Inf Retrieval
DOI 10.1007/s10791-011-9164-x

Page importance computation based on Markov processes

Bin Gao · Tie-Yan Liu · Yuting Liu · Taifeng Wang · Zhi-Ming Ma · Hang Li

Received: 13 January 2010 / Accepted: 17 February 2011

Abstract This paper is concerned with Markov processes for computing page importance. Page importance is a key factor in Web search. Many algorithms such as PageRank and its variations have been proposed for computing the quantity in different scenarios, using different data sources, and with different assumptions. Then a question arises, as to whether these algorithms can be explained in a unified way, and whether there is a general guideline to design new algorithms for new scenarios. In order to answer these questions, we introduce a *General Markov Framework* in this paper. Under the framework, a *Web Markov Skeleton Process* is used to model the random walk conducted by the web surfer on a given graph. Page importance is then defined as the product of two factors: *page reachability*, the average possibility that the surfer arrives at the page, and *page utility*, the

A short version of this paper, titled *A General Markov Framework for Page Importance Computation*, was accepted as a short paper in the 18th ACM Conference on Information and Knowledge Management (CIKM'09).

B. Gao (✉) · T.-Y. Liu · T. Wang · H. Li
Microsoft Research Asia, Sigma Center, No.49, Zhichun Road, Haidian District, Beijing 100190, People's Republic of China
e-mail: bingao@microsoft.com

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

T. Wang
e-mail: taifengw@microsoft.com

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

Y. Liu
Beijing Jiaotong University, No.3, Shangyuan Residence, Haidian District, Beijing 100044, People's Republic of China
e-mail: ytliu@bjtu.edu.cn

Z.-M. Ma
Academy of Mathematics and Systems Science, Chinese Academy of Sciences, No.55, Zhongguancun East Road, Haidian District, Beijing 100190, People's Republic of China
e-mail: mazm@amt.ac.cn

Published online: 03 March 2011

average value that the page gives to the surfer in a single visit. These two factors can be computed as the stationary probability distribution of the corresponding embedded Markov chain and the mean staying time on each page of the Web Markov Skeleton Process respectively. We show that this general framework can cover many existing algorithms including PageRank, TrustRank, and BrowseRank as its special cases. We also show that the framework can help us design new algorithms to handle more complex problems, by constructing graphs from new data sources, employing new family members of the Web Markov Skeleton Process, and using new methods to estimate these two factors. In particular, we demonstrate the use of the framework with the exploitation of a new process, named *Mirror Semi-Markov Process*. In the new process, the staying time on a page, as a random variable, is assumed to be dependent on both the current page and its inlink pages. Our experimental results on both the user browsing graph and the mobile web graph validate that the Mirror Semi-Markov Process is more effective than previous models in several tasks, even when there are web spams and when the assumption on preferential attachment does not hold.

Keywords Page importance · PageRank · BrowseRank · Web Markov skeleton process · Mirror semi-Markov process

1 Introduction

Page importance is a critical factor in web search, since it plays a key role in crawling of Web pages, indexing of the crawled pages, and ranking of the indexed pages. Many effective algorithms have been proposed to compute page importance in the literature, such as PageRank (Brin and Page 1998; Page et al. 1998) and TrustRank (Gyongyi et al. 2004).

PageRank and its variations are usually modeled as a discrete-time Markov process on the web link graph for page importance computation, the process of which actually simulates a random walk of a surfer along the hyperlinks on the web. By applying the above algorithms, people have solved many critical problems in Web search. However, these algorithms also have certain limitations as in the modeling for representing page importance. For example, PageRank only models a random walk on the graph, but does not consider the lengths of time that the web surfer spends on the pages during the browsing process. The staying time information can be used as good indicators of the page quality, which is highly related to the importance of the pages.

To solve this problem, BrowseRank (Liu et al. 2008) was proposed to leverage the user staying time information on webpages for page importance computation. It collects the user behavior data in web surfing and builds a user browsing graph, which contains both user transition information and user staying time information. A continuous-time Markov process is employed in BrowseRank to model the browsing behaviors of a web surfer, and the stationary distribution of the process is regarded as the page importance scores.

However, there are still challenges that cannot be well handled by the aforementioned algorithms. Here, we give two examples.

– In some new scenarios, the assumptions in existing algorithms may not hold. For example, in the mobile Web, owing to the specific business model, the owner of a website tends to create more hyperlinks to the pages of his own or his partners, than those of other websites. As a result, the topological property of the mobile web graph is significantly different from the general web (Jindal et al. 2008). There are more disconnected components in the

mobile web graph, and many links in it are not preferential attachments, but profit-oriented attachments. In this case, the page importance computed by algorithms like PageRank may not reflect the true importance of the pages.

– In some existing applications, the assumptions in existing algorithms may not be accurate either. Take importance calculation with the user browsing graph in BrowseRank (Liu et al. 2008) as an example. Basically BrowseRank trusts the user behavior data, and estimates the page importance from it. However, when there are click frauds, the data may not be trustworthy. Suppose a webmaster puts an online advertisement on his homepage. In order to earn money, he may artificially click the link of the advertisement. Sometimes a robot is even used to increase the frequency of clicks. As a result, we will observe a large volume of transitions from his homepage to the advertisement[1]. If we do not distinguish the sources of the transitions when estimating the staying time on the advertisement page, the estimation may be highly biased by these fraudulent clicks. In this case, the page importance computed by BrowseRank may not be accurate.

With these challenges lying ahead (not limited to the ones mentioned above), it may be necessary to develop new technologies to address the problems in both existing and new scenarios. For this purpose, it is helpful to study whether there is a common theory behind the existing algorithms, and whether the theory can lead to some general guidelines for designing new algorithms. This is just the motivation of this paper.

Considering that most previous works are based on random walks on a graph, and employ Markov processes in their mathematical modeling, we propose using a general Markov framework as the unified description of these algorithms. In the framework, we consider how to model page importance from the viewpoint of random surfer, assuming that there is a web link graph or a user browsing graph available. In this setting, the importance of a page means the value that the page can eventually provide to the random surfer. It can be considered that there are two factors that affect page importance: *page reachability* and *page utility*. The former represents the possibility that the surfer arrives at the page and the latter represents the value of the page given to the surfer in a visit. *Web Markov Skeleton Processes* can represent these two factors with the stationary probability distribution of their embedded Markov chain (EMC)[2] and the mean staying time. Specifically, the larger the stationary distribution of the EMC of a page is, the higher reachability the page has; the longer mean staying time a page retains, the higher utility the page provides. We can then take the product of stationary distribution of the EMC and the mean staying time as the page importance. In many cases, it can be proved that the product is proportional to the stationary distribution of the Web Markov Skeleton Process itself, if it exists.

Existing algorithms such as PageRank and BrowseRank can be well covered by the framework. For example, PageRank is based on the *Discrete-Time Markov Process* which is a special Web Markov Skeleton Process. BrowseRank is based on the *Continuous-Time Markov Process*, another special Web Markov Skeleton Process.

Furthermore, the general framework also provides us a guideline of designing new algorithms. We can attain new methods by defining the graph on new data sources,

[1] According to recent study, US companies paid a record $14.2 billion for paid keyword-driven contextual ads in 2009. Meanwhile, click fraud rates rose to 17.4 to 29.4% in the first 3 months of 2010. That is, we will observe a great number of transitions from the content pages (e.g., blogs and forums) to the landing pages of the click fraud ads. cf. http://lastwatchdog.com/botnet-driven-click-fraud-steals-millions-advertisers/

[2] It can also be called as the skeleton process of the Web Markov Skeleton Process.

employing new family members of the Web Markov Skeleton Process, or developing new methods to estimate the stationary distribution of the skeleton process and the mean staying time. To demonstrate the use of this framework, we propose employing a new process, which we call *Mirror Semi-Markov Process*. In the new process, the staying time on one page depends on not only this page but also the previous pages visited by the surfer. By doing so, we can address the aforementioned issues that existing algorithms suffer from. As for the mobile web graph, if we penalize the estimated staying time on a page when the transition is from the same website or a partner website, then the corresponding page utility will be decreased and the problem with the profit-oriented attachment will be tackled. And in the scenario of BrowseRank, if we assume that the distribution of staying time on one page varies when the transitions are from different websites, and perform some normalization among different source websites, then the calculated page importance will more accurately reflect the underlying truth even if there is spam or click fraud in the data.

We tested the Mirror Semi-Markov Process and the corresponding algorithms on both the mobile web graph and the user browsing graph. The experimental results show that the new algorithms can outperform existing methods in several tasks such as top ranked page finding and spam/junk page filtering. This well validates the effectiveness of the proposed framework.

