INCORPORATING THE SURFING BEHAVIOR OF WEB USERS INTO PAGERANK

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By
Shatlyk Ashyralyyev

August, 2013

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Cevdet Aykanat (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Fazlı Can

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Pınar Karagöz

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

INCORPORATING THE SURFING BEHAVIOR OF WEB USERS INTO PAGERANK

Shatlyk Ashyralyyev
M.S. in Computer Engineering
Supervisor: Prof. Dr. Cevdet Aykanat
August, 2013

One of the most crucial factors that determines the effectiveness of a large-scale commercial web search engine is the ranking (i.e., order) in which web search results are presented to the end user. In modern web search engines, the skeleton for the ranking of web search results is constructed using a combination of the global (i.e., query independent) importance of web pages and their relevance to the given search query. In this thesis, we are concerned with the estimation of global importance of web pages. So far, to estimate the importance of web pages, two different types of data sources have been taken into account, independent of each other: hyperlink structure of the web (e.g., PageRank) or surfing behavior of web users (e.g., BrowseRank). Unfortunately, both types of data sources have certain limitations. The hyperlink structure of the web is not very reliable and is vulnerable to bad intent (e.g., web spam), because hyperlinks can be easily edited by the web content creators. On the other hand, the browsing behavior of web users has limitations such as, sparsity and low web coverage.

In this thesis, we combine these two types of feedback under a hybrid page importance estimation model in order to alleviate the above-mentioned drawbacks. Our experimental results indicate that the proposed hybrid model leads to better estimation of page importance according to an evaluation metric that uses the user click information obtained from Yahoo! web search engine's query logs as ground-truth ranking. We conduct all of our experiments in a realistic setting, using a very large scale web page collection (around 6.5 billion web pages) and web browsing data (around two billion web page visits) collected through the Yahoo! toolbar.

Keywords: Page quality, web search, ranking, PageRank, BrowseRank.

ÖZET

WEB KULLANICILARIN TARAMA BİLGİLERİNİN PAGERANK İLE BİRLEŞTİRİLMESİ

Shatlyk Ashyralyyev
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Prof. Dr. Cevdet Aykanat
Ağustos, 2013

Büyük ölçekli ticari web arama motorunun kalitesini belirleyen en önemli faktörlerden biri arama motorunun bulduğu web arama sonuçlarının kullanıcıya sunulduğu sıralamadır. Modern web arama motorlarında, web arama sonuçlarının sıralamasının iskeleti sonuç sayfaların önemi ve sonuç sayfalarının verilen arama sorgusuyla ilişki bilgileri bir arada kullanılarak oluşturulmaktadır. Bu tez web sayfalarının küresel öneminin tahmin edilmesi ile ilgilidir. Şimdiye kadar, web sayfalarının önemini tahmin etmek için, iki farklı veri kaynağı birbirinden bağımsız bir şekilde ele alınmıştır: web sayfalarının arasındaki köprü bilgisi (PageRank) ve web kullanıcıların tarama bilgileri (BrowseRank). Ne yazık ki, her iki veri kaynağının da bazı sınırlamaları vardır. Web sayfalarının arasındaki köprü bilgisi pek güvenilir değildir, çünkü bu köprü bilgisi web içeriği yaratıcıları tarafından kolayca düzenlenebilmektedir ve kötü niyete karşı savunmasızdır. Öte yandan, web kullanıcıların tarama bilgilerinin en önemli sınırlamaları seyreklik ve düşük web kapsamasıdır.

Bu tezde, yukarıda belirtilen sınırlamaları kaldırmak için yukarıda bahsedilen iki tür veri kaynağının karışımını kullanarak web sayfalarının küresel öneminin tahmin eden model tasarlanmıştır. Yahoo! web arama motorunun sorgu günlüklerinden elde edilen kullanıcı tıklama bilgilerini gerçek sıralama olarak kullanan bir değerlendirme metriğine göre iki farklı veri kaynağının bir arada kullanılması sayfa öneminin daha iyi tahmin edilebildiğini göstermektdir. Deneyler sırasında çok büyük ölçekli web sayfa veri seti (yaklaşıl 6.5 milyar web sayfası) ve Yahoo! araç çubuğu üzerinden toplanan web tarama veri seti (iki milyar web sayfa ziyareti) kullanılmıştır.

Anahtar sözcükler: Web sayfa kalitesi, web araması, sıralama, PageRank, BrowseRank.

Acknowledgement

I would like to express my gratitude to my supervisor Prof. Dr. Cevdet Aykanat for his guidance and insightful suggestions during the past two years. He was patient and tolerant all the time, even when I was on the verge of dropout.

I am also more than thankful to Dr. Barla Berkant Cambazoğlu for his great contribution and guidance throughout the every step of this thesis. Thanks to him, my vision towards research has completely changed and I have decided to pursue PhD studies.

I am also thankful to Prof. Dr. Fazlı Can and Assoc. Prof. Dr. Pınar Karagöz for reading and commenting on this thesis.

Of course, my family: my mom Govherjan Ashyraliyeva and my dad Prof. Dr. Allaberen Ashyralyev, my siblings Assist. Prof. Dr. Maksat Ashyraliyev, Mahri Ashyraliyeva, Merjen Ashyraliyeva, Maral Ashyraliyeva and Gulruh Ashyraliyeva. Without their moral support and the motivating questions they used to ask (e.g., "When are your graduating?" and "How is your research going?"), it would be extremely hard to finish these studies. Moreover, I would like to mention my nephews and nieces: Annageldi, Akmuhammet, Hatyja and Davud. I am also grateful to my ancestors for choosing such a long surname and helping me to extend this thesis with few more lines.

I would like to thank all of my friends, especially Cansu, Eje, Fahrettin, Halil, Sema, Serkan, Tarı and Utku + Can, for simply being great friends.

The last but not the least, I had great colleagues. Special thanks to Salim for drawing funny comics, to permanent high schooler Etkin Barış, to devoted haxball teammates Alper, Selçuk Onur and Erdem, to my "thesis mates" Elif and Bengü, to my "cubic mate" Seher and to all members of the secret EA525 group.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the tremendous expansion of the Internet, searching for an information on the World Wide Web (WWW) became an important topic. To this purpose, hundreds of web search engines have been developed in the last few decades[1]. Most of them have failed to survive in the web search engine war because of the high quality search services served by their powerful opponents. The quality of search engines depends on many factors including the speed of the search process and the quality of the returned content. In this thesis, we are concerned with the latter issue, i.e., the quality of the search results returned by the engine for the given search queries.

The quality of search results usually depends on how the user is satisfied with the results. Here, the user satisfaction has various dimensions. One of them is the query-result relevance. User simply expects the results to be relevant to the query as much as possible. The problem of determining the most relevant pages can be resolved using query-dependent features, such as BM25, which are usually used to estimate the degree of relevance between a given query and a document. However, in the context of large-scale web search engines, quantifying only the relevance is not enough because of the following example. Consider a simple web page containing a single word: “Barack Obama”. This page would have a

[1] Search Engine History, `http://www.searchenginehistory.com/`

perfect relevance with the search query “Barack Obama”. However, if this page is returned as a top result for the search query “Barack Obama”, the end user would not be satisfied with it. This is because there are much better options to be ranked as a top result, such as the Wikipedia page of Barack Obama or latest news about Barack Obama. Therefore, the large size of the Web and high variation in content quality necessitate distinguishing the importance of web pages independent of the query. In our example, since the query-independent importance of the Wikipedia page would be higher than the importance of the simple web page, final results would rank the Wikipedia page in higher ranks than the simple page. To this end, most web search engines incorporate query-independent page importance scores into their ranking algorithms, either as separate features used in machine-learned ranking models [1] or as a linear combination with a query-dependent relevance score [2].

PageRank [3] is perhaps the most well-known and widely used technique for computing web page importance. This technique uses the hyperlink structure of the Web as a data source. It represents the hyperlink structure as a Markov chain, in which a web surfer is assumed to move across web pages following the hyperlinks or occasionally making random jumps. The stationary distribution of this Markov chain, obtained through an iterative process, provides the final importance scores of web pages. The basic idea behind this technique is to compute the importance of a web page based on the quantity of the links received from other pages as well as the quality of those referring pages. The former factor is motivated by the assumption that receiving many links from other pages is an indication of good content quality. The latter factor is due to the assumption that important pages tend to link other important pages.

