorite technique (e.g. link farm and link exchange) of web spammers [7], and such kinds of hyperlinks are not suitable for calculating page importance.

2. PageRank only models a random walk on the link graph, but does not take into consideration the lengths of time which the web surfer spends on the web pages during the random walk. Such information can be a good indicator of the quality and thus importance of the pages.

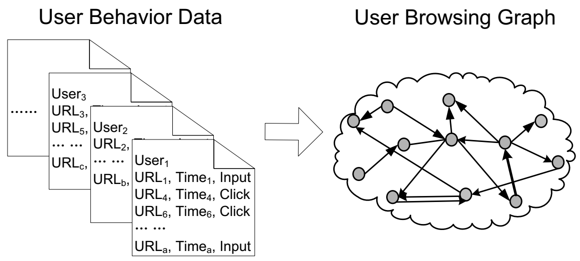

Figure 1: User behavior data and browsing graph

To overcome these drawbacks, we consider using a more reliable data source and employing a more powerful mathematical model.

The first question is whether we can find a better data source than the link graph. Our answer is to utilize the *user browsing graph*, generated from user behavior data.[1] User behavior data can be recorded by Internet browsers at web clients[2] and collected at a web server. An example of the data is shown in Figure 1. Each record in the data contains the information of a visit by an anonymous user: URL, time, and method of visiting (URL input or hyperlink click from previous page). In our experiment, such data was recorded and collected from an extremely large group of users under legal agreements with them. Information which could be used to recognize their identities was not included. By integrating the data from hundreds of millions of web users, we can build a user browsing graph, in which vertices represent web pages and directed edges represent real transitions between web pages by users, and furthermore the lengths of time spent on the pages by the users are also included.

The user browsing graph can more precisely represent the web surfer's random walk process, and thus is more useful for calculating page importance. The more visits of the page made by the users and the longer time periods spent by the users on the page, the more likely the page is important. With this graph, we can leverage hundreds of millions of users' *implicit voting* on page importance. In this regard, our approach is in accordance with the concept of Web 2.0.[3]

The second question is what kind of algorithm we should use to leverage the new data source. Obviously, the use of a discrete-time Markov process would not be sufficient. In this paper, we define a continuous-time Markov process [23] as the model on the user browsing graph. If we further assume the process to be time-homogenous (which is reasonable as discussed in Section 3), then the stationary probability distribution of the process can be used to define the importance of web pages. We employ the algorithm referred to as BrowseRank, to efficiently compute the stationary probability distribution of the continuous-time Markov process. We make use of an additive noise model to represent the observations with regard to the Markov process and to conduct an unbiased and consistent estimation of the parameters in the process. In addition, we adopt an embedded Markov chain based technology to speed up the calculation of the stationary distribution. Hereafter, if there is no confusion, we will call both the algorithm and the scores output by the algorithm 'BrowseRank'.

Experimental results show that BrowseRank can achieve better performance than existing methods, including PageRank and TrustRank [8] in important page finding, spam page fighting, and relevance ranking.

The novelty of this paper lies in the following points. First, we propose using a user browsing graph, mined from user behavior data for computing page importance, which is more reliable and richer than a web link graph. Second, we propose using the *continuous-time* Markov process to model a random walk on the user browsing graph, which can more powerfully represent page importance. Third, we propose an algorithm called BrowseRank to efficiently compute page importance scores.

The rest of the paper is organized as follows. Section 2 introduces related work. Section 3 describes the user browsing graph, the continuous-time Markov process model, and the BrowseRank algorithm. Experimental results are reported in Section 4. Conclusion and future work are given in Section 5.

2. RELATED WORK

PageRank [5, 18] and HITS [15] are popular link analysis algorithms in the literature. The basic idea of PageRank is as follows: the link from a webpage to another can be regarded as an endorsement of the linking page, the more links pointed to a page, the more likely it is important, and this importance information can be propagated across the vertices in the graph. A discrete-time Markov process model which simulates a web surfer's random walk on the graph is defined and page importance is calculated as the stationary probability distribution of the Markov process. HITS is based on the notions of hub and authority to model the two aspects of importance of a webpage. A hub page is one from which many pages are linked to, while an authority page is 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 to PageRank [1].

Many algorithms have been developed in order to further improve the accuracies and efficiencies of PageRank. Some work focuses on speed-up of the computation [9, 17], while others focus on refinement and enrichment of the model. For example, Topic-sensitive PageRank [12] and query-dependent PageRank [20] have been proposed. The basic idea of these algorithms is to introduce topics and assume that the endorsement from a page that belongs to the same topic is larger. Other variations of PageRank include those modifying the 'personalized vector' [11], changing the 'damping factor' [4], and introducing inter-domain and intra-domain link weights [16]. Besides, there is also work on theoretic issues of PageRank [3] and [10]. Langville *et al* [16] provide a good survey on PageRank and related work.