To sum up, the proposed general framework for page importance computation has the following characteristics:

- It can accurately represent the importance of web pages.
- It can well explain existing models and even include existing models as special cases.
- It has solid theoretical background.
- It can effectively guide the development of new algorithms. Such algorithms can deal with the problems that existing algorithms cannot handle.

The rest of the paper is organized as follows. Section 2 makes a survey on related work. Section 3 describes the general Markov framework built on the Web Markov Skeleton Process. Section 4 explains how PageRank is generalized to BrowseRank. Section 5 explains how BrowseRank is generalized to BrowseRank Plus, in which the Mirror Semi-Markov Process method is introduced. Section 6 makes more discussions on the Mirror Semi-Markov Process, while Sect. 7 makes more discussions on the general Markov framework. The experimental results are reported in Sect. 8. Section 9 makes discussions on the differences between the proposed framework and several other link analysis frameworks. Conclusion and future work are given in Sect. 10.

2 Related work

PageRank (Brin and Page 1998; Page et al. 1998) and its variants (Boldi et al. 2005; Haveliwala 1999; Haveliwala and Kamvar 2003; Haveliwala et al. 2003; Haveliwala 2002; Langville and Meyer 2004; McSherry 2005; Richardson and Domingos 2002) compute page importance by taking the web as a graph of pages connected with hyperlinks. The approach makes an assumption that the web link graph is used for the random surfer to carry out a random walk. A Discrete-Time Markov Process is employed to model the random walk, and the stationary distribution of the process is used as page importance. Other work such as (Bianchini et al. 2005; Kleinberg 1998) is also based on the random walk on a web link graph.

As PageRank can be easily spammed by tricks like link farms (Gyongyi and Garcia-Molina 2004), some robust algorithms against link spam have been proposed. For instance, TrustRank (Gyongyi et al. 2004) takes into consideration the reliability of webpages when calculating the page importance. In this approach, a set of reliable pages are identified as seed pages at first. Then the trust scores of the seed pages are propagated to other pages along links on the web graph. As the propagation starts from the reliable pages, TrustRank can be more immune to spam than PageRank.

BrowseRank (Liu et al. 2008) is a recently proposed method, which computes the page importance using user behavior data. Specifically, a user browsing graph is constructed from the web browsing history of users. The browsing graph contains richer information such as staying times on web pages by users. BrowseRank assumes that the more frequently users click on a page and the longer time the users stay on it, the more likely the page is important. A Continuous-Time Markov Process is employed to model the random walk on the user browsing graph. BrowseRank then takes the stationary distribution of the process as page importance. Other work such as (Berberich et al. 2004; Yu et al. 2005) also exploits time information to compute page importance. In the T-Rank algorithm (Berberich et al. 2004), freshness was defined to represent the timestamps of most recent updates of pages and links, and then freshness and its update rate were used to adjust the random walk and the resulting Markov chain in PageRank computation. In the Timed-PageRank algorithm (Yu et al. 2005), the inlinks of a Web page are assigned with different weights according to their creation time. A decaying rate was defined so that the latest links would get the highest weights. In another word, link information from different snapshots of graphs is compressed in one web graph, and then the PageRank algorithm is implemented on it. Therefore, both of the above works can be regarded as extensions of the weighted PageRank algorithm, which can also be covered by our proposed framework.

3 General framework

We first consider key factors for computing page importance, and then we introduce the Web Markov Skeleton Process and describe our framework of using it to represent the factors.

3.1 Page importance factors

We assume that there is a web surfer performing a random walk on the web graph. By random walk here we mean a trajectory consisting of successive random steps on the nodes of a graph. The web graph can be web link graph or user browsing graph. In the former, each node in the graph represents a web page, and each edge represents a hyperlink between two pages. In the latter, each node in the graph stands for a web page, and each edge stands for a transition between pages. The transition information can be obtained by aggregating behavior data of billions of web users (Liu et al. 2008). In that sense, the random surfer is a persona combining the characteristics of all the web users[3].

The importance of a page can be viewed as the average value that the page provides to the random surfers during the surfing process. Note that a visit of a page by a single surfer is random, and the value which a page can offer to the surfer in one visit is also random.

[3] In some algorithms, the random surfer sometimes does not follow the edges but performs random resets. In such case, we regard the graph as containing virtual edges corresponding to the resets.

Therefore, there are two intuitive factors that can affect page importance.

- Page reachability: the (average) possibility that a surfer arrives at the page.
- Page utility: the (average) value that the page gives to a surfer in a single visit.

Page reachability is mainly determined by the structure of graph. For example, in a web link graph, if a page has a large number of inlinks, it is likely to be more frequently visited. Page utility can be affected by several things, for example, the content of the page, the pages the surfer visited before. That is, page utility may depend on not only the current page but also other related pages.

3.2 Web Markov Skeleton process

Intuitively, a Web Markov Skeleton Process (WMSP) is a stochastic process which contains a Markov chain as its skeleton. It has been proposed and studied in probability theory (Hou and Liu 2005; Hou et al. 1998) and applied to many fields including queueing theory and reliability engineering.

3.2.1 An intuitive definition

A WMSP is a stochastic process Z defined as follows. Note that we try to provide an intuitive definition here. A more rigorous definition can be found in Sect. 3.2.2. Suppose that X is a Markov Chain with state space S and transition probability matrix P. Let $X_0, X_1,\ldots, X_n,\ldots$ denote a sequence of X, where X_n is a state and X_{n+1} is determined by the probability distribution $P(X_{n+1}|X_n)$, $(n = 0, 1,\ldots)$. Further suppose that Y is a stochastic process on the positive real-number set $\mathbf{R}^+$. Let $Y_0, Y_1,\ldots, Y_n,\ldots$ denote a sequence of Y, where Y_n, $(n = 0, 1,\ldots)$ is a positive real-number. Suppose that there are $S_0, S_1,\ldots, S_n,\ldots$, and $S_n \subseteq S, (n = 0, 1, \ldots)$. Y is determined by the probability distribution $P(Y_n|S_n)$, $n = 0, 1,\ldots$. Then, a Markov Skeleton Process Z is a Stochastic Process based on X and Y. A sequence of Z can be represented as

$$X_0 \xrightarrow{Y_0} X_1 \xrightarrow{Y_1} \cdots X_n \xrightarrow{Y_n} \cdots$$

where X_n denotes a state and Y_n denotes staying time at state $X_n, (n = 0, 1, \ldots)$. X_n depends on X_{n-1} and Y_n depends on multiple states S_n $(n = 1, 2, \ldots)$.

Many existing stochastic processes are special cases of WMSP. We just list a few of them as examples.

- Discrete-Time Markov Process: when Y_n is constant, then Z is called a *Discrete-Time Markov Process*.
- Continuous-Time Markov Process: when Y_n only depends on X_n following an exponential distribution $P(Y_n|X_n)$, then Z is called a *Continuous-Time Markov Process*.
- Semi-Markov Process: when Y_n only depends on X_n and X_{n+1} according to distribution $P(Y_n|X_n, X_{n+1})$, then Z is called a *Semi-Markov Process*.

Furthermore, Mirror Semi-Markov Process, proposed in Sect. 5.1.1, is also a special case of MSP.

- Mirror Semi-Markov Process: when Y_n only depends on X_n and X_{n-1} according to distribution $P(Y_n|X_n, X_{n-1})$, then Z is called a *Mirror Semi-Markov Process*.

3.2.2 A rigorous definition in mathematics

After an intuitive explanation, we would like to give a rigorous definition on WMSP.

Definition 1 A stochastic process $Z = \{Z(t),\ t \geq 0\}$ with life time ζ is called a *Markov Skeleton Process* if there exists a sequence of random times $\{\tau_n\}_{n\geq 0}$ such that $\tau_0 = 0, \tau_1 < \cdots < \tau_n < \tau_{n+1} < \cdots < \zeta, \lim_{n\to\infty} \tau_n = \zeta$, and $\{Z(\tau_n)\}_{n\geq 0}$ forms a Markov chain.

In probability theory, MSPs have been intensively studied and applied to queuing theory, reliability engineering, and other related fields (Hou and Liu 2005; Hou et al. 1998). Note that in (Hou and Liu 2005; Hou et al. 1998) the definition of MSP is slightly more narrow than ours, where it was required (among others) that for each $n \geq 0$, the future development of Z (namely, $Z(\tau_n + \cdot)$) given the information of $Z(\tau_n)$ should be conditionally independent of the history of Z prior to τ_n (see e.g. (Hou and Liu 2005) Definition 1.1.1).

