A General Markov Framework for Page Importance Computation

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

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

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

ABSTRACT

We propose a *General Markov Framework* for computing page importance. Under the framework, a *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 *page reachability* and *page utility*, which can be computed from the transition probability and the mean staying time of the pages in the Markov Skeleton Process respectively. We show that this general framework can cover many existing algorithms as its special cases, and that the framework can help us define new algorithms to handle more complex problems. In particular, we demonstrate the use of the framework with the exploitation of a new process named *Mirror Semi-Markov Process*. The experimental results validate that the Mirror Semi-Markov Process model is more effective than previous models in several tasks.

Categories and Subject Descriptors

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

General Terms

Algorithm, Experimentation, Theory

Keywords

Page importance, PageRank, BrowseRank, general Markov framework, mirror semi-Markov process.

1. INTRODUCTION

Page importance 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 [6], TrustRank [1], and BrowseRank [4]. However, there are still challenges which cannot be well handled by these algorithms. (1) In some new scenarios, the assumptions in existing algorithms may not hold. 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 [3]. 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. (2) In some existing applications, the assumptions in existing algorithms may not be accurate either. In BrowseRank [4], basically it 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 click the link of the advertisement artificially or by a robot to increase the frequency of visits. As a result, we will observe a large volume of transitions from his homepage to the advertisement. 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, it may be necessary to develop new technologies to address the problems in both existing and new scenarios. Also, 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. For this purpose, we propose using a general Markov framework as the unified description of the page importance computation 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*. *Markov Skeleton Process* can represent these two factors with transition probability and mean staying time. We can then take the product of transition probability and mean staying time as page importance. In many cases, it can be proved that the product is proportional to the stationary distribution of the Markov Skeleton Process, if the distribution exists.

CIKM'09, November 2–6, 2009, Hong Kong, China.

Existing algorithms such as PageRank and BrowseRank can be well covered by the framework. 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, employing new family members of the Markov Skeleton Process, or developing new methods to estimate transition probability and mean staying time. To demonstrate the use of this framework, we propose employing a new process named *Mirror Semi-Markov Process*. In the new process, the staying time on a page depends on both the current page and the previous pages visited by the surfer. By doing so, we can address the aforementioned issues that existing algorithms suffer from. 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.

2. GENERAL FRAMEWORK

We assume that there is a web surfer performing random walk on the web graph[1]. The importance of a page can be viewed as the value that the page provides to the random surfer during her entire surfing process. Note that a visit of a page by the surfer is random, and the value which a page can offer to the surfer in one visit is also random. Therefore, there are two factors that can affect page importance: *page reachability* represents the (average) possibility that the surfer arrives at the page, and *page utility* represents the (average) value that the page gives to the surfer in a single visit. Page reachability is mainly determined by the structure of graph, while page utility can be affected by several things like the content of the page or the pages the surfer visited before. That is, page utility may depend on not only the current page but also other related pages.

2.1 Markov Skeleton Process

Markov Skeleton Process (MSP) is a stochastic process Z defined as follows[2]. Suppose that X is a Markov Chain with state space S and transition probability matrix P. Let $x_1, x_2, \cdots, x_\tau, \cdots$ denote a sequence of X, where x_τ is a state and $x_{\tau+1}$ is determined by the probability distribution $P(x_{\tau+1}|x_\tau)$, $(\tau = 1, 2, \cdots)$. Further suppose that Y is a stochastic process on the positive real-number set $\mathbf{R}^+$. Let $y_1, y_2, \cdots, y_\tau, \cdots$ denote a sequence of Y, where $y_\tau, (\tau = 1, 2, \cdots)$ is a positive real-number. Suppose that there are $S'_1, S'_2, \cdots, S'_\tau, \cdots$, and $S'_\tau \subseteq S, (\tau = 1, 2, \cdots)$. Y is determined by the probability distribution $P(y_\tau|S'_\tau), \tau = 1, 2, \cdots$. Then, Markov Skeleton Process Z is a Stochastic Process based on X and Y. A sequence of Z can be represented as $x_1 \xrightarrow{y_1} x_2 \xrightarrow{y_2} \cdots x_\tau \xrightarrow{y_\tau} \cdots$, where x_τ denotes a state and y_τ denotes staying time at state x_τ, $(\tau = 1, 2, \cdots)$. x_τ depends on $x_{\tau-1}$ and y_τ depends on multiple states S'_τ $(\tau = 1, 2, \cdots)$.