Although PageRank has found many important use cases, there are two serious drawbacks in the application of this technique to estimation of web page importance. First, PageRank solely relies on the hyperlink structure of the Web without incorporating any kind of feedback from the real users surfing the Web. Therefore, all pages are treated equally, ignoring their importance for end users or the likelihood of being visited by a web surfer [4]. Second, since the hyperlink structure is mainly created by the web site owners, it is subject to manipulation.

As an example, link farms can be created to artificially boost the importance of certain web pages, making PageRank vulnerable to link spam [5].

An interesting alternative to PageRank is to exploit the web surfing behavior of users to assess the importance of web pages (e.g., BrowseRank [6]). In this approach, the existing hyperlink structure is completely omitted. Instead, a virtual link structure is created between web pages based on the web browsing patterns of users, i.e., the transitions they make between different pages when surfing the Web. Such patterns can be obtained by mining navigational user activity that is tracked by the toolbar applications, commonly installed in web browsers. This approach provides better quality feedback about page importance and also solves the previously mentioned spam problem associated with PageRank. However, it is not without any drawbacks. In practice, the web browsing patterns extracted from the toolbar logs are very sparse. Even with a toolbar application deployed at web scale, the obtained web browsing patterns can capture only a small fraction of pages in the Web. Hence, many web pages (especially, the less popular web pages) are not covered and their scores cannot be computed.

One of the main objectives of this thesis is to investigate whether combining web and user feedback (i.e., using both web data and browsing data) improves the quality of page rankings over using only one type of feedback. To this end, we define a discrete-time Markov chain constructed by aggregating web and browsing data with properly scaled page transition probabilities. Importance scores of pages are estimated using the standard procedure followed in PageRank computations. We refer to the proposed technique as PBRank (PageBrowseRank) since it can be considered as a mixture between PageRank and a discrete-time variant of BrowseRank. We conduct all of our experiments using a very large scale and realistic setting. In particular, we work with a large host-level graph, containing 230 million vertices obtained by processing a 6.5 billion web page collection. We also use a very large toolbar log containing two billion page visits. This work has been accepted for 22^{nd} ACM International Conference on Information and Knowledge Management (CIKM 2013).

The contributions of this thesis can be summarized as follows:

- We propose a hybrid ranking model that estimates the importance of a page by using a mixture of feedback obtained from the hyperlink structure of the Web as well as the web browsing patterns of users.
- We shed light into the overlap between the web data, browsing data, and web search click data as well as the correlation between the importance values assigned to web hosts by these data sources.
- We experiment in a realistic setting with very large data, orders of magnitude larger than the data used in earlier works in the same problem context.

The following are the selected findings of this thesis:

- Exploiting both web and user feedback at the same time improves the quality of the page ranking compared to using only one type of feedback.
- Using the web data increases the coverage (the number of web hosts for which an importance score can be computed) over using only the browsing data.
- When the web and user feedbacks are optimally combined, the user feedback has 99 times more influence on the quality of page rankings than the web feedback.
- We observe little correlation between web data and browsing data and a relatively stronger correlation between browsing data and click data in terms of the importance values they attribute to web hosts.
- It may be useful to customize page ranking models taking into account the location of users.

The rest of the thesis is organized as follows. Chapter 2 explains the related work done on this topic. Two previously mentioned algorithms, PageRank and BrowseRank, are described in Chapter 3 and Chapter 4, respectively. Our proposed solution, PBRank, is explained in Chapter 5. Then, in Chapter 6, we explain the proposed evaluation metric we use for the evaluation of PBRank. In Chapter 7, we provide the characteristics of our data together with our experimental setup. All experimental results are presented in Chapter 8. Finally, we conclude the thesis in Chapter 9.

Chapter 2

Related Work

PageRank is originally proposed in [3] and used as the skeleton of Google Search Engine[1]. The technique finds application in a variety of problems from different domains including bibliometrics [7], web crawling [8], spam detection [9], and NLP [10], besides web search result ranking [1]. HITS [11] and SALSA [12] are two techniques closely related to PageRank. Graph-theoretic techniques are employed in [13] to approximate the PageRank scores. So far, considerable research effort is spent to speed up PageRank computations, either by algorithmic improvements that aim to accelerate convergence [14, 15, 16, 17] or via distributed processing [18, 19, 20]. Interested reader may refer to [21] and [22] for a survey of further issues.

A large effort is spent to customize PageRank computations depending on the interests of users. This is mainly achieved by either adjusting the α constant, which shows the probability of following a link in the current page, or by customizing the page-specific jump probabilities in the teleportation vector $\mathbf{v}$ (see Eq. 3.3). Regarding the first possibility (customizing the random jump probability), several works investigated the effect of α on the quality of the final rankings [4, 23, 24, 25]. The order of pages in the final PageRank vector is found to be heavily affected by the α constant used [25]. The results reported in [24] show that α values close to 1 do not yield accurate rankings. Two latter works

[1] Google, `http://www.google.com/`

suggest using α values around 0.5 [23] or in the 0.6–0.725 range [4]. The approach proposed in [4] is relevant to ours in that it relies on the web browsing data to set the α constant.

Regarding the second possibility (customizing the teleportation vector), several attempts were made [15, 26, 27]. A comparison of three alternative techniques using PageRank for customization is available in [28]. In topic-sensitive PageRank [26], in an offline phase, the topics of the pages are determined and separate PageRank vectors are computed for a fixed number of topics. The PageRank computation is biased to yield higher scores for pages belonging to a certain topic by simply adjusting the jump probabilities in the teleportation vector. In the offline phase, a user query is mapped to a topic and the value in the corresponding PageRank vector is used in the score computations. In [15], a similar idea is described, restricting personalization preferences to blocks of web domains instead of topics. This approach is considerably more efficient than using the standard PageRank model for personalization. Nevertheless, the performance is far from generating query-time personalized rankings. In [27], a scalable personalization approach is presented. In this approach, an approximate personalized PageRank vector is computed based on precomputed basis vectors. The BrowseRank approach [6] relies on web browsing data to customize the teleportation vector.

Our work goes beyond these works in three different aspects. First, in the proposed ranking model, we use web browsing data of users to customize the probabilities in the transition matrix, instead of adapting only the α constant as in [4] or adjusting the probabilities in the teleportation vector as in [6]. In this respect, our model can accurately capture the variation in the quality of the links within web pages, unlike the above-mentioned two works, which assume a uniform probability for following a link in a page. Second, we show the spatio-temporal variation in user browsing behavior and apply our model to this scenario. Finally, we conduct our experiments in a very large setting, orders of magnitude larger than the settings in most previous work.

Previous work on web browsing data. Web browsing data obtained from toolbar applications is used for various other purposes, besides improving

PageRank. In [29], URLs in the browsing data are used to increase the web coverage of a commercial crawler and the impact of this on the search result quality is demonstrated. Web content change is investigated in [30], restricting the attention to URLs in browsing data. URL revisitation of toolbar users is analyzed in [31]. The concurrent web browsing behavior of users is investigated in [32]. A high-level taxonomy for online browsing behavior of users is presented in [33].

Chapter 3

PageRank

PageRank is first introduced in [3] and is motivated by the academic citation literature. It exploits the hyperlink structure of the Web to estimate the importance of web pages. PageRank first constructs a link graph using the hyperlink structure of the crawled web pages. Then, it represents the random surfing behavior of web users using a discrete-time Markov chain. Finally, the stationary probability distribution of the above-defined Markov chain becomes the importance of web pages. We would like to explain the basics of the random surfer model using examples and then mathematically describe the PageRank algorithm. Note that, we present PageRank in detail since some of the notation introduced in this Chapter is reused in Chapter 5, where we explain our proposed solution.