Let Z be a Markov Skeleton Process. We denote by $X_n = Z(\tau_n)$, and $Y_n = \tau_{n+1} - \tau_n$, for all $n = 0, 1, \ldots$. Then $X = \{X_n,\ n \geq 0\}$ is a Markov chain and $Y = \{Y_n,\ n \geq 0\}$ is a sequence of positive random variables. Clearly the pair of (X, Y) is uniquely determined by the MSP Z. Conversely, if $Z(t) = X(\tau_n)$ for $\tau_n \leq t < \tau_{n+1,}$ then Z is also uniquely determined by (X, Y). Specifically, suppose that we are given a Markov chain $X = \{X_n, n \geq 0\}$ and a sequence of positive random variables $Y = \{Y_n,\ n \geq 0\}$. Then a MSP Z satisfying $Z(t) = X(\tau_n)$ for $\tau_n \leq t < \tau_{n+1}$ can be uniquely determined by the following regime.

$$X_0 \overset{Y_0}{\to} X_1 \overset{Y_1}{\to} \cdots X_n \overset{Y_n}{\to} \cdots \tag{1}$$

More precisely, a MSP Z with the above specified property can be uniquely determined by the pair of (X, Y) by setting $\tau_n = \sum_{k=0}^{n} Y_k$ and setting $Z(t) = X_n$ if $\tau_n \leq t < \tau_{n+1}$. For ease of reference, in the future we shall write $Z = (X, Y)$ if a MSP Z is determined by (X,Y) with the above regime. Further we introduce the following definition.

Definition 2 Let $Z = (X, Y)$ be a Markov Skeleton Process described by the regime (1). Then X is called the *Embedded Markov Chain* (EMC) of Z.

For our purpose of studying the random walk on web link graph or user browsing graph, we shall focus on a special class of Markov skeleton processes defined below.

Definition 3 Let $Z = (X, Y)$ be a Markov Skeleton Process on S determined by the regime (1). Suppose that: (i) the state space S is finite or discrete, (ii) the embedded Markov chain $X = \{X_n, n \geq 0\}$ is time-homogeneous, (iii) given the information of X, the positive random variables $\{Y_n\}_{n\geq 0}$ are conditionally independent to each other. Then the process Z is called a *Web Markov Skeleton Process* (WMSP).

Note that when S is finite or discrete, then $X = \{X_n,\ n \geq 0\}$ is a Markov chain if and only if for all $n \geq 0, P(X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, \ldots, X_0 = i_0) = P(X_{n+1} = j | X_n = i)$; and the time-homogeneity of X means that the probability of single-step transition is independent of n, i.e., for any n, $P(X_{n+1} = j | X_n = i) = P(X_1 = j | X_0 = i) \triangleq p_{ij},\ \forall i, j \in S$. Note also that the conditional independency of $\{Y_n\}_{n\geq 0}$ given X means that $Y = \{Y_n,\ n \geq 0\}$ is determined by the conditional probability distribution $P(Y_n \leq t | \{X_k\}_{k\geq 0})$, $n = 0, 1, \ldots$. In various applications, we may assume further that for each n, there exists a subset S_n of $\{X_k\}_{k\geq 0}$ such that $P(Y_n \leq t | \{X_k\}_{k\geq 0}) = P(Y_n \leq t | S_n)$.

Let $Z = (X, Y)$ be a WMSP defined as above, then intuitively X_n denotes a state and Y_n denotes staying time at state X_n for each n. By the above explanation, X_n depends on X_{n-1}, and Y_n depends on multiple states S_n of $\{X_k\}_{k\geq 0}$. Our model of WMSP covers all the known stochastic processes used in studying web page importance (this is why we call it Web MSP). Below we just list a few of them as examples.

- *Example 1*. Discrete-Time Markov Process: a WMSP $Z = (X, Y)$ such that Y_n is constant, i.e., $P(Y_n = 1) = 1$ for all n.
- *Example 2*. Continuous-Time Markov Process: a WMSP $Z = (X, Y)$ such that Y_n depends only on X_n with exponential distributions which are independent of n, i.e., $P(Y_n \leq t|\{X_k\}_{k\geq 0}) = P(Y_n \leq t|\{X_n\}) = P(Y_0 \leq t|\{X_0\})$ for all n.
- *Example 3*. Semi-Markov Process: a WMSP $Z = (X, Y)$ such that Y_n depends only on X_n and X_{n+1}, and its distributions are independent of n, i.e., $P(Y_n \leq t|\{X_k\}_{k\geq 0}) = P(Y_n \leq t|\{X_n, X_{n+1}\}) = P(Y_0 \leq t|\{X_0, X_1\})$ for all n.

Furthermore, Mirror Semi-Markov Process, which will be proposed later in this paper, is also a special case of WMSP. Below we list only its short description, for more details see Sect. 5.1.1.

- *Example 4*. Mirror Semi-Markov Process: a WMSP $Z = (X, Y)$ such that Y_n depends only on X_n and X_{n-1}, and its distributions are independent of n, i.e., $P(Y_0 \leq t|\{X_k\}_{k\geq 0}) = P(Y_0 \leq t|\{X_0\})$, and $P(Y_n \leq t|\{X_k\}_{k\geq 0}) = P(Y_n \leq t|\{X_n, X_{n-1}\}) = P(Y_1 \leq t|\{X_1, X_0\})$ for all $n \geq 1$.

Note that WMSP is *different* from the high-order Markov process. In general, in a high-order Markov chain $X_0, X_1, \ldots, X_n, \ldots$, state X_n depends on several previous states, and there is no process Y. For example, in a two-order Markov chain, X_{n+2} is determined by the probability distribution $P(X_{n+2}|\{X_{n+1}, X_n\}), (n = 0, 1, \ldots)$. The discussion in this paper is not related to the high-order Markov process.

3.3 Modeling page importance

Since WMSP can naturally model the random walk of web surfer, in the rest of this paper we shall consider only the framework of WMSP. Let $Z = (X, Y)$ be a WMSP on a state space S. In our model the state space S corresponds to Web pages, and process Z represents a random walk on pages. The surfer randomly chooses the next page to visit based on the current page, according to X. He/she further randomly decides the length of staying time on the current page based on the page, and several other pages he/she visited before, and/or several other pages he/she will visit, according to Y.

Therefore, the aforementioned two factors are characterized by the two quantities in WMSP. Specifically,

- Stationary distribution of X (which exists under some conditions) represents page reachability.
- Mean staying time represents page utility.

Furthermore, given a page, we can define page importance as the product of the value of the stationary distribution of X on the page and the mean staying time on it. Note that there might be other forms for page importance besides the product defined above. The reason we use this definition is that in many cases the product is proportional to the limiting probability distribution of the Web Markov Skeleton Process if it exists.

WMSP is a very general model, which does not make any specific assumption on the distribution $P(Y_n \leq t|S_n)$. That is, staying time can be assumed to depend on a large number of other states. Therefore, there is a large room for people to develop new algorithms based on this class of general processes.

Note that the proposed framework is mainly intended for entity importance computation in graphs. Though link analysis algorithms like PageRank can be used for different purposes and in different graphs such as social networks, entity-relation graphs, and query graphs, we focus on entity importance computation and the general Markov framework in this paper.

4 From PageRank to BrowseRank

The proposed framework is suitable for web page importance computation and is theoretically sound, as described above. In addition, it can cover and explain many existing algorithms.

4.1 PageRank and its variations

In PageRank, a Discrete-Time Markov Process is used to model the random walk on the web link graph. As being illustrated by Example 1 of Sect. 3.2, this is a special WMSP.

According to Definition 2, we see that the embedded Markov chain X of a Discrete-Time Markov Process is equivalent to the Discrete-Time Markov chain Z. Under some conditions, EMC X has a unique stationary distribution denoted by $\tilde{\pi}$, which satisfies

$$\tilde{\pi} = \tilde{\pi}P. \tag{2}$$

In PageRank, page reachability is computed by the above stationary distribution $\tilde{\pi}$, and page utility is set to one for all pages under the assumption that all pages are equally useful for the random surfer, which is indicated by $P(Y_n = 1) = 1$.

As TrustRank is a modified PageRank by starting the iterative process from a reliable seed set, it can also be covered by the framework. Similarly, other PageRank-alike approaches such as (Nie et al. 2005; Poblete et al. 2008) can also be regarded as special cases of the framework.