MSP is a very general stochastic process. It even does not necessarily have a stationary probability distribution. In some of its special cases, however, stationary distribution exists. Many existing stochastic processes are special cases of MSP. Some examples are *Semi-Markov Process* (when y_τ only depends on x_τ and $x_{\tau+1}$ according to distribution $P(y_\tau|x_\tau, x_{\tau+1})$), *Continuous-Time Markov Process* (when y_τ only depends on x_τ following an exponential distribution $P(y_\tau|x_\tau)$), and *Discrete-Time Markov Process* (when y_τ is constant). Furthermore, Mirror Semi-Markov Process, proposed in Section 3.1, is also a special case of MSP.

2.2 Modeling Page Importance

MSP can naturally model the random walk of web surfer. Suppose that Z is an MSP and (X, Y) is the two stochastic processes associated with Z. Let states in MSP correspond to web pages. The random surfer randomly chooses the next page to visit based on the current page, according to X. She further randomly decides the length of staying time on the current page based on the page, and several other pages she visited before, and/or several other pages she will visit, according to Y. Therefore, the aforementioned two factors are characterized by the two quantities in MSP. Specifically, transition probability represents page reachability, and mean staying time represents page utility.

Furthermore, we can define page importance as the product of transition probability and mean staying time. Note that there might be other forms for page importance besides the product defined above. The reason we use this definition is that we can prove the product is proportional to the stationary distribution of the Markov Skeleton Process if the process has a stationary distribution.

The framework can cover many existing algorithms like [6, 5, 1, 4]. Moreover, MSP does not make any specific assumption on the distribution $P(y_\tau|S'_\tau)$. 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 general process.

3. APPLICATION ALGORITHMS

We give a showcase of the proposed framework on dealing with new problems of page importance computation, by employing new family members of the Markov Skeleton Process and using new methods to estimate the two factors.

3.1 Mirror Semi-Markov Process

DEFINITION 1. *In Markov Skeleton Process Z, if Y is a stochastic process for which y_τ depends only on $x_{\tau-1}$ and x_τ according to $P(y_\tau|x_{\tau-1}, x_\tau)$, then we call Z Mirror Semi-Markov Process (MSMP).* □

MSMP[3] is a special case of Markov Skeleton Process. The Markov Chain X in MSMP with transition probability matrix P is called Embedded Markov Chain (EMC) of MSMP.

[1]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 [4].

[2]Note that we try to provide an intuitive definition here. A more rigorous definition can be found in [2].

[3]MSMP is similar to Semi-Markov Process (see Section 2.1). In Semi-Markov Process, y_τ depends on the current state x_τ and the next state $x_{\tau+1}$, while in MSMP y_τ depends on the current state x_τ and the previous state $x_{\tau-1}$. The dependencies are in two opposite directions. That is why we call the new model Mirror Semi-Markov Process.

Table 1: MSMP construction algorithm

Input: Web graph and metadata.
Output: Page importance score π

1. Generate transition probability matrix P of EMC from web graph and metadata.
2. Calculate stationary distribution $\tilde{\pi}$ of EMC using power method. (**page reachability**)
3. For each page j, identify its inlink websites and inlink pages, and compute contribution probability $p(\phi_{jk})$.
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 (2). (**page utility**)
6. Compute page importance for web graph with (1). (**page importance**)

Let $\tilde{\pi}$ denote the stationary distribution of EMC X. We have $\tilde{\pi} = \tilde{\pi}P$. Here $\tilde{\pi}$ can be calculated by the power method. We use j to represent a state ($j \in S$) and use t_j to represent staying time on state j. Suppose that $p(t_j)$ is the corresponding probability density function on staying time. Then the mean staying time on state j is defined as $\tilde{t}_j \triangleq E(t_j) = \int_0^\infty t_j p(t_j) dt_j$. We further define the following distribution using $\tilde{\pi}$ and $\tilde{t}_j$.

$$\pi_j = \frac{\tilde{\pi}_j \tilde{t}_j}{\sum_{i=1}^{n} \tilde{\pi}_i \tilde{t}_i}. \quad (1)$$

It can be proved that the stationary distribution for MSMP exists and the stationary distribution is exactly that in (1).