3.1 Random surfer on a sample Web graph

WWW is composed of web pages, where a web page is composed of HTML content including hyperlinks to other web pages. A sample Web composed of 5 web pages is given in Fig. 3.1. Now, consider a web user who randomly surfs on the Web by clicking on the hyperlinks. In the rest of this thesis, we call this web user as a random surfer and the clicking process as *transportation*. Here, we assume that all hyperlinks in a particular web page have same probabilities to be clicked by

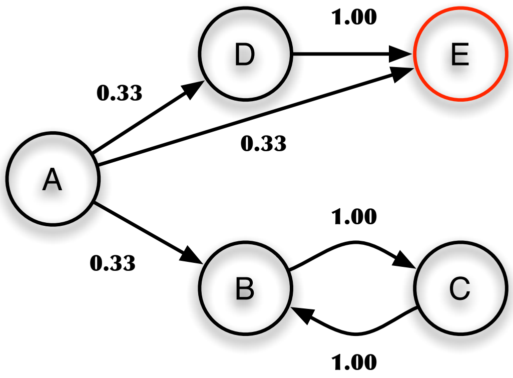


Figure 3.1: A sample Web composed of 5 web pages: A, B, C, D and E. There are links among web pages, such that, page A links to pages B, D and E; page D links to page E; and pages B and C have mutual links. Dangling page E is highlighted with red.

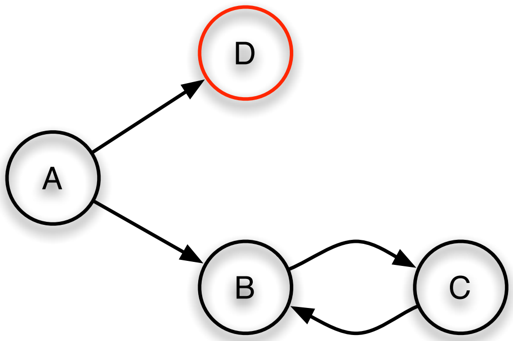


Figure 3.2: The solution for dangling pages.

the random surfer. Fig. 3.1 shows the clicking probabilities of all hyperlinks.

An obvious problem occurs on the pages containing zero hyperlinks (called as *dangling* pages). Random surfer stops when reaches a dangling page, because there are no available options for the next step. There is only one dangling page in the sample Web, which is page E and highlighted with red color in Fig. 3.1. One solution for this problem is to remove all dangling pages from the web before then random surfer starts surfing. This is shown in Fig. 3.2. Unfortunately, removing dangling pages from the Web may introduce other dangling pages (i.e., page D). Of course, one may continue removing dangling pages until no dangling page left on the Web, but we do not consider this solution. Instead, we describe another solution for the dangling page problem. We assume that when the random surfer

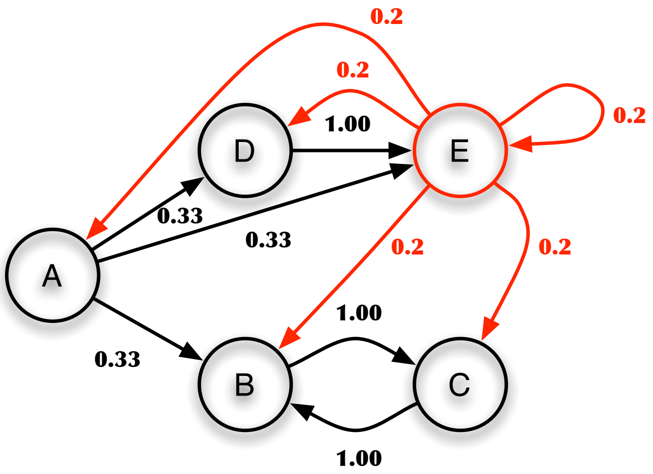


Figure 3.3: The solution for dangling pages.

reaches a dangling page, surfer jumps to any other page on the Web. Moreover, we assume that all pages on the Web have same probabilities to be jumped to. Fig. 3.3 shows how the random surfers jumps to other pages when reaches the page E. In the rest of this thesis, we the jumping process as *teleportation*.

Although this model seem to serve a perfect environment for the random surfer, there is one last problem. For the sample Web in Fig. 3.3, assume that the random surfer reaches either page B or page C. After that point, the surfer enters a loop and never goes back to pages A, D or E. This is called as a *loop* problem. In order to overcome loops, we extend the jumping process (defined for dangling pages) to all pages as follows. We assume that when the surfer is on a particular page, the probability that the surfer will click on a hyperlink is α and the probability that the surfer will jump to other pages is $(1 - \alpha)$, where α is in the $[0, 1]$ range. This introduces a possibility of jumping from any page to any other page. Fig. 3.4 shows the jumping probability from the page C.

The model in Fig. 3.4 serves a perfect environment for the random surfer. After fixing the problems in the hyperlink structure of the Web, PageRank defines the importance of a particular web page as the probability that the random surfer will be at that page after infinite steps of clicks and jumps. In particular, for $\alpha = 0.85$ the probability that the random surfer will be at that page A, B, C,

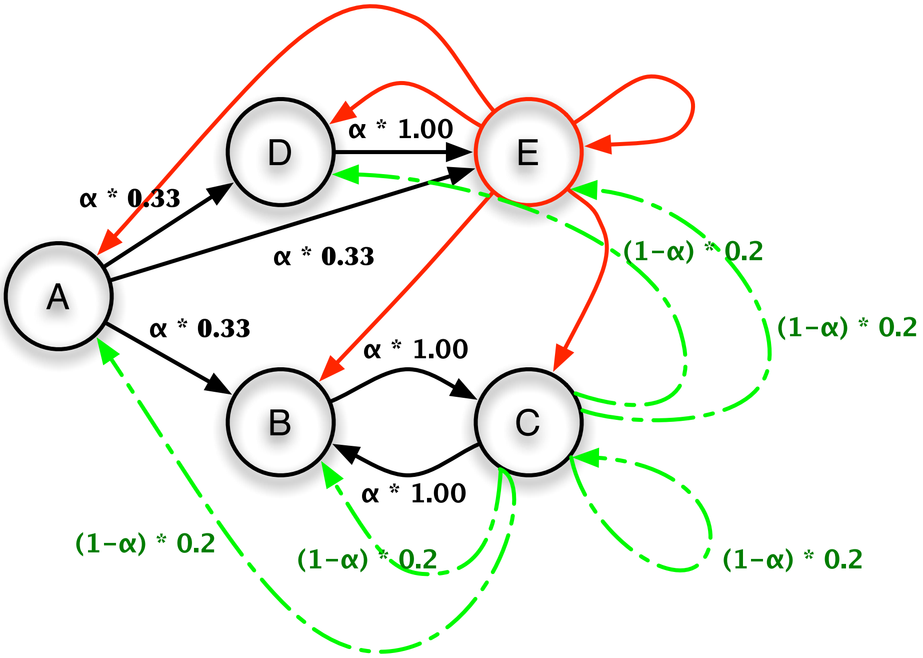


Figure 3.4: Random jumps in PageRank.

D or E after infinite steps of clicks and jumps is 0.05, 0.39, 0.38, 0.07 and 0.12, respectively. This means, the importance ranking of the pages is $<B, C, E, D, A>$, where the page B is the most important page and the page A is the least important page.

3.2 PageRank definition

As explained in previous section, in PageRank, the computation of scores relies on a probabilistic model known as the random surfer model, where the score of a page is defined by the stationary probability that the surfer will be at that particular page at some time step in the future. This model consists of a Markov chain induced by a random walk on a web graph having n vertices. Each state of the chain corresponds to a different vertex in the web graph. A transition matrix $\mathbf{P} = (p_{ij})$ is associated with this chain such that

$$p_{ij} = \begin{cases} 1/|\mathcal{L}_i|, & |\mathcal{L}_i| > 0; \\ 0, & \text{otherwise.} \end{cases}, \qquad (3.1)$$

where $|\mathcal{L}_i|$ denotes the set of out-links of page i. This transition matrix stands for the probabilities of hyperlinks to be clicked (see Fig. 3.1). Given this transition matrix, the PageRank vector $\mathbf{p}=(p_i)$, where p_i indicates the score of page i, can be computed by finding the Markov chain's stationary distribution that satisfies $\mathbf{p}=\mathbf{P}^{\mathrm{T}}\mathbf{p}$, i.e., the principal eigenvector of the chain. The solution can be obtained through a series of iterations of the form $\mathbf{p}^{k+1}=\mathbf{P}^{\mathrm{T}}\mathbf{p}^{k}$ using the power method [34]. The existence of a solution, i.e., the convergence of iterations, requires the $\mathbf{P}$ matrix to be stochastic, irreducible, and aperiodic, neither of which are guaranteed for $\mathbf{P}$.