4.2 BrowseRank

In BrowseRank, the random walk is defined on the user browsing graph, and a Continuous-Time Markov Process is employed. According to Example 2 of Sect. 3.2, this is also a special WMSP,

Denote T_j for the random variable of staying time on page j, and as Y_n is a random variable following an exponential distribution $P(Y_n \leq t|\{X_n\})$, we can get,

$$F_{T_j}(t) \triangleq P(T_j \leq t) = P(Y_n \leq t|X_n = j) = 1 - e^{-\lambda_j t}, t \geq 0. \tag{3}$$

Here λ_j is the parameter for the exponential distribution, $F_{T_j}(t)$ is the cumulative probability distribution of random variable T_j. The mean staying time on page j is calculated as,

$$\begin{aligned}\tilde{T}_j \triangleq E(T_j) &= \int_0^\infty t F_{T_j}(\mathrm{d}t) \\ &= \int_0^\infty t\lambda_j e^{-\lambda_j t}\mathrm{d}t \\ &= \int_0^\infty (-t)\mathrm{d}e^{-\lambda_j t} \\ &= (-te^{-\lambda_j t})\,|_0^\infty + \int_0^\infty e^{-\lambda_j t}\mathrm{d}t \\ &= -\frac{1}{\lambda_j}\int_0^\infty \mathrm{d}e^{-\lambda_j t} \\ &= -1\lambda_j e^{-\lambda_j t}\,|_0^\infty \\ &= 1\lambda_j.\end{aligned} \tag{4}$$

By Definition 2, we see that X is the embedded Markov chain of the Continuous-Time Markov Process, and its unique stationary distribution $\tilde{\pi}$ can also be calculated as (2).

In BrowseRank, it is equivalent to defining page reachability as the stationary probability distribution of the Embedded Markov Chain of the Continuous-Time Markov Process, and defining page utility as the mean staying time on the page. According to Sect. 3.3, we can define the following distribution to compute the BrowseRank scores, using $\tilde{\pi}$ and $\tilde{T}_j$, where α is a normalized coefficient.

$$\pi_j = \alpha\tilde{\pi}_j\tilde{T}_j. \tag{5}$$

It is not difficult to see that the above distribution is proportional to the limiting probability distribution of the Continuous-Time Markov Process.

5 From BrowseRank to BrowseRank plus

The framework can also help devise new algorithms. Particularly the dependency of staying time on multiple states can be leveraged to address issues that existing algorithms cannot cope with. Here we take the extension from BrowseRank to BrowseRank Plus[4] as an example to show how to design the computing mechanism under the guide of the framework.

As discussed in Sect. 1, spam is pervasive on the web and how to effectively eliminate it is an issue we must consider when calculating page importance, especially for algorithms like BrowseRank that are based on user browsing graph. As discussed before, BrowseRank builds a Continuous-Time Markov Process model based on user browsing graph. Click fraud is the biggest challenge for BrowseRank. The reason is that BrowseRank fully trusts the user behavior data, and calculates the mean staying time directly from user behavior data. This leaves a big room for web spammers to conduct click fraud by manually or automatically clicking the pages they want to boost from their websites. As a result, we will observe a large

[4] Note that we briefly discussed BrowseRank Plus in a short paper (Gao et al. 2009).

volume of transitions to the spam pages from their websites. A straightforward application of BrowseRank will result in a heavily biased page importance calculation.

To solve the problem, we may treat the transitions from different inlink websites by different weights. If a big portion of transitions are from a specific inlink website, we may downgrade the weight of each of these transitions, so as to reduce the influence from click fraud. However, in this situation, the Continuous-Time Markov Process would not work well for it does distinguish from which websites the transitions come. Therefore, we need to design a new and specific model of WMSP, which is suitable to deal with the above task. We call the new model of WMSP as Mirror Semi-Markov Process (MSMP), and refer to the new algorithm derived from it as BrowseRank Plus. The major advantage of BrowseRank Plus is that it can deal with click fraud, which the original BrowseRank algorithm may suffer from. Note that the idea of generalizing BrowseRank to BrowseRank Plus is somehow similar to SimRank(Jeh and Widom 2002).

5.1 Mirror Semi-Markov Process method

To better explain the BrowseRank Plus algorithm, we define the Mirror Semi-Markov Process and introduce an implementation of it for page importance computation.

5.1.1 MSMP model

Mirror Semi-Markov Process (MSMP) is a new stochastic process that can be used in not only web search but also other applications.

Definition 4 (cf. Example 4 of Sect. 3.2) A Web Markov Skeleton Process $Z = (X, Y)$ is called a *Mirror Semi-Markov Process* (MSMP), if Y_n depends only on X_n and X_{n-1}, and the distributions of Y_n are independent of n. More precisely, a WMSP $Z = (X, Y)$ is called a MSMP, if $P(Y_0 \le t|\{X_k\}_{k\ge 0}) = P(Y_0 \le t|\{X_0\})$, and $P(Y_n \le t|\{X_k\}_{k\ge 0}) = P(Y_n \le t|X_n, X_{n-1}\}) = P(Y_1 \le t|\{X_1, X_0\})$ for all $n \ge 1$.

Obviously MSMP is a special case of Web Markov Skeleton Process. Furthermore, MSMP is similar to the Semi-Markov Process (see Example 3 of Sect. 3.2). In a Semi-Markov Process, Y_n depends on the current state X_n and the next state X_{n+1}, while in MSMP Y_n depends on the current state X_n and the previous state X_{n-1}. The dependencies are in two opposite directions. That is why we call the new model Mirror Semi-Markov Process.

Let X be the Embedded Markov Chain of a MSMP Z, as defined in Sect. 3.2. In what follows we assume that the state space S is finite, which represents all Web pages. We use the same denotation $P = (p_{ij})_{i,j\in S}$ to represent the transition probability matrix of process X. Assume further that there exists a unique stationary distribution of process X, which will be denoted again by $\tilde{\pi}$. Then $\tilde{\pi}$ satisfies also the (2), and can be calculated by the power method (Golub and Loan 1996).

As in Sect. 4.2, we use j to represent a state ($j \in S$) and use T_j to represent staying time on state j. By Definition 4 we can get the following equation:

$$P(Y_n \le t|X_{n-1} = i, X_n = j) = P(Y_1 \le t|X_0 = i, X_1 = j) = P(T_j \le t|X_0 = i) \qquad (6)$$

We use ${}_iF_{T_j}(t) \triangleq P(T_j \le t|X_0 = i)$ to denote the cumulative probability distribution of staying time on state j from state i, and use $F_{T_j}(t)$ to represent the the cumulative probability distribution of staying time on state j. It is easy to get the following result:

$$\begin{aligned} F_{T_j}(t) &= P(T_j \le t) = P(Y_1 \le t | X_1 = j) \\ &= \sum_{i \in S} P(Y_1 \le t | X_1 = j, X_0 = i) P(X_0 = i | X_1 = j) \\ &= \sum_{i \in S} {}_iF_{T_j}(t) P(X_0 = i | X_1 = j) \end{aligned} \tag{7}$$

For easy of reference, we have the following definition of Contribution Probability.

Definition 5 We define $c_{ij} \triangleq P(X_0 = i | X_1 = j)$, and call it the contribution probability of state i to state j.

Owing to the time-homogeneity of EMC X, we can find that contribution probability c_{ij} varies with two successive states i and j no matter which time of the jump from i to j occurred.

Based on the definition of contribution probability and cumulative probability distribution of staying time T_j, we can calculate the mean staying time on state j as follows.

$$\begin{aligned} \tilde{T}_j \triangleq E(T_j) &= \int_0^\infty t F_{T_j}(\mathrm{d}t) \\ &= \sum_{i \in S} c_{ij} \int_0^\infty t \times {}_iF_{T_j}(\mathrm{d}t). \end{aligned} \tag{8}$$

We further calculate the following distribution, which is defined as the page importance score as mentioned before, by using of the stationary distribution $\tilde{\pi}$ of EMC X and mean staying time $\tilde{T}_j$, where α is a normalized coefficient to make equation $\sum_{j \in S} \pi_j = 1$ true.

$$\pi_j = \alpha \tilde{\pi}_j \tilde{T}_j, \quad \forall j \in S. \tag{9}$$

5.1.2 Implementation of MSMP

As explained above, we can apply MSMP to page importance computation. Given a web graph and its metadata, we build an MSMP model on the graph. We first estimate the stationary distribution of EMC X. We next compute the mean staying time using the metadata. Finally, we calculate the product of the stationary distribution of the EMC and the mean staying time on pages, which we regard as page importance. As the stationary distribution of the EMC can be conveniently computed by power method (Golub and Loan 1996), we will focus on the staying time calculation in the next subsection.

From (8), we obtain that the mean staying time is determined by two parts: contribution probability, and cumulative probability distribution from previous state. Hereafter, we analyze these two quantities in details.

5.1.2.1 Contribution probability Suppose that for page j there are n_j pages linked to it: $\Xi_j = \{\xi_{j1}, \xi_{j2}, \ldots, \xi_{jn_j}\} \subseteq S$. In fact, contribution probability from i to j is the probability that the surfer comes from page i when given the condition that he/she has come to page j, then we can easily obtain the following proposition.

Proposition 1 *Suppose c_{ij} is the contribution probability from i to j, then*

(i)

$$\sum_{i\in S} c_{ij} = \sum_{i\in \Xi_j} c_{ij} = 1. \tag{10}$$

(ii)

$$c_{ij} = \frac{p_{ij}\tilde{\pi}_i}{\tilde{\pi}_j}. \tag{11}$$

where p_{ij} is the transition probability from i to j according to the transition probability matrix of process X, and $\tilde{\pi}$ is the stationary distribution of process X.