Given a web graph and its metadata, we build an MSMP model on the graph. We first estimate the stationary distribution of EMC and view it as transition probability. We next compute the mean staying time using the metadata. Finally, we calculate the product of transition probability and mean staying time, which is actually the stationary distribution of MSMP. We regard it as page importance. Table 1 gives the detailed algorithm for creating Mirror Semi-Markov Process. As the transition probability can be conveniently computed by power method, we will focus on the staying time calculation in the next subsection.

3.2 Staying Time Calculation

Suppose that for page j there are n_j pages linked to it: $\Xi_j = \{\xi_{j1}, \xi_{j2}, ..., \xi_{jn_j}\}$. Suppose $p(\xi_{jh})$ is the probability that the surfer comes to page j from page ξ_{jh}, then we have $\sum_{h=1}^{n_j} p(\xi_{jh}) = 1$. As probability $p(\xi_{jh})$ can stand for the contribution of page ξ_{jh} to page j, we refer to it as *contribution probability* in this paper. Suppose that the n_j inlinks of page j are from m_j websites, and from website k ($k = 1, ..., m_j$) there are n_{jk} inlinks. Thus we have $n_j = \sum_{k=1}^{m_j} n_{jk}$. 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: $\Phi_j = \{\phi_{j1}, \phi_{j2}, ..., \phi_{jm_j}\}$, and $p(\phi_{jk})$ is the probability that the surfer comes to page j from site k, referred to as construction probability of the site. Then we have $\sum_{k=1}^{m_j} p(\phi_{jk}) = 1$ and $p(\phi_{jk}) = \sum_{l=1}^{n_{jk}} p(\xi_{jl})$.

Let t_j be the random variable of staying time on page j and $p(t_j)$ be the probability density function on it. Then we have $p(t_j) = \sum_{h=1}^{n_j} p(t_j|\xi_{jh}) p(\xi_{jh})$, and the mean staying time $\tilde{t}_j$ on page j can be calculated as[4] $\tilde{t}_j \triangleq E(t_j) = \int_0^\infty t_j p(t_j) dt_j = \int_0^\infty t_j \sum_{h=1}^{n_j} p(t_j|\xi_{jh}) p(\xi_{jh}) dt_j$.

Here we assume that staying time t_j follows an exponential distribution in which the parameter is related to both page j and website k, i.e., $p(t_j|\phi_{jk}) = \lambda_{jk} e^{-\lambda_{jk} t_j}$. Furthermore, we assume $p(t_j|\xi_{jl}) = p(t_j|\phi_{jk}), l = 1, ..., n_{jk}$, i.e., staying time depends on page j and the website of inlink k, not the inlink page itself.

Combining the above discussions, we can calculate the mean staying time $\tilde{t}_j$ as

$$\tilde{t}_j = E(t_j) = \int_0^\infty t_j p(t_j) dt_j = \sum_{k=1}^{m_j} \frac{1}{\lambda_{jk}} p(\phi_{jk}). \quad (2)$$

The question then becomes how to calculate the contribution probabilities from different sites $p(\phi_{jk})$ and to estimate the parameters from different sites λ_{jk}. If we have enough observations of staying times, we can estimate the parameters $\lambda_{jk}, k = 1, ..., m_j$. In other cases (insufficient observations or web link graph), we can employ heuristics to calculate mean staying times. For contribution probabilities $p(\phi_{jk})$, we can also use heuristics to calculate them.

3.3 Anti-Spam: BrowseRank Plus

As explained, BrowseRank employs user browsing graph and Continuous-Time Markov Process. The biggest challenge for the algorithm is click fraud, because it trusts the user behavior data, and directly estimates the mean staying time from user behavior data. BrowseRank Plus addresses the problem by using MSMP. It calculates the mean staying time of a page on the basis of its inlinked websites. Specifically it partitions the samples of observed staying time according to inlink websites and estimate parameters $\lambda_{jk}, (k = 1, ..., m_j)$. We can use the same method in BrowseRank to estimate the parameters λ_{jk} from partitioned samples. Furthermore, BrowseRank Plus sets the contribution probability $p(\phi_{jk})$, also based on inlink websites, $p(\phi_{jk}) = \frac{1}{m_j}, (k = 1, ..., m_j)$. Finally, it calculates mean staying time using (2). Therefore, if the inlink websites are different, then the mean staying times from them will also differ.