The reason behind matrix $\mathbf{P}$ not being stochastic is the presence of dangling pages with no out-links. Although there are other possibilities [15, 27, 35], the common solution [3, 36] to this problem is to add artificial links from such pages to every other page in the Web. This is exactly the same solution we presented for dangling nodes in Fig. 3.3 and it results in a stochastic transition matrix $\mathbf{P}'$, computed as

$$\mathbf{P}' = \mathbf{P} + \mathbf{d}\mathbf{v}^{\mathrm{T}}, \tag{3.2}$$

where $\mathbf{d}=(d_i)$ is a dangling page vector (if i is a dangling page, $d_i=1$; otherwise, $d_i=0$) and $\mathbf{v}=(v_i)$ is a vector, where v_i indicates the transition probability from dangling pages to a specific page i. Typically, the transition probabilities are set equal for all pages, i.e., $v_i=(1/n)$, but there are other alternatives as well [37]. The resulting matrix $\mathbf{P}'$ is stochastic, but not irreducible. Applying a similar technique on $\mathbf{P}'$, an irreducible stochastic transition matrix $\mathbf{P}''$ can be obtained, also guaranteeing aperiodicity as

$$\mathbf{P}'' = \alpha\mathbf{P}' + (1-\alpha)\mathbf{e}_n\mathbf{t}^{\mathrm{T}}. \tag{3.3}$$

Here, $\mathbf{e}_n$ is a vector of size n containing all ones. α denotes the probability that the surfer will follow one of the links in the current page while $(1–\alpha)$ is the probability that the surfer will jump to a page that is not necessarily linked by the current page. Again, this is the mathematical representation of the solution presented in Fig. 3.4. In practice, α values between 0.85 and 0.9 are used although this value can be further tuned using feedback obtained from external sources [4, 23]. The

$\mathbf{t} = (t_i)$ vector is referred to as the teleportation vector, where t_i indicates the probability of jumping to page i. Typically, this probability is set to $1/n$ for all pages. In case of personalized or topical teleportation vectors, non-uniform jump probabilities can also be used [26].

Chapter 4

BrowseRank

In this section we briefly summarize the BrowseRank algorithm presented in [6]. BrowseRank differs from PageRank in two main ways. First, instead of using a link graph based on the hyperlink structure of the Web, BrowseRank mines the user behavior data collected from users and constructs a “user browsing graph”. Second, rather than using a discrete-time Markov process on the link graph, the random walk on the user browsing graph is represented as a continuous-time Markov process and the staying times of users on the pages are taken into account. Moreover, [6] presents an efficient algorithm (i.e., BrowseRank) for computing the stationary probability distribution of this process.

Now, we briefly explain the construction of a user browsing graph, the representation of a random walk as a continuous-time Markov process, and finally the computation of the stationary probability distribution of this process. For further details of the BrowseRank we refer the reader to [6, 38].

Table 4.1: An example user browsing history used by BrowseRank.

URL	TIME	TYPE
`http://www.aaa.com/`	2013-01-05, 17:30:05	INPUT
`http://www.bbb.com/`	2013-01-05, 17:35:56	CLICK
`http://www.ccc.com/`	2013-01-05, 17:40:45	CLICK

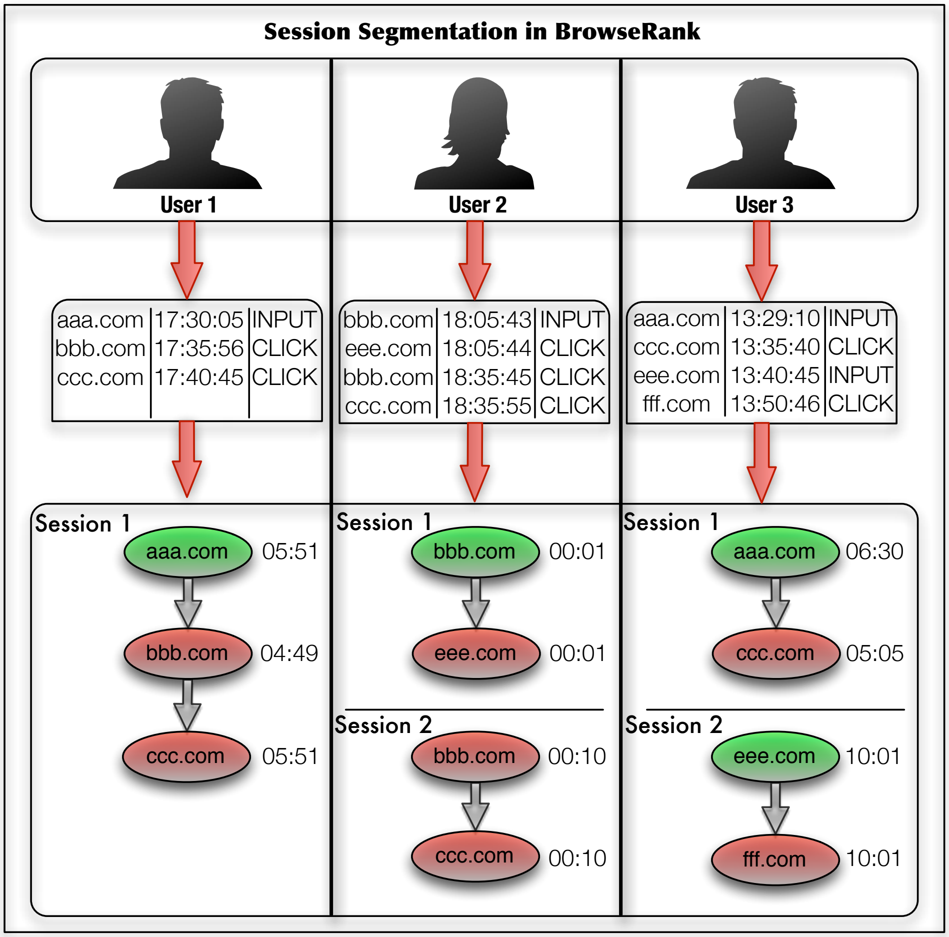
4.1 User Browsing Graph

A user browsing graph constructed by BrowseRank is a weighted graph where vertices represent web pages, directed edges between the vertices represent transitions between web pages by users and the edge weights stand for the total number of transitions between corresponding two pages by all users. Additionally, vertices are associated with staying times of web users on respective pages and reset probabilities [1] (i.e., teleportation probabilities) of those pages.

Web Browsing History. The user browsing data needed for the construction of a user browsing graph is extracted from web browsing history of a user recorded by Internet browsers at web clients. In the web browsing history of a user, each page visit is recorded in triples: *URL*, *TIME* and *TYPE*. Here, *URL* is the URL of the visited web page, *TIME* is the timestamp of the page visit, and *TYPE* is either “CLICK” or “INPUT” depending how user has arrived to the visited page. “CLICK” type occurs when the user clicks on a hyperlink from the previous page and it stands for *transportation* in PageRank. On the other hand, the page visit type is “INPUT” when the user arrives at the page by manually typing the URL or by clicking a bookmark link. Similarly, the “INPUT” type represents the *teleportation* in PageRank. An example browsing history of a web user is given in Table 4.1. Note that the rows in the browsing history are sorted in chronological order.

Session segmentation. An obvious problem with this data is the absence of the referring URLs for the records with “CLICK” types, i.e., the page from which a user clicked on a hyperlink is unknown. This problem is resolved by

[1] The BrowseRank paper uses the term “reset probability” instead of the term “teleportation probability”. In this chapter, in order to stay consistent with the original paper, we use the term “reset probability”.



aaa.com	17:30:05	INPUT
bbb.com	17:35:56	CLICK
ccc.com	17:40:45	CLICK

bbb.com	18:05:43	INPUT
eee.com	18:05:44	CLICK
bbb.com	18:35:45	CLICK
ccc.com	18:35:55	CLICK

aaa.com	13:29:10	INPUT
ccc.com	13:35:40	CLICK
eee.com	13:40:45	INPUT
fff.com	13:50:46	CLICK

Figure 4.1: Session Segmentation in BrowseRank.

segmenting the browsing logs of an individual user into *sessions*. A *session* is a sequence of consecutive records in the browsing history of an individual user. Records in a browsing history are segmented into sessions using two rules. *Type rule:* any record with an "INPUT" type is accepted as a start of a new session. *Time rule:* if there is a 30 minute gap before a record with a "CLICK" type, then the corresponding record is also assumed to be the start of a new session [39].