Proof

(i) It is easy to get the following deduction:

$$\begin{aligned}\sum_{i\in S} c_{ij} = \sum_{i\in S} P(X_0 = i|X_1 = j) &= \frac{\sum_{i\in S} P(X_0 = i, X_1 = j)}{P(X_1 = j)} = \frac{P(X_1 = j)}{P(X_1 = j)} = 1 \\ &= \frac{\sum_{i\in \Xi_j} P(X_0 = i, X_1 = j)}{P(X_1 = j)} = \sum_{i\in \Xi_j} c_{ij}\end{aligned}$$

(ii) First, due to the time-homogeneity of process X, we get

$$c_{ij} = \frac{P(X_{n+1} = j|X_n = i)P(X_n = i)}{P(X_{n+1} = j)} = \frac{p_{ij}P(X_n = i)}{P(X_{n+1} = j)};$$

Second, process X has stationary distribution, that is, its limiting probability distribution exists, then, we calculate limit on the above equation,

$$\begin{aligned}c_{ij} &= \lim_{n\to+\infty} \frac{p_{ij}P(X_n = i)}{P(X_{n+1} = j)} \\ &= \frac{p_{ij}\tilde{\pi}_i}{\tilde{\pi}_j}.\end{aligned}$$

□

From (11), we can easily calculate the contribution probability. In this paper, we use another heuristic method to compute such probability as a demonstration. Because in this paper, we just want to show the impact from different websites on the calculation of page importance.

Suppose that the n_j inlinks of page j are from m_j websites, and from website k ($k = 1, \ldots, m_j$) there are n_{jk} inlinks. Thus we have,

$$n_j = \sum_{k=1}^{m_j} n_{jk}. \tag{12}$$

Note that the website which page j belongs to might also exist in the m_j websites.

Suppose that the m_j sites that linked to page j are: $\Phi_j = \{\phi_{j1}, \phi_{j2}, \ldots, \phi_{jm_j}\}$. $c_{\phi_{jk}j}$ is the probability that the surfer comes to page j from site ϕ_{jk}, referred to as contribution probability of site.

$$c_{\phi_{jk}j} = \sum_{l=1}^{n_{jk}} c_{\xi_{jl}j}. \tag{13}$$

In this work, we assume that the contribution probability of different webpages belong to the same website are identical, that is,

$$c_{\xi_{jl}j} = \frac{c_{\phi_{jk}j}}{n_{jk}}, l = 1, 2, \ldots, n_{jk}. \tag{14}$$

5.1.2.2 Cumulative probability distribution With the same reason as before, here we assume that the cumulative probability distribution from different webpages belong to the same website are identical. Let $\hat{S} = \{\phi_1, \ldots, \phi_m\}$ denote the set of all websites, therefore, we have $\forall i, k \in S$, if $i, l \in \phi_k, k = 1, 2, \ldots, m$,

$${}_iF_{T_j}(t) = {}_lF_{T_j}(t) \triangleq {}_{\phi_k}F_{T_j}(t). \tag{15}$$

We use the denotation ${}_{\phi_k}F_{T_j}(t)$ to represent the cumulative probability distribution of T_j from website ϕ_k.

By contrast with the other members of WMSP, without loss of generality, we further assume that the staying time T_j follows an exponential distribution in which the parameter is related to both page j and inlink website ϕ_k,

$${}_{\phi_k}F_{T_j}(t) = 1 - e^{-\lambda_{jk}t}, \quad k = 1, 2, \ldots, m. \tag{16}$$

That is, staying time depends on page j and the website of inlink, not the inlink page itself.

5.1.2.3 Summary Based on the above analysis, hereafter, we use contribution probability of site and cumulative probability distribution from site to calculate the mean staying time, and we can rewrite (8) as the following one, still let $\hat{S} = \{\phi_1, \ldots, \phi_m\}$ denote the set of all websites,

$$\begin{aligned}
\tilde{T}_j = E(T_j) &= \sum_{i \in S} c_{ij} \int_0^\infty t \times {}_iF_{T_j}(\mathrm{d}t) \\
&= \sum_{\phi_k \in \hat{S}} c_{\phi_k j} \int_0^\infty t \times {}_{\phi_k}F_{T_j}(\mathrm{d}t) \\
&= \sum_{\phi_k \in \hat{S}} c_{\phi_k j} \int_0^\infty t\lambda_{jk} e^{-\lambda_{jk}t} \mathrm{d}t \\
&= \sum_{\phi_k \in \hat{S}} \frac{c_{\phi_k j}}{\lambda_{jk}}
\end{aligned} \tag{17}$$

The major idea here is to assume that the staying time (utility) of a page is conditioned on the website of inlink page of it and calculate the mean staying time based on *inlink sites*. Moreover, we can change the mean staying time through the changes of two quantities: $c_{\phi_k j}$ and λ_{jk}.

Intuitively, the utility of web page does change according to the previous websites visited by the surfer. In the link graph case, if the surfer comes to the current page from a website with high utility (authority, quality, etc), then the utility of the page will also be

Table 1 MSMP construction algorithm

Input: Web graph and metadata.

Output: Page importance score π

Algorithm:

1. Generate transition probability matrix P of the EMC from web graph and metadata.
2. Calculate stationary distribution $\tilde{\pi}$ of the EMC using power method. (**page reachability**)
3. For each page j, identify its inlink websites and its inlink pages, and compute contribution probability $c_{\phi_k j}$.
4. For each page j, estimate parameter λ_{jk} from sample data included in metadata.
5. Calculate mean staying time $\tilde{T}_j$ for each page j with (17). (**page utility**)
6. Compute page importance for web graph with (9). (**page importance**)

high. In the user browsing graph case, if the surfer comes to the current page from a website with high utility (popularity, quality, etc.) and following a transition with high frequency, then the utility of the page will also be high.

The question then becomes how to calculate the contribution probabilities from different sites $c_{\phi_k j}$ and to estimate the parameters from different sites λ_{jk}. If we have enough observations of staying times, we can estimate the parameters $\lambda_{jk}, k = 1, \ldots, m$. In other cases (insufficient observations or web link graph), we can employ heuristics to calculate mean staying times. For contribution probabilities $c_{\phi_k j}$, we can also use heuristics to calculate them. We will discuss more about these in specific applications in Sect. 5.2.

Table 1 gives the detailed steps for creating the Mirror Semi-Markov Process model.

5.2 BrowseRank Plus

After introducing MSMP, we explain how to design the BrowseRank Plus algorithm to deal with click fraud in page importance computation. BrowseRank Plus tackled the problem of click fraud by using MSMP. In this algorithm, the mean staying time of a page is computed based on its inlinked websites. Specifically, the samples of observed staying time are distinguished according to different inlink websites by estimating different parameters $\lambda_{jk}, (k = 1, \ldots, m = |\hat{S}|)$. The estimation of the parameters λ_{jk} is similar to the method used in BrowseRank. Furthermore, in BrowseRank Plus, the contribution probability $c_{\phi_k j}$ also depends on the inlink websites. Suppose that the m_j sites that linked to page j are: $\Phi_j = \{\phi_{j1}, \phi_{j2}, \ldots, \phi_{jm_j}\}$, then the contribution probability is defined as below:

$$c_{\phi_k j} = \begin{cases} \frac{1}{m_j}, & \text{if } \phi_k \in \Phi_j \\ 0, & \text{if } \phi_k \notin \Phi_j \end{cases} \tag{18}$$

That is, we assume that the contributions from different inlink sites are all equal. Finally, the mean staying time is calculated by (17). Therefore, the mean staying times from different inlink websites will mostly differ from each other.

The main ingredient of BrowseRank Plus is the website normalization (18). We explain its effect through an example. Suppose that there are a large number of inlink sites to the page on the user browsing graph, that is, the sites from which users have made transitions to the current page. Click fraud usually has the following behavior patterns. (a) The number of clicks is high and the observed mean staying time is high. (In this way, the spammer can cheat BrowseRank-like algorithms in order to maximize their spamming

effect). (b) However, the visits usually come from a very limited number of websites. (This is because otherwise it will be very costly for the spammer). In BrowseRank Plus we make effective use of the fact (b) and normalize contributions from different sites.

Note that this is only one simple and example way to calculate the mean staying time. One can certainly consider more advanced ways for performing the task. Note that the key point is to control the contributions from different sites, and MSMP provides a framework to do it in a principled approach.