Link analysis algorithms that are robust against link spam have been proposed. For example, TrustRank [8] is a link analysis technique which takes into consideration the reliability of web pages when calculating the importance of pages. 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 on the web link graph. Since the propagation in TrustRank starts from the reliable pages, TrustRank can be more spam-resistant than PageRank.

3. BROWSERANK

3.1 User Behavior Data

Many web service applications assist users in their accesses to the web; sometimes they record user behaviors under agreements with them.

When a user surfs on the web, she usually has some information need. To browse a new page, the user may choose to click on the hyperlink on another page pointing to it, or to input the URL of it

---

[1]Note that it is also possible to combine link graph and user behavior data to compute page importance. We will not discuss more about this possibility in this paper, and simply leave it as future work.

[2]Web search engines such as Google, Yahoo, and Live Search provide client software called toolbars, which can serve the purpose.

[3]http://en.eikipedia.org/wiki/Web_2

Table 1: Examples of user behavior data

URL	TIME	TYPE
http://aaa.bbb.com/	2007-04-12, 21:33:05	INPUT
http://aaa.bbb.com/1.htm	2007-04-12, 21:34:11	CLICK
http://ccc.ddd.org/index.htm	2007-04-12, 21:34:52	CLICK
http://eee.fff.edu/	2007-04-12, 21:39:03	INPUT
...	...	...

into the web browser. The user may repeat this until she finds the information or gives up. The user behavior data can be recorded and represented in triples consisting of <URL, TIME, TYPE> (see Table 1 for examples). Here, URL denotes the URL of the webpage visited by the user, TIME denotes the time of the visit, and TYPE indicates whether the visit is by a URL input (INPUT) or by a hyperlink click on the previous page (CLICK). The records are sorted in chronological order.

From the data we extract transitions of users from page to page and the time spent by users on the pages as follows:

1) Session segmentation

We define a session as a logical unit of user's browsing. In this paper we use the following two rules to segment sessions. First, if the time of the current record is 30 minutes behind that of the previous record, then we will regard the current record as the start of a new session [24]; otherwise, if the type of the record is 'INPUT', then we will regard the current record as the start of a new session. We refer to the two rules as the *time rule* and the *type rule* hereafter.

2) URL pair construction

Within each session, we create URL pairs by putting together the URLs in adjacent records. A URL pair indicates that the user transits from the first page to the second page by clicking a hyperlink.

3) Reset probability estimation

For each session segmented by the *type rule*, the first URL is directly input by the user and not based on a hyperlink. Therefore, such a URL is 'safe' and we call it *green traffic*[4] . When processing user behavior data, we regard such URLs as the destinations of the random reset (when users do not want to surf along hyperlinks). We normalize the frequencies of URLs being the first one in such sessions to get the reset probabilities of the corresponding web pages.

4) Staying time extraction

For each URL pair, we use the difference between the time of the second page and that of the first page as the observed staying time on the first page. For the last page in a session, we use the following heuristics to decide its observed staying time. If the session is segmented by the *time rule*, we randomly sample a time from the distribution of observed staying time of pages in all the records and take it as the observed staying time [5]. If the session is segmented by the *type rule*, we use the difference between the time of the last page in the session and that of the first page of the next session (INPUT page) as the staying time.

By aggregating the transition information and the staying time information extracted from the records by an extremely large number of users, we are able to build a user browsing graph (see Figure 1). Each vertex in the graph represents a URL in the user behavior data, associated with reset probability and staying time as metadata. Each directed edge represents the transition between two vertices, associated with the number of transitions as its weight. In other words, the user browsing graph is a weighted graph with vertices containing metadata and edges containing weights. We denoted it as $G = < V, W, T, \sigma >$, where $V = \{v_i\}$, $W = \{w_{ij}\}$, $T = \{T_i\}$, $\sigma = \{\sigma_i\}$, $(i, j = 1, \ldots, N)$ denote vertices, weights of edges, lengths of staying time, and reset probabilities, respectively. N denotes the number of web pages in the user browsing graph.

3.2 Model

To better leverage the information on staying time, we propose employing a continuous-time time-homogeneous Markov process for representing a random walk on the user browsing graph.

3.2.1 Assumptions

When using the new model, we need to make the following assumptions.

1) Independence of users and sessions

The browsing processes of different users in different sessions are independent. In other words, we treat web browsing as a stochastic process, with the data observed in each session by a user as an *i.i.d.* sample of this process. This independence assumption is widely used when one estimates parameters from observed data in statistics.

2) Markov property

The page that a user will visit next only depends on the current page, and is independent of the pages she visited previously. This assumption is also a basic assumption in PageRank.

3) Time-homogeneity

The browsing behaviors of users (e.g. transitions and staying time) do not depend on time points. Although this assumption is not necessarily true in practice, it is mainly for technical convenience. Note that this is also a basic assumption in PageRank.

Based on these assumptions, we can build a model of continuous-time time-homogeneous Markov process to mimic a random walk on the user browsing graph. In a similar way as in PageRank, the stationary probability distribution of this process can be used to measure the importance of pages.