Staying times. After session segmentation, the staying time on a page is calculated for every page visit. The staying time on a page is defined as the difference between the visit time of the next record within the same session and the visit time of the current record. Obviously, last record of a session needs a special handling. Let p denote the last record of a session. If the session of the

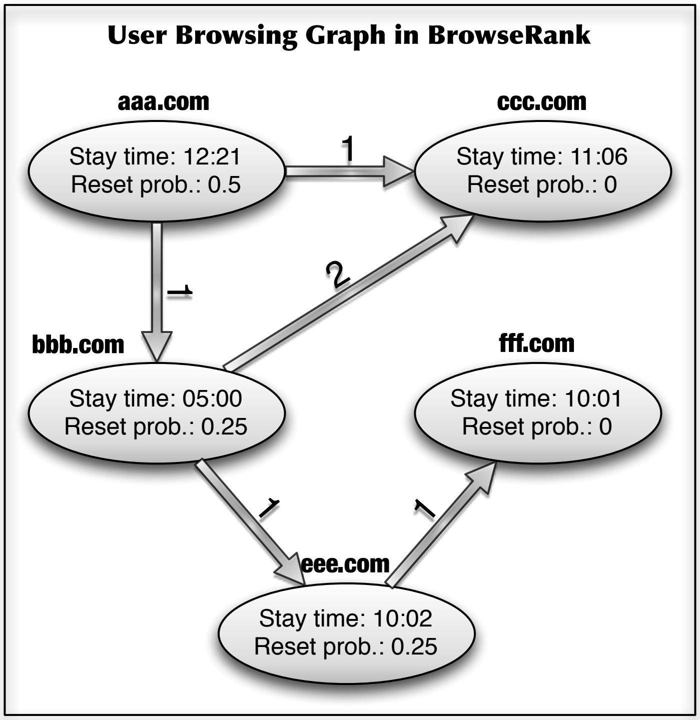


Figure 4.2: User Browsing Graph in BrowseRank.

record that comes after p in the browsing history is segmented because of the *time rule*, then the staying time on p is randomly sampled from the staying times of the other records in p's session. Otherwise, the staying time on p is simply the difference between the visit time of a record that comes after p and the visit time of p. Fig. 4.1 shows the session segmentation process and the calculated staying times on the web pages.

Reset Probabilities. One more interesting observation is that the reset probabilities of web pages can be estimated using the browsing records with "INPUT" types. In [6], web pages visited in such records are called as *green traffic*, because a web page visited by typing its URL is assumed to be safe and important. Moreover, such records perfectly represent the "random jump" (i.e., teleportation) process in the random surfer model. Therefore, frequencies of URLs that appear in records with "INPUT" types are normalized to get the reset probabilities of the corresponding web pages. Fig. 4.1 shows the *green traffic* using green vertices and Fig. 4.2 shows the reset probabilities of web pages.

Finally, all sessions extracted from browsing histories of extremely large number of users are aggregated into the final "user browsing graph". Fig. 4.2 shows the user browsing graph obtained from sample browsing histories of 3 web users given in Fig. 4.1. Here, vertices are associated with total staying times of users on respective pages and the reset probabilities of those pages. Formally, user browsing graph is denoted as $G =< V, W, T, \sigma >$, where $V = \{v_i\}$ denotes vertices (i.e., web pages), $W = \{w_{ij}\}$ denotes edge weights (i.e., transition between web pages), $T = \{T_i\}$ denotes the staying times on the web pages, and $\sigma = \{\sigma_i\}$ denotes the reset probabilities of the web pages $(i, j = 1, ..., n)$. n is the total number of vertices, i.e., $|V|$.

4.2 Continuous-time Markov Model

Given a web browsing graph, assume that there is a random web surfer surfing on this graph. Let X_s denote the page that the surfer is visiting at time s $(s \geq 0)$ and $p_{ij}(s, t)$ denote the probability of the following event:

- the transition of the surfer at page i at time s, to the page j at time t $(t \geq s)$.

Consequently, the transition matrix is defined as $\mathbf{P}(s, t) = (p_{ij}(s, t))$. Now, consider the following two assumptions based on the notation given above:

(i) Given the current state X_s, then the state after X_s depends only on X_s and does not depend on any state visited before X_s. This can be clarified as

$$P(X_t = c \,|\, X_s = a, X_u = b) = P(X_t = c \,|\, X_s = a) \tag{4.1}$$

where s, t, u can be any time series satisfying $0 \leq u \leq s \leq t < +\infty$.

(ii) Surfing behavior does not depend on time points. That is, if the state at time s is X_s and at time $s + s'$ is $X_{s+s'}$ $(s' \geq 0)$, then for any t $(t \neq s)$ if $X_t = X_s$, then $X_{t+s'} = X_{s+s'}$. Mathematically,

$$p_{ij}(s,t) = P(X_t = b \mid X_s = a) = P(X_{t-s} = b \mid X_0 = a) = p_{ij}(0, t-s) \quad (4.2)$$

which means that the transition probability depends only on the length of the transition period. Therefore, we can use $p_{ij}(t)$ (instead of $p_{ij}(s,t)$) to denote the transition probability from state i to state j with a transition period of time t. Similarly, the transition matrix $\mathbf{P}(s,t)$ can be denoted as $\mathbf{P}(t) = (p_{ij}(t))$.

While, the first assumption is known as a *Markov property*, the latter one emphasizes the *time-homogeneity property* of the process. Given that these two assumptions hold, the web surfing process on the user browsing graph can be represented as a continuous-time time-homogenous Markov process $X = (X_s, s \geq 0)$.

For a given continuous-time time-homogenous Markov process, one may obtain a unique stationary probability distribution $\boldsymbol{\pi}$, that does not depend on t, such that for any $t > 0$,

$$\begin{gathered} \boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P} \\ \text{or} \\ \boldsymbol{\pi} = \mathbf{P}^{\mathrm{T}}\boldsymbol{\pi} \end{gathered} \quad (4.3)$$

where $\mathbf{P}^{\mathrm{T}}$ is the transpose of $\mathbf{P}$ and $\boldsymbol{\pi} = (\pi_i)$ is a dense vector of size n [40]. The importance of the stationary probability distribution $\boldsymbol{\pi}$ can be explained as follows. π_i stands for the time spent by the surfer on page i (normalized with the total surfing time), when the total surfing time goes to ∞. Hence, $\boldsymbol{\pi}$ can be perfectly used as a page importance measure.

4.3 Stationary probability distribution of $\mathbf{P}(t)$

The question now is, how to compute the stationary probability distribution of $\mathbf{P}(t)$? Before that, we need to obtain the transition matrix $\mathbf{P}(t)$ itself. Unfortunately, it is a nontrivial job to obtain such information for all possible transition periods. Therefore, BrowseRank algorithm applies the following steps to calculate π:

1. Consider a transition rate matrix $\mathbf{Q}=(q_{ij})$ where $\mathbf{Q}=\frac{d\mathbf{P}}{dt}|_{t=0}$, i.e., $\mathbf{Q}=\mathbf{P}'(0)$. In [40], it has been proven that $\mathbf{P}$ is differentiable with respect to t and there is a one-to-one correspondence between $\mathbf{Q}$ and $\mathbf{P}$, if $\mathbf{P}$'s state space is finite, which is true in our case (i.e, n is finite). Therefore, one may use the Q-process to represent the original continuous-time Markov process X. Here, $\mathbf{Q}=(q_{ij})$ and $q_{ij}=p'_{ij}(0)$ $(1 \leq i,j \leq n)$. Moreover, it is known that $-\infty < q_{ii} < 0$, and $-q_{ii} = \sum_{i \neq j} q_{ij}$. Detailed analysis of Q-process is available in [40].

2. Consider an embedded Markov chain (EMC) [41], a discrete-time Markov process, using the matrix $\mathbf{Q}$ defined above. EMC is obtained using $\mathbf{Q}$ by setting the diagonal positions with 0 values, and non-diagonal positions with the values $-\frac{q_{ij}}{q_{ii}}$.