Proposition 2 *If parameters* $\lambda_{jk}, k = 1, \ldots, m$ *are all set to the same value, then BrowseRank Plus will degenerate to BrowseRank.*

Proof If we have $\lambda_{j1} = \lambda_{j2} = \cdots = \lambda_{jm} \triangleq \lambda_j$, then from (17) and (10) we obtain,

$$\tilde{T}_j = \frac{1}{\lambda_j} \sum_{\phi_k \in \hat{S}} c_{\phi_k j} = \frac{1}{\lambda_j}. \tag{19}$$

$\tilde{T}_j$ is exactly the mean staying time used in BrowseRank. □

6 Beyond BrowseRank plus: further discussions on MSMP

In this section, we skip out from the scope of BrowseRank Plus, and make more discussions on MSMP. First, we give a mathematical proof on the stationary distribution of MSMP, to make the new process more self-contained. Second, we provide another scenario of using MSMP to design page importance computation algorithm for Mobile web graph. Third, we give a brief analysis on the computational complexity of MSMP based algorithms.

6.1 Proof on stationary distribution of MSMP

As mentioned in Sect. 3.3, we use the product of stationary distribution of EMC X and the mean staying time on pages to model the page importance. The main reason is that in many cases it can be proved that such product is proportional to the limiting probability distribution of MSMP, if it exist. In this section, we will give the detailed explanation.

Generally, the limiting probability distribution of MSMP will not exist, except in some special conditions. We introduce them in the following lemma. Before it, we give some denotations here.

Let T_{jj} be the period between two successive departures from state j in MSMP. That means that the random surfer walks a circle from state j to state j during the period. All the circlings follow the *Staggered Renewal Process* (Papoulis and Pillai 2001). Let $E(T_{jj})$ stands for the mean time of the circling on state j.

Lemma 1 *Suppose Z is a Mirror Semi-Markov Process, if the transition probability matrix P of the embedded Markov chain X is irreducible, and probability distribution $P(T_{jj} \leq t)$ is not a lattice distribution*[5]*, and $E(T_{jj}) < \infty$, then the limiting probability distribution of Z exists, and by applying the Central Limit Theorem, we have*,

[5] Lattice distribution is a discrete distribution of a random variable such that every possible value can be represented in the form $a + bn$, where a, $b \neq 0$ and n is an integer.

$$\lim_{t\to\infty} P(Z_t = j) = \frac{E(T_j)}{E(T_{jj})} \tag{20}$$

which is independent of the initial state i of MSMP.

The lemma ensures the existence of the limiting probability distribution of MSMP, we denote it as $\pi_j \triangleq \frac{E(T_j)}{E(T_{jj})}$.

Theorem 1 *Suppose MSMP is defined on a finite state space S. If the Embedded Markov Chain X is ergodic, which implies both the limiting probability and the stationary distribution (denote as $\tilde{\pi}$) exist, and they are identical, i.e., $\lim_{n\to\infty} P^n = 1^T\tilde{\pi}$, furthermore, if the probability distribution $P(T_{jj} \le t)$ is not a lattice distribution, and $E(T_{jj}) < \infty$, then we have,*

$$\pi_j = \frac{\tilde{\pi}_j E(T_j)}{\sum_i \tilde{\pi}_i E(T_i)}. \tag{21}$$

Proof Let $T_j^{(k)}$ be the staying time of the k^{th} visit on state j, and $N_j^{(n)}$ be the number of times of leaving from state j in the past n transitions. Let $p_j^{(n)}$ be the proportion of staying time on state j during the previous n transitions. According to the lemma, we get that $\pi_j = \lim_{n\to\infty} p_j^{(n)}$, and due to the *Strong Law of Large Numbers*, we have

$$\begin{aligned} p_j^{(n)} &= \frac{\sum_{k=1}^{N_j^{(n)}} T_j^{(k)}}{\sum_i \sum_{k=1}^{N_i^{(n)}} T_i^{(k)}} \\ &= \frac{\frac{N_j^{(n)}}{n}\frac{1}{N_j^{(n)}}\sum_{k=1}^{N_j^{(n)}} T_j^{(k)}}{\sum_i \frac{N_i^{(n)}}{n}\frac{1}{N_i^{(n)}}\sum_{k=1}^{N_i^{(n)}} T_i^{(k)}} \\ &\xrightarrow{n\to\infty} \frac{\tilde{\pi}_j E(T_j)}{\sum_i \tilde{\pi}_i E(T_i)}. \end{aligned}$$

Thus we obtain (21). □

By comparing the (9) to the (21), we can easily find the product of stationary distribution of the EMC and the mean staying time is proportional to the limiting probability distribution of MSMP.

6.2 MobileRank

In this subsection, we discuss more about the MSMP model in specific applications, and design another algorithm whin this model for mobile web search.

Suppose that mobile graph data is available. Mobile graph contains web pages connected with hyperlinks and is for mobile phone accesses. Mobile web differs largely from general web in many aspects. For example, the topology of mobile web graph is significantly dissimilar from that of general web graph (Jindal et al. 2008). This is because the owners of websites on mobile web tend to create hyperlinks only to their own pages or pages of their business partners. As a result, there are more disconnected components in the mobile web graph, and usually links do not mean recommendation, but business

connection. In such case, the scores computed by algorithms like PageRank may not reflect the true importance of the pages.

We propose a new algorithm called MobileRank[6] for computing page importance on mobile web using MSMP. We actually consider a new way of calculating the mean staying time.

Note that in MSMP implementation we assume that staying time depends on not only the current page but also the inlink website, that means that MSMP has the ability to represent relation between websites and to utilize the information for promoting or demoting staying time (utility) of page. Specifically, if the inlink is from a partner website, then we can demote the staying time of visits from the website.

We define the contribution probability $c_{\phi_{jk}j}$ in the same way as in BrowseRank Plus (see (18)). We heuristically calculate the parameter λ_{jk}.

Suppose that for page j there is an observed mean staying time $\frac{1}{\lambda_j}$. λ_{jk} is assumed to follow a partnership-based discounting function L_{jk},

$$\frac{1}{\lambda_{jk}} = L_{jk}\left(\frac{1}{\lambda_j}\right). \tag{22}$$

The discounting function can have different forms for different business relations between websites. For example, we use a *Reciprocal Discounting* function in this paper,

$$L_{jk}(\eta) = \frac{cm_j^2}{n_{jk}}\eta. \tag{23}$$

where c denotes coefficient.

Therefore, we can calculate the mean staying time in MobileRank as below,

$$\tilde{T}_j = \frac{cm_j}{\lambda_j}\sum_{k=1}^{m_j}\frac{1}{n_{jk}}. \tag{24}$$

From (24) we can see that: (a) the larger number of inlink websites (i.e., m_j), the smaller penalty on the mean staying time; (b) given a fixed number of inlink websites (i.e., m_j), the larger number of inlink pages from the k^{th} website (i.e., n_{jk}), the larger penalty on the mean staying time by visits from the website.

Note that this is only one simple and example way of coping with the partnership problem on mobile web. One can certainly think about other ways of doing it.

Proposition 3 *If parameters $\lambda_j, j \in S$ are all set to the same value, and the discounting function is written as*,

$$L(\eta) = \frac{c}{m_j}\frac{1}{c_{\phi_{jk}j}}\eta, \tag{25}$$

then MobileRank will be reduced to PageRank.

Proof Substituting (25) into (22), and (22) into (17), we obtain

$$\tilde{T}_j = \sum_{k=1}^{m_j}\frac{c}{m_j}\frac{1}{c_{\phi_{jk}j}}\frac{1}{\lambda_j}c_{\phi_{jk}j} = \frac{c}{\lambda_j}. \tag{26}$$

[6] Note that we briefly discussed MobileRank in a short paper (Gao et al. 2009).

Since we further have $\lambda_1 = \lambda_2 = \cdots = \lambda_n \triangleq \lambda$, all $\tilde{T}_j$ are equal to each other, i.e., page importance only depends on the stationary distribution of the EMC. Therefore, it becomes equivalent to PageRank. □

6.3 Computational complexity

From Table 1, we can see that the computational complexity consists of calculation of page reachability, estimation of page utility, and combination of page importance. The calculation of page reachability is a PageRank-alike power method, and thus its complexity is $O(n^2)$, where n is the number of pages. Considering the transition matrix is usually very sparse, the actual complexity might be much lower than $O(n^2)$. For the estimation of page utility, as λ_{jk} can be computed in parallel on different pages, we only need to check (17). From it we see the computational complexity is $O(mn)$, where $m = \max_{j=1,\ldots,n} m_j$ which is usually a much smaller number than n. The combination of page importance is run by (2), from which we can see the complexity is $O(n)$. Therefore, the overall computational complexity of the proposed general framework is comparable with the PageRank algorithm, and thus it can scale up for importance computation on large graphs.

7 Beyond MSMP: further discussions on WMSP

In the previous sections, after introducing the general framework, we have made stepwise generalization following the line of PageRank, BrowseRank, BrowseRank Plus, and we have also discussed other related algorithms like MobileRank. In this section, we will stand at a high level and overseas all the discussed algorithms and Markov processes.