3.2.2 Continuous-time Markov Model

Suppose there is a web surfer walking through all the webpages. We use X_s to denote the page which the surfer is visiting at time s, $s > 0$. Then, with the aforementioned three assumptions, the process $X = \{X_s, s \geq 0\}$ forms a continuous-time time-homogenous Markov process. Let $p_{ij}(t)$ denotes the transition probability from page i to page j for time interval (also referred to as time increment in statistics) t in this process. One can prove that there is a stationary probability distribution π, which is unique and independent of t [23], associated with $P(t) = [p_{ij}(t)]_{N \times N}$, such that for any $t > 0$,

$$\pi = \pi P(t) \tag{1}$$

The i^{th} entry of the distribution π stands for the ratio of the time the surfer spends on the i^{th} page over the time she spends on all the pages when time interval t goes to infinity. In this regard, this distribution π can be a measure of page importance.

In order to compute this stationary probability distribution, we need to estimate the probability in every entry of the matrix $P(t)$. However, in practice, this matrix is usually difficult to obtain, because it is hard to get the information for all possible time intervals. To tackle this problem, we propose a novel algorithm which is instead based on the transition rate matrix [23]. The details of this algorithm will be given in Section 3.3.

3.3 Algorithm

[4]In practice, users often visit web pages by typing the URLs of the pages or selecting from bookmarks at web browsers. We call such kind of visits *green traffic*, because the pages visited in this way are safe, interesting, and/or important for the users.

[5]For the definition on the distribution of observed staying time, please refer to Section 4.1.1.

3.3.1 Overview

We make use of the transition rate matrix to compute the stationary probability distribution of $P(t)$, as a measure of page importance. We call the corresponding algorithm as BrowseRank.

The transition rate matrix is defined as the derivative of $P(t)$ when t goes to 0, if it exists. That is, $Q = P'(0)$. We call the matrix $Q = (q_{ij})_{N\times N}$ the Q-matrix for short. It has been proven that when the state space is finite there is a one-to-one correspondence between the Q-matrix and $P(t)$, and $-\infty < q_{ii} < 0$; $\sum_j q_{ij} = 0$[23]. Due to this correspondence, one also uses Q-Process to represent the original continuous-time Markov process, that is, the browsing process $X = \{X_s, s \geq 0\}$ defined before is a Q-Process because of the finite state space.

There are two advantages of using the Q-matrix:

1. The parameters in the Q-matrix can be effectively estimated from the data, according to the discussions in Section 3.3.2 and Section 3.3.3.

2. Based on the Q-matrix, there is an efficient way of computing the stationary probability distribution of $P(t)$, according to the following theorem.

Before giving the theorem of how to efficiently compute the stationary probability distribution of Q-process (Theorem 1), we need to introduce a concept named embedded Markov chian (EMC) [22]corresponding to a Q-process. The so-called EMC is a discrete-time Markov process featured by a transition probability matrix with zero values in all its diagonal positions and $-\frac{q_{ij}}{q_{ii}}$ in the off-diagonal positions, where all parameters $q_{ij}, i, j = 1, \ldots, N$ have the same definitions as before.

THEOREM 1. *Suppose X is a Q-process, and Y is the Embedded Markov Chain derived from its Q-matrix. Let* $\pi = (\pi_1, \ldots, \pi_N)$ *and* $\tilde{\pi} = (\tilde{\pi}_1, \ldots, \tilde{\pi}_N)$ *denote the stationary probability distributions of the process X and Y, then we have*

$$\pi_i = \frac{\frac{\tilde{\pi}_i}{q_{ii}}}{\sum_{j=1}^{N} \frac{\tilde{\pi}_j}{q_{jj}}} \tag{2}$$

Refer to [22] for the proof of Theorem 1.

Note that the process Y is a discrete-time Markov chain, so its stationary probability distribution $\tilde{\pi}$ can be calculated by many simple and efficient methods such as the power method [6].

Next we will explain how to estimate the parameters in the Q-matrix, or equivalently parameter q_{ii} and the transition probabilities $-\frac{q_{ij}}{q_{ii}}$($-\frac{q_{ij}}{q_{ii}} \geq 0$ due to $q_{ii} < 0$) in the EMC.

3.3.2 Estimation of q_{ii}

According to [22], for a Q-Process, the staying time T_i on the i^{th} vertex is governed by an exponential distribution parameterized by q_{ii}:

$$P(T_i > t) = exp(q_{ii}\, t) \tag{3}$$

This implies that we can estimate q_{ii} from large numbers of observations on the staying time in the user behavior data.

This task is, however, non-trivial because the observations in the user behavior data usually contain noise due to Internet connection speed, page size, page structure, and other factors. In other words, the observed values do not completely satisfy the exponential distribution [6]. To tackle this challenge, we use an additive noise model to represent the observations and to conduct an unbiased and consistent estimation of parameter q_{ii}.