3. According to Theorem 1 in [6], if the stationary probability distribution the EMC (denoted as $\tilde{\boldsymbol{\pi}}$) and the entries of the matrix $\mathbf{Q}$ are available, then the stationary probability distribution of the Q-process (can be denoted as $\boldsymbol{\pi}$ due to one-to-one correspondence) can be easily computed as

$$\pi_i = \frac{\frac{\tilde{\pi}_i}{q_{ii}}}{\sum_{j=1}^{n} \frac{\tilde{\pi}_j}{q_{jj}}} \tag{4.4}$$

   Proof is available in [41].

4. Since, EMC is a discrete-time Markov process, one can calculate its stationary probability distribution using power method [34]. The only unknown

part is the entries of **Q**. An effective method for the estimation of those entries is proposed in [6].

5. To sum up,

   - The entries of **Q** are estimated using the methods proposed in [6].

   - A discrete-time Markov process, an EMC, is defined based on those estimated values.

   - The stationary probability distribution of the above-defined EMC is computed using power method.

   - The stationary probability distribution of the Q-process is calculated using the entries in **Q** and the stationary probability distribution of EMC.

Although, BrowseRank employs a sophisticated continuous-time Markov model, the basic idea is that the continuous-time Markov model is converted into a discrete-time model and the conventional methods for the computation of the stationary probability distribution of the discrete-time Markov model are used.

Chapter 5

PBRank

The main idea behind PBRank is to combine two different types of feedback, i.e., those provided by the web data and browsing data in a meaningful way. Our goal is to come up with a simple extension to the standard procedure summarized in Section 3, leaving the theoretical foundations unchanged. To this end, we use a transition matrix $\mathbf{X}$ corresponding to the pages in the union of the web and browsing data. $\mathbf{X}$ is a square matrix of size $m \times m$ and is expressed as a linear combination of two other matrices of the same size:

$$\mathbf{X} = \lambda \mathbf{P}'' + (1 - \lambda)\mathbf{B}''. \tag{5.1}$$

Here, $\mathbf{P}''$ is an $m \times m$ version of the final PageRank matrix used in the power method iterations (see Eq. 3.3), i.e., this matrix is created based on the web feedback. In addition, using the user feedback, we define another matrix $\mathbf{B}''$, which we will describe next. λ is a constant in the $[0, 1]$ range and is used to adjust the influence of one type of feedback over the other. The page importance scores can be obtained by finding the principal eigenvector of $\mathbf{X}$ using the power method as usual.

In Eq. 5.1, we form the $\mathbf{B}''$ matrix in a similar fashion to Eq. 3.3:

$$\mathbf{B}'' = \beta \mathbf{B}' + (1 - \beta)\mathbf{e}_n \mathbf{r}^{\mathrm{T}}, \tag{5.2}$$

where β and $\mathbf{r} = (r_i)$ are the counterparts of the α constant and the $\mathbf{t}$ vector in Eq. 3.3, respectively. We use biased teleportation probabilities in $\mathbf{r}$, instead of uniformly setting them to $1/n$ as in $\mathbf{t}$. The teleportation probability r_i of a particular page i is computed as

$$r_i = \frac{1 + T_i}{m + \sum_{j=1}^{m} T_j}, \tag{5.3}$$

where T_i denotes the number of visits to page i by means other than following a link in a page. This way, the jumping behavior of the surfer is biased towards more popular pages. Here, we add one to visit counts for smoothing purposes.

Following the idea in [4], β can be computed as

$$\beta = \frac{\sum_{j=1}^{m} (V_j - T_j)}{\sum_{j=1}^{m} V_j}, \tag{5.4}$$

where V_j denotes the total visit count of page j. The β constant reflects the users' tendency to reach a page by following the hyperlinks in web pages.

The $\mathbf{B}'$ matrix is computed by the following equation:

$$\mathbf{B}' = \mathbf{B} + \mathbf{d}\mathbf{v}^{\mathrm{T}}, \tag{5.5}$$

where $\mathbf{d}$ and $\mathbf{v}$ are defined as before (see Eq. 3.2). The probabilities in the page transition matrix $\mathbf{B} = (b_{ij})$ are set depending on the likelihood of a hyperlink being followed by users. Therefore, the links within a page are not treated equally as in Eq. 3.1. Instead, the transition probability from page i to page j is computed in a biased manner by taking into account the share of the click volume of page j in the overall click volume observed on page i as

$$b_{ij} = \frac{V_{ij}}{\sum_{k \in \mathcal{L}_i} V_{ik}}, \tag{5.6}$$

where V_{ij} is the click volume from page i towards page j.

PBRank can be considered as a variant of BrowseRank since both techniques use page visit probabilities extracted from browsing data. In practice, one may prefer PBRank to BrowseRank because of the following reasons. First, as we will show later in Section 8, PBRank achieves a better coverage of web pages than BrowseRank due to the use of web data in scoring computations, i.e., a larger number of pages receive non-zero scores. Second, PBRank is a relatively straightforward extension to PageRank. Hence, its implementation is easier than BrowseRank, which employs a relatively more sophisticated continuous-time Markov model. Finally, the transition probabilities computed in PBRank are accurate values computed over actual user clicks on links. The transition probabilities computed in BrowseRank, however, are only approximations because they are computed based on a timestamp-sorted sequence of page visits in user sessions, not the links that are actually followed by users. Given that many users browse the Web by opening multiple browser tabs [32] and concurrently following links in different tabs, a time-ordered sequence of page visits may not be sufficient to obtain the actual transitions between pages. Hence, the transition probabilities computed in BrowseRank may not reflect the true surfing patterns of users.

We note that the existence of a solution is guaranteed since the $\mathbf{X}$ matrix is irreducible and aperiodic because both summation terms in Eq. 5.1 already have these properties. When $\lambda = 0$ or $\lambda = 1$, $\mathbf{X}$ may not be row-stochastic, but this does not prevent the convergence of iterations. If λ is set to zero or one in Eq. 5.1, PBRank reduces to a discrete-time variant of BrowseRank or PageRank, respectively. As we will see in Section 8, the best ranking quality will be obtained for λ values close to zero.

Chapter 6

Evaluation Metrics

One can obtain different ranking techniques using our hybrid ranking model by setting the λ parameter with values in the $[0, 1]$ range (see Eq. 5.1). However, two of those ranking techniques obtained using corner values of the range (i.e., 0 and 1) can be treated as special cases. While the $\lambda = 0$ case produces a ranking method that exploits only browsing behavior of web users, the ranking method for $\lambda = 1$ case uses only hyperlink information. For any other λ value $(0 < \lambda < 1)$, our hybrid ranking model generates a ranking technique that uses both types of feedback.

In order to show the effectiveness of combining two types of feedback, we evaluate our hybrid ranking model by comparing the quality of the ranking techniques for λ in the $(0, 1)$ range with the quality of two ranking techniques for $\lambda = 0$ and $\lambda = 1$. Thus, if ranking methods for λ in the $(0, 1)$ range perform better than the ranking techniques for $\lambda = 0$ and $\lambda = 1$, we can argue that combining data sources leads to a better importance ranking. Here, "comparison of the qualities of ranking techniques" needs more detailed explanation.

Our initial motivation to design a hybrid ranking model was to overcome the limitations of using single type of feedback. While the main limitation of exploiting only browsing data is the low page coverage, the main problem of the hyperlink structure is its vulnerability to malicious intent (i.e., link farms). Therefore, we quantify two different aspects of the hybrid ranking model: coverage quality and ranking quality. The former aspect refers to the ability of the hybrid model to compute a non-zero score for many pages. The second aspect refers to the ability of the hybrid model to rank "important" pages at higher ranks. Herein, the actual importance of a web page is taken from the ground-truth ranking which is explained in next Section.

Next three sections describe the ground-truth ranking, the coverage quality metric and the ranking quality metric, respectively.

6.1 Ground-truth Ranking

We define two quality metrics for evaluation purposes of the hybrid ranking model. Both of the metrics rely on a ground-truth ranking of the web pages. We assume that this ground-truth ranking represents the actual importance ranking of the web pages.