Figure 1 shows the relationship among the Markov processes and the corresponding page importance computation algorithms derived from them. In the figure, an solid directed edge from a to b means that a contains b as its special case, while a dashed directed edge from a to c means that c is an algorithm derived from a.

For the processes we can see that, (i) the Semi-Markov Process and the Mirror Semi-Markov Process are special cases of the Web Markov Skeleton Process, (ii) the Continuous-Time Markov Process is a special case of both the Semi-Markov Process and the Mirror Semi-Markov Process, and (iii) the Discrete-Time Markov Process is a special case of the Continuous-Time Markov Process.

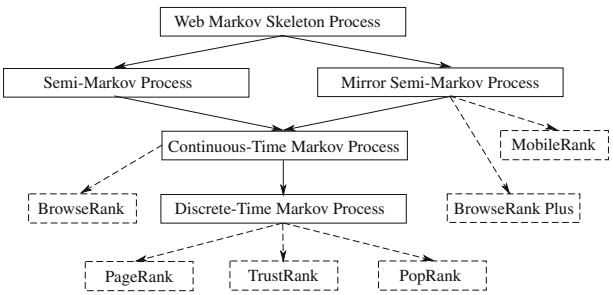

Fig. 1 Relationships of Markov processes and page importance computation algorithms (*arrow* means inclusion)

For the algorithms we can see that, (i) PageRank, TrustRank, and PopRank are derived from the Discrete-Time Markov Process, (ii) BrowseRank is derived from the Continuous-Time Markov Process, and (iii) BrowseRank Plus and MobileRank are derived from the Mirror Semi-Markov Process.

To give a detailed overview of the Markov processes and their related page importance computation algorithms, we organized them in Table 2. It summarizes the conditional probability of y_n and distributions of y_n on different states of S_n.

- For Discrete-Time Markov Process, the distributions of y_n on different states of S_n is a deterministic value, indicating that the mean staying time is regarded as an constant.
- For the other three cases of conditional probability of y_n, if the distributions of y_n on different states of S_n are different, we have not found corresponding page importance computation algorithms; if the distributions of y_n on different states of S_n are exactly exponential distributions, they will correspond to Continuous-Time Markov Process, Semi-Markov Process, and Mirror Semi-Markov Process, respectively; if the distributions of y_n on different states of S_n are from the same distribution family but are not the exponential distributions, they will correspond to renewal processes (Papoulis and Pillai 2001), which are far beyond the scope of this paper.

8 Experimental results

In order to validate the effectiveness of the proposed general framework, we conducted experiments to test the performances of the proposed algorithms (BrowseRank Plus and MobileRank) on two specific issues, important websites finding and spam/junk sites filtering. Two datasets were used in the experiments, with the first one being user browsing behavior data and the second one being mobile web data.

8.1 Ranking on user browsing graph

8.1.1 Dataset and baselines

We used a user behavior dataset from a commercial search engine for the experiments. All possible privacy information was filtered out and the data was sampled and cleaned as much as possible. There are in total over 3-billion records and 950-million unique URLs.

We first constructed the website-level user browsing graph as described in (Liu et al. 2008), by merging web pages in the same website, ignoring the transitions between the pages within the same website, and aggregating the transitions from (or to) the pages in the same website. This user browsing graph consists of 5.6-million vertices and 53-million edges. We then ran BrowseRank Plus and BrowseRank on this graph. We also obtained a *link graph* containing the 5.6-million websites from the commercial search engine, and computed PageRank and TrustRank from it as baselines.

For PageRank, we used the uniform teleport vector. For TrustRank, we first ran inverse PageRank on the link graph and selected the top 2000 websites as seed candidates. Then we manually checked the 2000 sites and removed spam, mirror, and porn sites from them, resulting in a seed set with about 1700 reliable websites. TrustRank was computed by

Table 2 Relationships of Markov processes and page importance computation algorithms

Conditional probability of y_n	Distributions of y_n on different states S_n		Stochastic process	Algorithms
$P(Y_n = 1) = 1$	Deterministic value		Discrete-Time Markov Process	PageRank, TrustRank
$P(Y_n \leq t \mid \{X_n\})$	Different distributions			
	The same distribution family	Exponential distribution	Continuous-Time (Time-Homogeneous) Markov Process	BrowseRank
		Others	Renewal Process (Papoulis and Pillai 2001)	
$P(Y_n \leq t \mid \{X_n, X_{n+1}\})$	Different distributions			
	The same distribution family	Exponential distribution	Semi-Markov process	
		Others	Renewal process	
$P(Y_n \leq t \mid \{X_{n-1}, X_n\})$	Different distributions			
	The same distribution family	Exponential distribution	Mirror Semi-Markov Process	BrowseRank Plus, MobileRank
		Others	Renewal process	

propagating trust scores from this seed set. For BrowseRank, we referred to the same setting as in (Liu et al. 2008).

8.1.2 Top-20 Websites

Table 3 shows the top-20 websites ranked by the different algorithms. For ease of reference, we use PR, TR, BR, and BR+ to denote PageRank, TrustRank, BrowseRank, and BrowseRank, respectively. We have the following observations from this table.

- BR+ gives high ranking scores to Web 2.0 websites (marked in bold), such as *myspace.com*, *youtube.com*, and *facebook.com*, and gives low ranking scores to the websites which have large number of inlinks in the web link graph but only get small number of visits by users. Therefore, compared with TR and PR, BR+ reflects users' preference much better. This observation is similarly to what we observe on BR. This is reasonable since BR is a special case of BR+ , and BR+ keeps the nice properties that BR has.
- BR+ can better demote the websites with high local transitions than BR. For example, *google.co.th* is such kind of website. We observe that *google.co.th* is ranked very high by BR due to its large click number and/or long mean staying time in the user behavior data. Further study into the user behavior data shows that most of the transitions are contributed by local websites in Thailand, while the number of transitions from websites outside Thailand is small. Therefore, in the BR computation, we get a large mean staying time for *google.co.th* because the BR algorithm does not distinguish the

Table 3 Top 20 websites by different algorithms

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

contributions from different inlink websites. However, BR+ will treat different inlink websites differently, and the website normalization in (18) can effectively reduce the influence of large number of transitions from the same website. Thus the ranking score of *google.co.th* is effectively decreased. Overall, the result given by BR+ looks a bit more reasonable than that given by BR.

8.1.3 Spam filtering

From the 5.6 million websites, we randomly sampled 10,000 websites and asked human labelers to make spam judgments on them. In the labeling process, the pure advertisement pages that do not contain much helpful information are also regarded as spam. Finally, 2,714 websites are labeled as spam and the rest 7,286 websites are labeled as non-spam.

To compare the performances of the algorithms, we draw the spam bucket distribution, which is similar to the setting in (Gyongyi et al. 2004; Liu et al. 2008). Specifically, for each algorithm, the 5.6-million websites are sorted in the descending order of the scores calculated by the algorithm. Then the sorted websites are put into fifteen buckets. The statistics of the spam websites over the fifteen buckets for different algorithms are summarized in Table 4.

We can have the following observations: (i) TR performs better than PR, for TR considers the trustiness scores calculated from the human labels and the link graph. (ii) Among all the algorithms, BR+ pushes the largest number of spam websites to the tail buckets. (Though BR has a bit more spam websites than BR+ in the last bucket, BR+ still outperforms BR in the total spam numbers of the last 6,5,4,3,2 buckets.) For example, we have observed that in the data http://www.latestlotteryresults.com is ranked high by BR but very low by BR+ . Data analysis shows that 26% transitions to this website come from one single website which contains an advertisement of http://www.latest-lotteryresults.com on

Table 4 Number of spam websites over buckets

Bucket no.	# of Websites	PR	TR	BR	BR+
1	15	0	0	0	0
2	148	2	1	1	1
3	720	9	11	4	6
4	2231	22	20	18	9
5	5610	30	34	39	27
6	12600	58	56	88	68
7	25620	90	112	87	95
8	48136	145	128	121	99
9	87086	172	177	156	155
10	154773	287	294	183	205
11	271340	369	320	198	196
12	471046	383	366	277	283
13	819449	434	443	323	335
14	1414172	407	424	463	482
15	2361420	306	328	756	753

Table 5 Top 10 websites by different algorithms

No.	PR	MR
1	wap.sohu.com	wap.sohu.com
2	wap.cnscu.cn	sq.bang.cn
3	m.ixenland.com	wap.joyes.com
4	i75.mobi	i75.mobi
5	planenews.com	wap.cetv.com
6	wap.joyes.com	waptx.cn
7	sq.bang.cn	zhwap.net
8	wap.ifeng.com	wapiti.sourceforge.net
9	u.yeahwap.com	www.download.com
10	wap.cetv.com	wap.kaixinwan.com

its homepage[7]. The data implies that the webmaster of the website seems to have purposely clicked the advertisement on the homepage. By website normalization, BR+ can effectively decrease the influence of this click fraud.