Suppose for page i, we have m_i observations on its staying time in the user behavior data, denoted as $Z_1, Z_2, \ldots, Z_{m_i}$, and they have the same distribution as random variable Z. Without loss of generality, we suppose that Z is the combination of real staying time T_i and noise U, i.e.,

$$Z = U + T_i \tag{4}$$

Suppose that noise U is governed by a Chi-square distribution as $Chi(k)$ [7], then its mean and variance will be k and $2k$ respectively. Further suppose that the mean and variance of Z are μ and σ^2. By assuming U and T_i to be independent, we have [19]:

$$\mu = E(Z) = E(U + T_i) = k - \frac{1}{q_{ii}} \tag{5}$$

$$\sigma^2 = Var(Z) = Var(U + T_i) = 2k + \frac{1}{q_{ii}^2} \tag{6}$$

Note that the *sample mean* $\bar{Z} = \frac{1}{m_i}\sum_{l=1}^{m_i} Z_l$ and *sample variance* $S^2 = \frac{1}{m_i - 1}\sum_{l=1}^{m_i}(Z_l - \bar{Z})^2$are unbiased and consistent estimators for μ and σ^2 [19]. We then estimate q_{ii} by solving the following optimization problem [8]:

$$\begin{aligned} \min_{q_{ii}} \quad & ((\bar{Z} + \tfrac{1}{q_{ii}}) - \tfrac{1}{2}(S^2 - \tfrac{1}{q_{ii}^2}))^2 \\ s.t. \quad & q_{ii} < 0 \end{aligned} \tag{7}$$

3.3.3 Estimation of Transition Probability in EMC

Transition probabilities in the EMC describe the 'pure' transitions of the surfer on the user browsing graph. Estimation of them can be based on the observed transitions between pages in the user behavior data. It can also be related to the green traffic in the data. We use the following method to integrate these two kinds of information for the estimation.

We start with the user browsing graph $G =< V, W, T, \sigma >$. We then add a pseudo-vertex (the $(N+1)^{th}$ vertex) to G, and add two types of edges: the edges from the last page in each session to the pseudo-vertex, associated with the click number of the last page as its weight; and the edges from the pseudo-vertex to the first page in each session, associated with the reset probability. We denote the new graph as $\tilde{G} =< \tilde{V}, \tilde{W}, T, \tilde{\sigma} >$, where $|\tilde{V}| = N + 1$, $\tilde{\sigma} =< \tilde{\sigma}_1, \ldots, \tilde{\sigma}_N, 0 >$. Then we explain the EMC model as the random walk on this new graph $\tilde{G}$. Based on *the law of large number* [19], the transition probabilities in the EMC are estimated as below,

$$-\frac{q_{ij}}{q_{ii}} = \begin{cases} \alpha \frac{\tilde{w}_{ij}}{\sum_{k=1}^{N+1} \tilde{w}_{ik}} + (1-\alpha)\,\sigma_j, & i \in V, j \in \tilde{V} \\ \sigma_j, & i = N+1, j \in V \end{cases} \tag{8}$$

The intuitive explanation of the above transition is as follows. When the surfer walks on the user browsing graph, she may go ahead along the edges with the probability α, or choose to restart from a new page with the probability $(1-\alpha)$. The selection of the new page is determined by the reset probability.

One advantage of using (8) for estimation is that the estimation will not be biased by the limited number of observed transitions. The other advantage is that the corresponding EMC is primitive, and thus has a unique stationary distribution (see Theorem 2).

[6]Figure 2 shows the distribution of observed staying time, which is not an exponential distribution, as we expected (see section 4.1.1).

[7]Chi-square distribution is widely used to model additive noises whose support is within $[0, +\infty)$.

[8]Note that to get this optimization problem, we actually solve k from equation (5) and (6) respectively, and minimize the difference between these two solutions. In this way, we can leverage both mean and variance for parameter estimation.

Table 2: The BrowseRank algorithm

Input: the user behavior data.
Output: the page importance score π
Algorithm:
1. Construct the user browsing graph (see Section 3.1).
2. Estimate q_{ii} for all pages(see Section 3.3.2).
3. Estimate the transition probability matrix of the EMC and then get its stationary probability distribution by means of power method (see Section 3.3.3).
4. Compute the stationary probability distribution of the Q-process by using of equation (2).

Therefore, we can use the power method to calculate this stationary distribution in an efficient manner.

THEOREM 2. *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 equation (8), the process Y is primitive, i.e., the transition graph of process Y is strongly connected (which means there is a directed path from any node to any other node in the graph).*

Refer to the appendix for the proof of Theorem 2.

In summary, we get the flow chart of BrowseRank as Table 2.

4. EXPERIMENTAL RESULTS

We have conducted experiments to verify the effectiveness of the proposed BrowseRank algorithm. We report the experimental results in this section. The first experiment was conducted at the website level, to test the performance of BrowseRank on finding important websites and depressing spam sites. The second experiment was conducted at the webpage level, to test the effectiveness of BrowseRank on improving relevance ranking.