The question now is, how to construct a ground-truth ranking? It is a non-trivial job to obtain a reliable ground-truth data for ranking problems. Even so, in our context at least, ground-truth ranking can be generated from several data sources including search result click logs, web browsing logs and web traffic analytics.

(i) **Search result click logs.** One of the reliable sources for the ground-truth ranking is the click logs of web search results. Here, the click amount of a page in search results stands for page's importance, i.e., the more a page is clicked in search results, the greater its importance. Although, the click probability of a page in search results depends on the relevance of the page to the search query, the click information, when aggregated over many different queries, gives a notion of fair page importance ranking.

(ii) **Web browsing logs.** Another ground-truth importance ranking of web pages can be obtained by sorting the pages according to their visit counts in the browsing data. The more a page is visited in browsing logs, the greater its importance. Again, note that, the variety of visited pages is highly relevant to the interests of an individual web user. However, as the browsing information is aggregated over many different users, the visit count of a page becomes a reasonable importance measure of the page.

(iii) **Web traffic analytics.** There are services that monitor the browsing activities of millions of worldwide internet users using different types of toolbars and add-ons for modern internet browsers. Two well-known examples are Quantcast[1] and Alexa[2]. They provide a daily updated ranking of top one million most popular web sites according to the network traffic. In some sense, this ranking is similar to the ranking obtained from web browsing logs, but it has much larger user community.

Among three options mentioned above, in our context, ground-truth rankings obtained from sources (ii) and (iii) create an unfair bias towards the rankers that directly exploit the browsing behavior of the web users (i.e., rankers for $\lambda < 1$). In this work we focus on the impact of the generated page rankings on web search. Therefore, ranking obtained from (i) forms a more natural basis.

6.2 Coverage Quality

In order to evaluate the coverage quality aspect of a given ranking technique we define a page coverage metric χ. A simple motivation behind this coverage metric is to find out the fraction of ground-truth pages which are accessible (i.e., can be positively scored) by the given ranking technique. This fraction can be calculated in a straightforward way. First we introduce some notation, then we formally define the above-explained page coverage metric.

Let ρ denote the page ranking technique and $\mathcal{R}^{\rho}$ denote the set of pages which are positively scored by this technique. Similarly, let ρ^{*} be an oracle ranker that

[1] Quantcast.com homepage, `https://www.quantcast.com/`

[2] Alexa.com homepage, `http://alexa.com`

has an access to ground-truth importance values for a set $\mathcal{R}^*$ of pages. Here, $\mathcal{R}^*$ is the set of ground-truth pages and we assume that the oracle ranker computes positive scores for every page in $\mathcal{R}^*$.

Given these definitions, the page coverage χ^ρ of a ranking technique ρ is defined as

$$\chi^\rho = \frac{|\mathcal{R}^\rho \cap \mathcal{R}^*|}{|\mathcal{R}^*|}. \tag{6.1}$$

For example, let ρ_1 and ρ_2 be two ranking methods that rank the following sets of pages: $\mathcal{R}^{\rho_1} = \{a, b, d\}$ and $\mathcal{R}^{\rho_2} = \{a, e\}$. Assume that the ground-truth pages are $\mathcal{R}^* = \{a, b, c\}$. Then, we have $\chi^{\rho_1} = \frac{2}{3}$ and $\chi^{\rho_2} = \frac{1}{3}$. Obviously, higher coverage values indicate better coverage.

6.3 Ranking Quality

Our second evaluation metric quantifies the ranking quality aspect of the hybrid model. Given a page ranking technique ρ and the importance ranking $\mathcal{R}^\rho$ produced by ρ. There are several ways to evaluate the quality of $\mathcal{R}^\rho$. One approach is to calculate the rank correlation between $\mathcal{R}^\rho$ and the ground-truth importance ranking. Another approach is to combine $\mathcal{R}^\rho$ with a separate query-dependent relevance ranking (e.g., BM25 [42])and use query-dependent evaluation techniques based on human relevance judgements.

First, we briefly explain well-known evaluation techniques. Then, we state the drawbacks of existing methods and devise our ranking quality metric.

6.3.1 Rank Correlation

Kendall's tau. Kendall's τ is a rank correlation coefficient that was first introduced by M. G. Kendall in 1938 [43]. It was originally addressed to solve the problem of comparing two different rankings (produced by two separate observers) of the same set of individuals. Since significant part of the research in Information Retrieval is concerned with ranked lists of items, τ is widely used in IR as a rank correlation statistic [44].

The correlation coefficient τ varies in the $[-1, 1]$ range. The higher (lower) is the value of τ, the stronger (weaker) is the relevance between two rankings. Thus, $\tau = 1$ occurs when two rankings are exactly same, and $\tau = -1$ occurs when two rankings are exactly inverted.

Correlation is calculated as follows. Let N be the number of individuals, C be the number of pairs of individuals that are in the same order in both rankings, and D be the number of pairs of individuals that are in the reverse order in both rankings. Then, Kendall's τ is defined as

$$\tau = \frac{C - D}{C + D} = \frac{C - D}{\binom{N}{2}} = \frac{2(C - D)}{N(N - 1)}$$

where the denominator $C + D$ (i.e., the total number of all possible pairs) is used for normalization. As alluded to earlier, when all pairs are in the same (reverse) order in both rankings, D (C) equals to 0, and τ equals to 1 (-1).

As an example, consider a set of four individuals, numbered from 1 to 4, and three arbitrary rankings of those individuals: $\sigma_1 = <1, 2, 3, 4>$, $\sigma_2 = <2, 1, 3, 4>$ and $\sigma_3 = <4, 1, 3, 2>$. It is clear that the distance between σ_1 and σ_2 would be much less than the distance between σ_1 and σ_3. Indeed, τ values reports the same results: while τ between σ_1 and σ_2 is 0.66, τ between σ_1 and σ_3 is -0.33.

Spearman's footrule distance. Denoted as r_s, Spearman's footrule distance is simply the l_1 distance between two rankings [45, 46]

$$r_s = \sum_{i=1}^{N} |\sigma_1(i) - \sigma_2(i)|$$

where $\sigma_1(i)$ and $\sigma_2(i)$ are ranks of i^{th} individual in the first and the second rankings, respectively. Unlike τ, the lower (higher) is the value of r_s, the stronger (weaker) is the relevance between two rankings. In order to be consistent with τ, r_s can be normalized into the $[-1, 1]$ range.

As an example consider three rankings described for τ's explanation. While r_s between σ_1 and σ_2 is 2, r_s between σ_1 and σ_3 is 6. As expected, distance between σ_1 and σ_2 is less than the distance between σ_1 and σ_3.

Comparing partial rankings. Both of τ and r_s operate on fully ranked lists. Unfortunately, there are cases where comparison techniques for partially ranked lists are required, simply because the full ranking is not available due to ties or because it is very expensive to construct one. In [46], τ and r_s are extended for comparing partially ranked lists.

A partial ranking σ is composed of ordered *buckets*, where *bucket* is a set of tied items. σ becomes fully ranked when every bucket contains exactly one item, otherwise it is a partial ranking. In a given partial ranking σ, if a bucket $\mathcal{B}_i$ is ranked higher than some other bucket $\mathcal{B}_j$, then, it is safe to assume that all items in $\mathcal{B}_i$ are ranked higher than all items in $\mathcal{B}_j$.

Let σ_1 and σ_2 be two partial rankings. For any (x, y) pair of items, consider the following three cases in which x and y can appear:

(i) x and y are in different buckets in both rankings.

(ii) x and y are in same buckets in both rankings.

(iii) x and y are in same buckets in one of the rankings and are in different buckets in the other ranking.

All three cases are penalized with some pre-defined penalties. Penalties are defined similar to those which are implicitly used in Kendall's τ. Let $\mathcal{B}^1(x)$, $\mathcal{B}^1(y)$, $\mathcal{B}^2(x)$ and $\mathcal{B}^2(y)$ denote the bucket of x in σ_1, bucket of y in σ_1, bucket of x in σ_2 and the bucket of y in σ_2, respectively. Then, for case (i), $\tau'_{xy} = 0$ (τ'_{xy} denotes the penalty for (x, y) pair) if $\mathcal{B}^1(x)$ and $\mathcal{B}^1(y)$ are in the same order as $\mathcal{B}^2(x)$

and $\mathcal{B}^2(y)$, otherwise $\tau'_{xy} = 1$. For case (ii), $\tau'_{xy} = 0$, because x and y are tied in both rankings. For case (iii), $\tau'_{xy} = p$, where p $(0 \leq p \leq 1)$ is a fixed parameter. Finally, total distance between two partial rankings is the sum of all possible τ'_{xy} penalties.