8.2 Ranking on mobile web graph

8.2.1 Dataset and baselines

The mobile web graph we used in the experiments is provided by a Chinese mobile search engine. The graph crawling was done in October 2008. There are about 80% Chinese webpages and 20% non-Chinese webpages in the graph. The numbers of webpages and hyperlinks are about 158-million and 816-million respectively. By our statistical study, the basic properties of this graph are similar to those reported in (Jindal et al. 2008). We computed PageRank (denoted by PR) and MobileRank (denoted by MR) on this graph.

8.2.2 Top-10 websites

We list the top-10 webpages ranked by the algorithms in Table 5. We conducted a user study and found that MR performs better than PR. In Table 5, *wap.cnscu.cn* hits the 2nd position in PR, while MR kicks it out of the top 10. After data analysis, we found that the homepage of the website has 78,331 inlinks from 829 other websites. The top 5 inlink websites contribute 8,720, 5,688, 3,764, 3,557, 2,920 inlinks respectively. That is, the top 5 contribute 31.5% inlinks. By MR, the effect from possible business models between *wap.cnscu.cn* and the above contributors on page importance is decreased. Another example is *wap.motuu.com*, which is ranked 11 by PR and 474 by MR. The homepage of the website has 12,327 inlinks only from 79 websites. Its top 5 contributors bring 3,350, 2,450, 1,662, 1,541, 1,144 inlinks respectively, occupying 82.3% of the 12,327 inlinks. MR was able to successfully decrease the effect from these websites and give it a low ranking.

[7] For privacy consideration, we do not list the name of this website in the paper.

Table 6 Number of junk pages over buckets

Bucket no.	# of Pages	PR	MR
1	23470	9	3
2	2751839	43	17
3	13285456	76	24
4	17766451	51	48
5	19411228	39	49
6	20299877	40	44
7	20916468	43	36
8	21278962	61	63
9	21278962	36	68
10	21278962	43	89

8.2.3 Junk filtering

We selected 1,500 pages from the mobile web graph in a random manner, and asked human labelers to make judgment on whether they are junk pages or not. Finally, 441 pages are labeled as junk and the rest are regarded as non-junk pages.

Similar to the spam filtering experiments in Sect. 8.1.3, we again use the bucket distribution to compare the performance of the algorithms. For each algorithm, we sort the 158-million pages in the descending order of their scores, and then put the pages into ten buckets. Table 6 summarizes the statistics of the junk pages over buckets.

We can see that MR performs better than PR in demoting junk pages to the tail buckets. For example, in the data a junk page (advertisement page) has a large number of inlinks to it. It appears that the links were added by a robot *only from several forum sites*. This falls into a very typical pattern, because the number of large forum sites which the spammers can use is limited. For MR (refer to formula (24)), given a fixed number of inlink websites, the larger the number of inlink pages from the site, the larger punishment on the mean staying time. That is why MR was able to demote more junk pages to the tail buckets than PR.

9 Discussion

In early years, several unified frameworks for link analysis were proposed. Ding et al. (2001) combined the concepts of PageRank and HITS into a unified framework from the viewpoint of matrix computation. Chen et al. (2002) extended HITS to analyze both hyperlinks embedded in the webpages and the interactions of the users with the webpages. They considered the two factors in a unified framework, and performed link analysis on the graph data. Poblete et al. (2008) combined hyperlink graph and click graph into a hyperlink-click graph, and ran a random walk on the combined graph. It fused multiple types of information to compute a more reliable random walk. The works described above are all very different from the proposed framework in the paper. Some of them (Ding et al. 2001) stand in the point of view of matrix computation, and the others (Chen et al. 2002; Poblete et al. 2008) focus on multiple signal fusion. The proposed general Markov framework is based on Markov processes and explains the factors of page importance computation. It can cover (Poblete et al. 2008) as a special case.

10 Conclusions and future work

In this paper, we have proposed a general Markov framework for page importance computation. In this framework, the Web Markov Skeleton Process is employed to model the random walk by the web surfer. Page importance is assumed to consist of two factors: page reachability and page utility. These two factors can be respectively computed as transition probabilities between webpages and the mean staying times of webpages in the Web Markov Skeleton Process. The proposed framework can cover many existing algorithms as its special cases, and can also provide us with a powerful tool in designing new algorithms. We showcase its advantage by developing a new Web Markov Skeleton Process called Mirror Semi-Markov Process and applying it to two application tasks: anti-spam on general web and page importance calculation on mobile web. Our experimental results indicate that the framework allows modeling different notions of page importance, based on which we can obtain different ranked lists. However, a more thorough experimental evaluation will be necessary to draw final conclusions on the practical importance for IR systems in web and mobile search.

As future work, we plan to try new ways of calculating the mean staying time; develop new Markov processes and new algorithms within the framework; and apply the general framework to the cases of heterogeneous web graph and web graph series.

Acknowledgments We thank Chuan Zhou for his valuable suggestions and comments on this work, and thank Liang Tang for his help on part of the experiments.

References

Berberich, K., Vazirgiannis, M., & Weikum, G. (2004). Time-aware authority ranking. *In Algorithms and Models for the Web-Graph: Third International Workshop, WAW'04* (pp. 131–141). Springer.

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

Boldi, P., Santini, M., & Vigna, S. (2005). Pagerank as a function of the damping factor. In *WWW '05*. ACM.

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.

Chen, Z., Tao, L., Wang, J., Liu, W., & Ma, W. (2002). A unified framework for web link analysis. In *WISE'02*.

Ding, C., He, X., Husbands, P., Zha, H., & Simon, H. (2001). PageRank, HITS, and a unified framework link analysis. LBNL Tech Report 49372, Nov 2001 (updated September 2002).

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

Gao, B., Liu, T., Ma, Z., Wang, T., & Li, H. (2009). A general markov framework for page importance computation. In *the Proceedings of the 18th ACM conference on information and knowledge management (CIKM 2009)* (pp. 1835–1838).

Gyongyi, Z., & Garcia-Molina, H. (2004). Web spam Taxonomy. *Technical report*, Stanford Digital Library Technologies Project.

Gyongyi, Z., Garcia-Molina, H., & Pedersen, J. (2004). Combating web spam with trustrank. In *VLDB '04* (pp. 576–587). VLDB Endowment.

Haveliwala, T. (1999). Efficient computation of pageRank. Technical Report 1999–31.

Haveliwala, T., & Kamvar, S. (2003). The second eigenvalue of the google matrix.

Haveliwala, T., Kamvar, S., & Jeh, G. (2003). An analytical comparison of approaches to personalizing pagerank.

Haveliwala, T. H. (May 2002). Topic-sensitive pagerank. In *WWW '02*, Honolulu, Hawaii.

Hou, Z., & Liu, G. (2005). *Markov Skeleton processes and their applications*. USA: Science Press and International Press

Hou, Z., Liu, Z., & Zou, J. (June 1998). Markov Skeleton Processes. *Chinese Science Bulletin, 43*(11), 881–889.

Jeh, G., Widom, J. (2002). SimRank: A measure of structural-context similarity. In *KDD '02*.

Jindal, A., Crutchfield, C., Goel, S., Kolluri, R., & Jain, R. (2008). The mobile web is structurally different. In *the Proceedings of the 11th IEEE global internet symposium*.

Kleinberg, J. M. (1998). Authoritative sources in a hyperlinked environment. In *SODA '98* (pp. 668–677). Philadelphia, PA, USA: 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., Zhang, Y., Ma, Z., He, S., et al. (2008). BrowseRank: Letting users vote for page importance. In *SIGIR '08* (pp. 451–458).

McSherry, F. (2005). A uniform approach to accelerated pagerank computation. In *WWW '05* (pp. 575–582). New York, NY, USA: ACM.

Nie, Z., Zhang, Y., Wen, J., & Ma, W. (2005). Object-level ranking: Bringing order to web objects. In *WWW'05*.

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

Papoulis, A., & Pillai, S. U. (2001). *Probability, random variables and stochastic processes*. New York: McGraw-Hill Science/Engineering/Math.

Poblete, B., Castillo, C., & Gionis, A. (2008). Dr. Searcher and Mr. Browser: A unified hyperlink-click graph. In *CIKM '08: Proceeding of the 17th ACM conference on information and knowledge mining* (pp. 1123–1132).

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*. Cambridge: MIT Press.

Yu, P. S., Li, X., & Liu, B. (2005). *Adding the temporal dimension to search—A case study in publication search. Proceedings of the 2005 IEEE/WIC/ACM international conference on web intelligence*.