4.1 Website-Level BrowseRank

4.1.1 Dataset and Baselines

We used a user behavior dataset, collected from the World Wide Web by a commercial search engine in the experiments. All possible privacy information was rigorously filtered out and the data was sampled and cleaned to remove bias as much as possible. There are in total over 3-billion records, and among them there are 950-million unique URLs. The distribution of the observed staying time of one web page randomly selected from a part of the dataset in which all web pages have a large number of observations is shown in a log-linear scale in Figure 2[9]. From the figure, we can see that the curve is not straight at the beginning, indicating that it does not follow an exact exponential distribution. This validates our arguments on the noisy observations on the staying time.

When running BrowseRank at website-level, we did not distinguish web pages in the same website. That is, we ignored the transitions between the pages in the same website and aggregated the transitions from (or to) the pages in the same website. As a result, we created a user browsing graph at website-level, consisting of 5.6-million vertices and 53-million edges. We also obtained a *link graph* containing the 5.6-million websites from the commercial search engine. There are in total 40-million websites in this link graph. We computed PageRank and TrustRank from it as baselines.

4.1.2 Top-20 Websites

We listed the top-20 websites ranked by using different algorithms in Table 3. From this table, we can make the following observations:

[9]We plot the top 100 seconds graph to show the curve clearly.

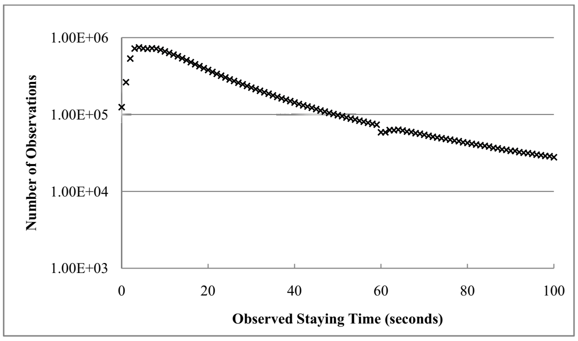

Figure 2: The distribution of the observed staying time

First, BrowseRank tends to give high ranks to Web 2.0 websites (marked in bold) such as *myspace.com*, *youtube.com*, *facebook.com*. The reason is that web users visit these websites with high frequencies and often spend much time on them, even if they do not have as many inlinks as Web 1.0 websites like *adobe.com* and *apple.com* do. Note this reflects users' real information needs.

Second, some websites like *adobe.com* are ranked very high by PageRank. One reason is that *adobe.com* has millions 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 than by PageRank, mainly because these websites are likely to be included or pointed to by websites in the seed set.

In summary, the ranking results given by BrowseRank seem to better represent users' preferences than PageRank and TrustRank.

4.1.3 Spam Fighting

We randomly sampled 10,000 websites from the 5.6 million websites and asked human experts to make spam judgments on them. 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 5.6-million websites in descending order of the scores that the algorithm produces. Then we put these sorted websites into 15 buckets. This experiment is similar to the experiments in [8]. The numbers of the labeled spam websites over buckets for PageRank, TrustRank, and BrowseRank are listed in Table 4.

We see that BrowseRank can successfully push many spam websites to the tail buckets, and the number of spam websites in the top buckets in BrowseRank is smaller than PageRank and TrustRank. That is to say, BrowseRank is more effective in spam fighting than PageRank and TrustRank. The reasons that BrowseRank outperforms the other algorithms are as follows:

1) Creating inlinks, which can hurt PageRank, cannot hurt BrowseRank so much, because the link information is not used in BrowseRank.

2) The performance of TrustRank can be affected by the selection of the seed set and the determination of the seed distribution in the link graph. For BrowseRank, seed selection and seed distribution determination are not necessary.

Furthermore, the performance of TrustRank is better than PageRank, which is consistent with the result obtained in previous work [8].

Table 3: Top 20 websites by three different algorithms

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

4.2 Page-Level BrowseRank

4.2.1 Ranking in Web Search

In web search engines, the retrieved web pages for a given query are often ranked based on two factors: relevance rank and importance rank. A linear combination of these two ranking lists is then created [2]:

$$\theta \times rank_{relevance} + (1 - \theta) \times rank_{importance} \tag{9}$$

Here $0 \le \theta \le 1$ is the combining parameter.

4.2.2 Dataset

Again, we used the user behavior data and the link graph. This time we ran all the algorithms at the page level.

In addition, we also obtained a large dataset from the same search engine, containing 8000 queries and their associated webpages. For the associated webpages of each query, three researchers in web search were hired to independently score each page's relevancy to the given query (1 - relevant, 0 - irrelevant). These scores were then summed, with pages having total scores of at least 2 labeled as relevant, and the others marked as irrelevant. This dataset has been preprocessed, and spam pages within it have been removed in advance. Therefore, it is not necessary to evaluate TrustRank on this dataset, and we only take PageRank as baseline. We reserve the webpages which are in this dataset and also appear in the user behavior data for the experiment.