3.4 MobileRank

To tackle the problems described in Section 1, we propose a new algorithm called MobileRank for computing page importance on mobile web using MSMP. We actually consider a new way of calculating 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. Specially, we define the contribution probability $p(\phi_{jk})$ in the same way as in BrowseRank Plus. 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}, i.e., $\frac{1}{\lambda_{jk}} = L_{jk}(\frac{1}{\lambda_j})$. The discounting function can have

[4]In practice, if a page does not have any inlinks, then it can be assigned the minimum mean staying time.

Table 2: 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

different forms for different business relations between websites. For example, we use a *Reciprocal Discounting* function like $L_{jk}(\eta) = \frac{cm_j^2}{n_{jk}}\eta$, where c denotes coefficient. Therefore, we can calculate the mean staying time in MobileRank as $\tilde{t}_j = \frac{cm_j}{\lambda_j}\sum_{k=1}^{m_j}\frac{1}{n_{jk}}$.

4. 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.

4.1 Ranking on User Browsing Graph

We used exactly the same datasets and settings as [4]. The top-20 websites ranked by using different algorithms are listed in Table 2. For ease of reference, we denote PageRank, TrustRank, BrowseRank, and BrowseRank Plus as PR, TR, BR, and BR+. From this table, we can see that: (1) similarly to BR, BR+ ranks Web 2.0 sites (marked in bold) high and ranks those sites low which have large number of inlinks in the link graph but small number of visits. In this regard, BR+ better reflects users' preference than TR and PR; (2) As compared to BR, BR+ can demote the sites with high local transitions, such as *google.co.th*. Study shows that most of the transitions are from local websites in Thailand, and the number of transitions from websites in other places is small. In this case, BR gets a large mean staying time for this website because it does not distinguish the contributions from different inlink websites. In BR+, however, different inlink websites are treated differently, and thus the influence of large number of transitions from the same website can be effectively reduced. In this way, the rank of *google.co.th* is effectively demoted.

We randomly sampled 10,000 websites from 5.6 million websites and asked human experts to conduct spam judgments on the websites. 2,714 websites are labeled as spam and the rest are labeled as non-spam. We use the spam bucket distribution to show the performances of the algorithms. Given an algorithm, we sorted the 5.6-million websites in the descending order of their scores given by the algorithm. Then we put these sorted websites into fifteen buckets. The numbers of labeled spam websites over the buckets for different algorithms are listed in Table 3. We see that among all the algorithms, BR+ pushes the largest number of spam websites to the tail buckets.

Table 3: 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 4: 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

4.2 Ranking on Mobile Web Graph

We used a mobile web graph collected from a Chinese mobile search engine to conduct the experiments. It was crawled in October 2008, containing about 80% Chinese pages and 20% pages in other languages. The graph contains 158-million webpages and 816-million hyperlinks. We studied the basic properties of the graph and found that they have similar tendencies as those reported in [3]. We took PageRank on the graph as the baseline, and applied MobileRank (denoted by MR).

We randomly sampled 1,500 pages from the mobile web graph and asked human judges to label them as junk page or non-junk page. As a result, 441 pages are labeled as junk and the rest are non-junk pages. We also use the bucket distribution to show the performances of the algorithms. The numbers of the labeled junk pages over buckets are listed in Table 4. We can see that MR can produce much better results than PR in demoting junk pages to the tail buckets.

5. CONCLUSION

In this paper, we have proposed a general Markov framework for page importance computation. In this framework, the Markov Skeleton Process is employed to model the random walk by the web surfer. The framework can cover many existing algorithms as its special cases, and can also provide us with a powerful tool in designing new algorithms.

6. REFERENCES

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

[2] Z. Hou, Z. Liu, and J. Zou. Markov Skeleton Processes. In Chinese Science Bulletin, vol 43, no 11, pages 881-889, June, 1998.

[3] A. Jindal, C. Crutchfield, S. Goel, R. Kolluri, and R. Jain. The mobile web is structurally different. In the Proceedings of the 11th IEEE Global Internet Symposium, 2008.

[4] Y. Liu, B. Gao, T. Liu, Y. Zhang, Z. Ma, S. He, and H. Li. BrowseRank: letting users vote for page importance. In SIGIR '08, pages 451–458, 2008.

[5] Z. Nie, Y. Zhang, J. Wen, and W. Ma. Object-Level Ranking: Bringing Order to Web Objects. In WWW'05, 2005.

[6] 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.