4.2.3 Results

In this experiment, we compared the performances on ranking using PageRank and BrowseRank as the page importance models for ranking. BM25 [21]was used as the relevance model for ranking.

We adopted three measures to evaluate the ranking performances: MAP [2], Precision (P@n) [2], and Normalized Discount Cumulative Gain (NDCG@n) [13, 14]. The experimental results are presented in Figures 3 to 9.

From the figures, we can see that BrowseRank consistently outperforms PageRank in all parameter settings in terms of all evaluation measures. For example, from Figure 7, we can see that NDCG@5 of BM25 is 0.853 (when $\theta = 1$), BrowseRank hits its peak NDCG@5 value of 0.876 when $\theta = 0.70$, and the peak NDCG@5 value of PageRank is 0.862 when $\theta = 0.80$.

Table 4: Number of spam websites over buckets

Bucket No.	Number of Websites	PageRank	TrustRank	BrowseRank
1	15	0	0	0
2	148	2	1	1
3	720	9	11	4
4	2231	22	20	18
5	5610	30	34	39
6	12600	58	56	88
7	25620	90	112	87
8	48136	145	128	121
9	87086	172	177	156
10	154773	287	294	183
11	271340	369	320	198
12	471046	383	366	277
13	819449	434	443	323
14	1414172	407	424	463
15	2361420	306	328	756

We also conducted t-tests at a confidence level of 95%. In terms of MAP, the improvement of BrowseRank over PageRank is statistically significant with a p-value of 0.0063. In terms of P@3, P@5, NDCG@3, and NDCG@5, the improvements are also statistically significant with p-values are 0.00026, 0.0074, 3.98×10^{-7}, and 3.57×10^{-6}, respectively.

4.2.4 Discussion

To further understand the user behavior data and our proposed algorithm, we consider two simple algorithms that also use user behavior data or user browsing graphs: PageRank-UBG (weighted PageRank computed on the user browsing graph), and Naive BrowseRank (product of the number of clicks and average observed staying time).

Based on the experimental results given in Figures 8, 9, and 10, we can make the following observations:

1. Both simple methods can also outperform PageRank, as can be seen from the comparing Figures 3, 5 and 7 with Figures 8, 9, and 10. This seems to indicate that the user browsing graph is more reliable and effective than the link graph, as a data source to compute page importance.

2. NaiveBR performs better than PageRank-UBG. This is reasonable since PageRank-UBG only makes use of the transition information in the user browsing graph, while in addition to the transition information NaiveBR also utilizes the information on staying time.

3. BrowseRank consistently performs better than the simple algorithms. This indicates that the model and algorithm we propose are more effective than the simple methods.

5. CONCLUSION AND FUTURE WORK

In this paper, we have pointed out that a web link graph is not a reliable data source for computing page importance. Furthermore, existing link analysis algorithms like PageRank are also too simple to infer page importance. To deal with these problems, we propose using user behavior data to mine a user browsing graph, building a continuous-time Markov process model on the graph, and employing an efficient algorithm to calculate page importance scores with the model.

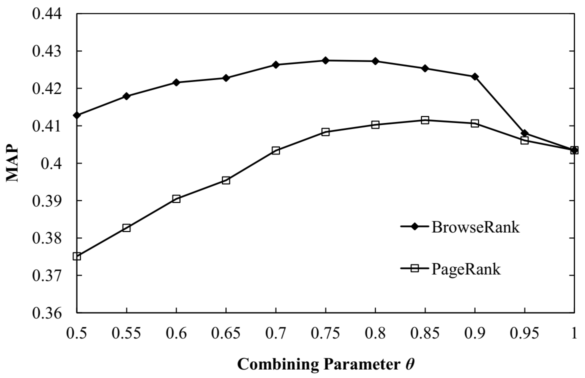

Figure 3: Search performance in terms of MAP for BrowseRank and PageRank

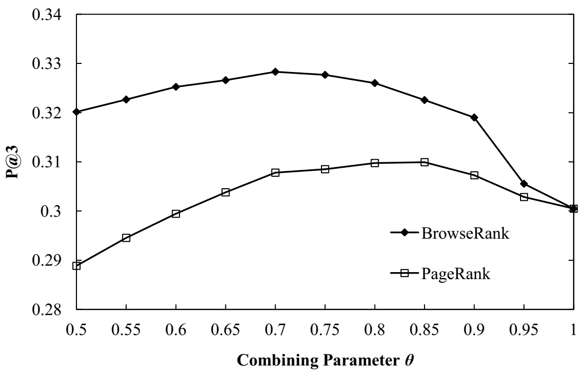

Figure 4: Search performance in terms of P@3 for BrowseRank and PageRank

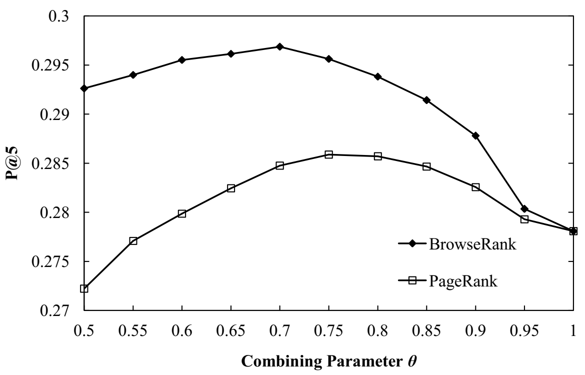

Figure 5: Search performance in terms of P@5 for BrowseRank and PageRank

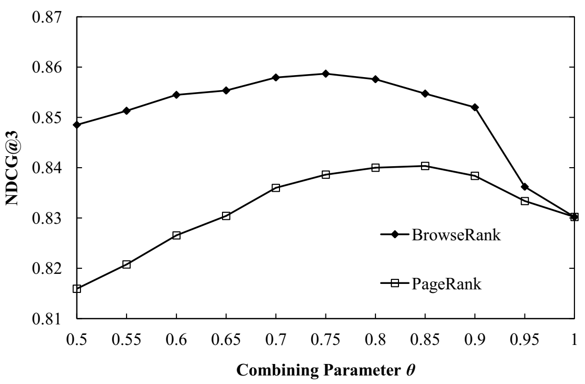

Figure 6: Search performance in terms of NDCG@3 for BrowseRank and PageRank

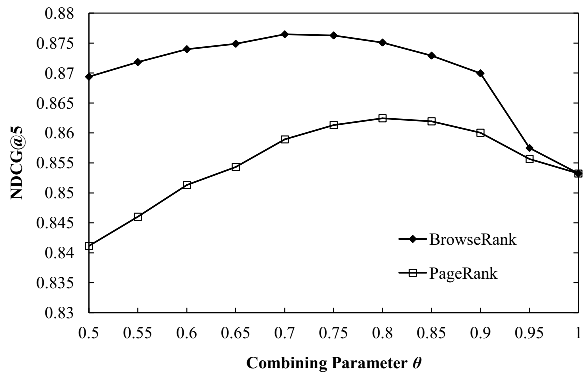

Figure 7: Search performance in terms of NDCG@5 for BrowseRank and PageRank

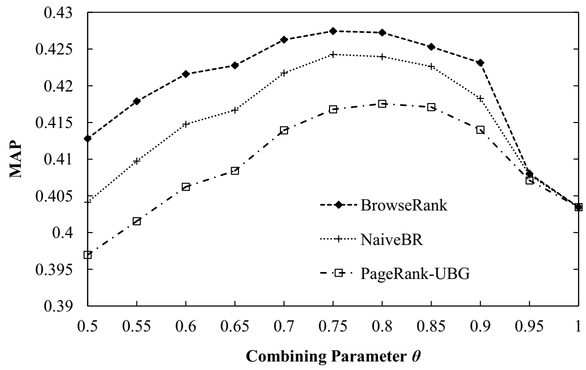

Figure 8: Search performance in terms of MAP for BrowseRank and two simple algorithms

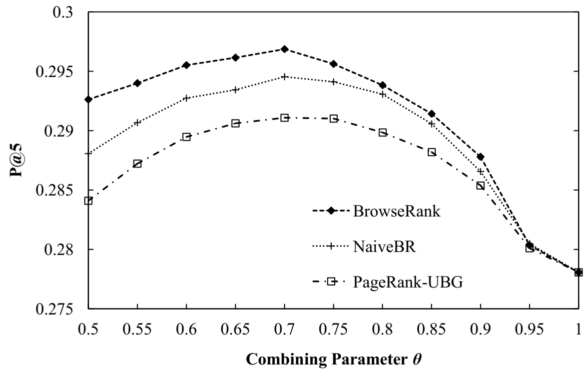

Figure 9: Search performance in terms of P@5 for BrowseRank and two simple algorithms

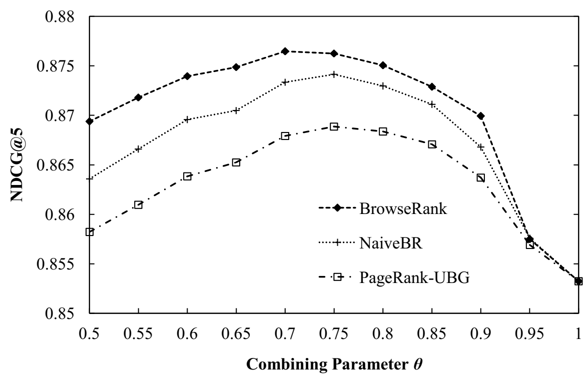

Figure 10: Search performance in terms of NDCG@5 for BrowseRank and two simple algorithms

The user browsing graph data is more reliable and richer than the conventional link graph data, and furthermore the continuous-time Markov model is more powerful than the existing models. Thus the use of them will result in more accurate results in page importance calculation. We name the new algorithm BrowseRank. Our experimental results show that BrowseRank outperforms PageRank and TrustRank in two web search tasks, indicating that the proposed approach really does have the stated advantages.

There are still several technical issues which need to be addressed as future work:

1) User behavior data tends to be sparse. The use of user behavior data can lead to reliable importance calculation for the head web pages, but not for the tail web pages, which have low frequency or even zero frequency in the user behavior data. One possibility is to use the link graph to conduct some smoothing. We need to find a principled way to deal with this problem.

2) The assumption on time homogeneity is made mainly for technical convenience. We plan to investigate whether we can still obtain an efficient algorithm if this assumption is withdrawn.

3) The content information and metadata was not used in BrowseRank. However, in general, a larger page often means longer staying time. We will take the metadata like page size into consideration to normalize the user staying time in the next version.

6. ACKNOWLEDGMENTS

We thank Daxin Jiang, Zhi Chen, Tao Qin, Zhen Liao, and Congkai Sun for their help in the data preparation and code review for the work, and thank Andrew Arnold for giving many helpful suggestions and his intensive polishing of the paper.

7. REFERENCES

[1] B. Amento, L. Terveen, and W. Hill. Does authority mean quality? Predicting expert quality ratings of web documents. In *SIGIR ' 00*. ACM, 2000.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, May 1999.

[3] M. Bianchini, M. Gori, and F. Scarselli. Inside pagerank. *ACM Trans. Interet Technol.*, 5(1):92–128, 2005.

[4] P. Boldi, M. Santini, and S. Vigna. Pagerank as a function of the damping factor. In *WWW ' 05*. ACM, 2005.

[5] 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.

[6] G. H. Golub and C. F. V. Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[7] Z. Gyongyi and H. Garcia-Molina. Web spam taxonomy. In *AIRWeb '05*. 2005.

[8] Z. Gyongyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *VLDB '04*, pages 576–587. VLDB Endowment, 2004.

[9] T. Haveliwala. Efficient computation of pageRank. Technical Report, Stanford University, 1999.

[10] T. Haveliwala and S. Kamvar. The second eigenvalue of the google matrix. Technical Report, Stanford University, 2003.

[11] T. Haveliwala, S. Kamvar, and G. Jeh. An analytical comparison of approaches to personalizing pagerank. Technical Report, Stanford University, 2003.

[12] T. H. Haveliwala. Topic-sensitive pagerank. In *WWW ' 02*, Honolulu, Hawaii, May 2002.

[13] K. Jarvelin and J. Kekalainen. IR evaluation methods for retrieving highly relevant documents. In *SIGIR '00*, pages 41–48, New York, USA, 2000. ACM.

[14] K. Jarvelin and J. Kekalainen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[15] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. In *SODA '98*, pages 668–677, Philadelphia, PA, USA, 1998. Society for Industrial and Applied Mathematics.

[16] A. N. Langville and C. D. Meyer. Deeper inside pagerank. *Internet Mathematics*, 1(3):335–400, 2004.

[17] F. McSherry. A uniform approach to accelerated pagerank computation. In *WWW '05*, pages 575–582, New York, USA, 2005. ACM.

[18] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[19] J. A. Rice. *Mathematical Statistics and Data Analysis (2nd ed.)*. Duxbery Press, 1995.

[20] M. Richardson and P. Domingos. The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2002.

[21] S. E. Robertson. Overview of okapi projects. *Journal of Documentatioin*, 53(1):3–7, 1997.

[22] W. J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, Princeton, N.J., 1994.

[23] Z. K. Wang and X. Q. Yang. *Birth and Death Processes and Markov Chains*. Springer-Verlag, New York, 1992.

[24] R. W. White, M. Bilenko, and S. Cucerzan. Studying the use of popular destinations to enhance web search interaction. In *SIGIR '07*, pages 159–166, New York, USA, 2007. ACM.

APPENDIX

Proof of Theorem 2

We prove the theorem by showing that there is a directed path in the transition graph between any two pages. As for the user browsing graph, we have the following three observations:

1) For the reset probability, we have $\sigma_i \geq 0$,for any $i = 1, 2, \ldots, N$. Suppose that among them there are T entries that are strictly positive. Without loss of generality, we regard the first T entries as such, i.e., $\sigma_i > 0$,for any $i = 1, 2, \ldots, T$. In other words, all sessions begin with the first T pages.

2) Since all the pages in the state space V come from the user behavior data, any page will belong to a session.

3) A session corresponds to a path in the transition graph: from the first page in a session, there is a path to all the other pages in the same session.

For any $i, j \in V$, based on the second observation, we assume that page i is in the session whose first page is b_i, and page j is in the session whose first page is b_j.

If page i is the last page in the session, then according to the reset rule and the first observation, there will be a transition path from it to page b_j with the probability $\sigma_{b_j} > 0$. Then based on the third observation, a path can be found from page b_j to page j. As a result, there is a directed path from page i to page j.

If page i is not the last page in its session, based on the third observation, there will be a path from i to the last page in that session. Since there is a path from the last page in that session to page j, as proved above, we actually can come to the conclusion that there is a direct path from page i to page j.