To clarify, consider the following two sample partial rankings $\sigma_1 = <\{1,4\},\{2,5\},\{3\}>$, $\sigma_2 = <\{2\},\{3,5\},\{1,4\}>$ and take $p = 1/2$. Pairs that suit the case (i) are $\{(1,2),(1,3),(1,5),(2,3),(2,4),(3,4),(4,5)\}$, and accumulate a penalty of 6 in total. There is only one pair that suits the case (ii): $(1,4)$. Remaining two pairs suit the case (iii): $\{(2,5),(3,5)\}$. Last two pairs have a penalty of $2 * p = 2 * 1/2 = 1$ in total. Total penalty is 6 + 0 + 1 = 7, which is the distance between σ_1 and σ_2.

6.3.2 Query-Dependent Evaluation

Second evaluation approach simulates the behavior of search engines by combining $\mathcal{R}^\rho$ with a query-dependent relevance ranking (e.g., BM25 [42]). Then, for a given set of search queries, search engine's ability to retrieve highly relevant and important pages is measured. Well-known techniques for such measurements are, but not limited with: Precision at n (P@n) [47], Mean Average Precision (MAP) [47] and Discounted cumulative gain (DCG) [48]. Indeed, BrowseRank is evaluated using these measures.

P@n and MAP. Consider a ranked list of search results for a given query and assume that relevance judgements for all query-result pairs are available. Then, P@n is defined as

$$P@n = \frac{r}{n}$$

where r is the number of relevant pages ranked among top n pages of the search result list.

In order to describe MAP, we first would like to explain *average precision*

(AP). For a given search query and its ranked search result list, AP is the average of P@n's computed after retrieval of every relevant page. Then, MAP is the mean of APs of all queries.

DCG. Main property of DCG measure is that it devaluates high-ranked pages (i.e., less valuable pages) by applying discount factors to their relevance scores. DCG is computed as follows. Given a ranked list of N pages and their relevance scores (i.e., gain values). Relevance scores vary from 0 to 3 (3 denotes high relevance, 0 denotes no relevance).

First, ranked list is converted into a *gain vector*, G', where each page is replaced with its relevance score. For example, consider a 5-page search result list in which first page has a relevance score of 3, third page has a relevance score of 2, second and fifth pages have relevance scores of 1, and fourth page is irrelevant (i.e., has relevance score of 0). Then, $G' = <3, 1, 2, 0, 1>$.

Next, *cumulative gain vector*, CG', is defined as

$$CG[i] = \begin{cases} G[1], & \text{if } i = 1 \\ CG[i-1] + G[i], & \text{otherwise.} \end{cases}$$

For the sample G' given above, CG' will be $<3, 4, 6, 6, 7>$.

Finally, we define *discounted cumulative gain vector*, DCG', as

$$DCG[i] = \begin{cases} CG[i], & \text{if } i<b \\ DCG[i-1] + \dfrac{G[i]}{\log_b i}, & \text{if } i \geq b. \end{cases}$$

where the base of the logarithm, b, controls how much a page appearing at a lower rank is penalized. Let $b = 2$. From sample G' given above, we obtain $DCG' = <3, 4, 5.26, 5.26, 5.76>$.

Normalized-DCG (i.e., NDCG) measure is obtained by dividing DCG' by DCG'_I, where DCG'_I is the discounted cumulative gain vector of the ideal ranking.

Here, ideal ranking is the ranking where pages with relevance score of 3 are ranked higher than all other pages, pages with relevance score of 2 are ranked higher than all pages with relevance scores of 1 or 0, and pages with relevance scores of 1 are ranked higher than the pages with relevance scores of 0. Ideal ranking of the sample G' is $G'_I = <3, 2, 1, 1, 0>$.

6.3.3 Our Ranking Quality Metric

Both τ and r_s metrics have two important drawbacks. First problem is that they operate on fully ranked lists. In our case, we have partial rankings (i.e., some pages in ground-truth ranking are not ranked by ρ and vice versa). Second limitation is that these two metrics penalize the ranking errors made in the upper part and lower part of the ranking with the same penalty. In our problem, the correctness of the ranking's head (i.e., top pages) is much more important than the correctness of its tail. Methods presented in [46] for comparing partially ranked lists also fail to handle the second problem. One more important reason we do not use τ is because of its computational time complexity. A naive algorithm that checks every possible pair of pages has a time complexity of $O(N^2)$, where N is the number of pages in the data set. This is is unacceptable in our case where N is around 200 millions. In [49], an efficient method for the calculation of τ is presented. It is based on the Merge Sort algorithm and has $O(N \log N)$ time complexity. Unfortunately, it's implementation is not straightforward. Therefore, in our evaluations, we prefer not to use τ, r_s or their extended versions for partially ranked lists.

Although P@n, MAP and DCG (or NDCG) metrics that obey the second query-dependent evaluation approach are commonly used in IR, they necessitate user studies to obtain the relevance judgements among search queries and web pages. Instead of using metrics that rely on relevance judgements, we prefer to use fully automated evaluation methods because of the following reason. We conduct our experiments using data sets in the scale of hundreds of millions of web pages (details of the data sets are explained in Chapter 6). In order to satisfy the needs of the experiments on such large data sets, one should perform large scale user studies for big variety of search queries. Performing such user studies is very challenging simply because of the human factor. One more reason we do not use DCG (or NDCG) is that it heavily weights the top pages of the ranking and highly devaluates the later retrieved pages. In our case, this is not very meaningful because our rankings are very long and tail pages should not be ruled out. Therefore, in our evaluations, we prefer not to use P@n, MAP or DCG (or NDCG).

Due to above-mentioned reasons we devise our own quality metric that carefully takes into account the following aspects;

(i) Weight of the penalties given for the errors made in the upper part of the ranking should be higher than those which are given for the errors made in the lower part.

(ii) Popularity of a page (i.e., click count of a page) in the ground-truth ranking should be taken into account.

(iii) Meaningful results should be produced for the rankings with a large number of tail pages (in the scale of hundreds of millions of pages).

(iv) The last but not the least: implementation should not be too complicated and the computational time complexity should be acceptable when the metric is used for large scale data sets.

Now, we define a ranking quality metric. Let ρ denote the page ranking technique and $\mathcal{R}^{\rho}$ denote the ranking it produces (all pages in $\mathcal{R}^{\rho}$ are positively scored by ρ). Let ρ^{*} be an oracle ranker that ranks all pages in $\mathcal{R}^{\rho}$ in the best possible way ("the best possible way" will be explained later). Let $\mathcal{R}^{*}$ denote the ground-truth ranking, where every page has a positive visit count, i.e., $\mathcal{R}^{*}$ is a list of pages sorted in descending order of their visit counts.

First, we define a metric $\mathcal{C}^{\mathcal{R}}$ using recursive function

$$\mathcal{C}^{\mathcal{R}}(k) = \begin{cases} 0, & \text{if } k = 0; \\ \mathcal{C}^{\mathcal{R}}(k-1) + I(\mathcal{R}_k), & \text{if } 1 \leq k \leq |\mathcal{R}|; \\ \mathcal{C}^{\mathcal{R}}(|\mathcal{R}|), & \text{if } k > |\mathcal{R}|. \end{cases}, \tag{6.2}$$

where $\mathcal{R}_k$ denotes the k-th ranked page in a given ranking $\mathcal{R}$ of pages and $I(p)$ denotes the page p's visit count in the ground-truth ranking. We assume that $I(p)=0$ if $p\notin\mathcal{R}^*$. Here, $\mathcal{C}^{\mathcal{R}}(k)$ calculates the sum of visit counts of top k pages in $\mathcal{R}$. Moreover, it gives us some hints about the following question: how important are those top k pages in the ground-truth ranking?

Although $\mathcal{C}^{\mathcal{R}}(k)$ gives us a useful information about the quality of top k pages, it does not report anything about the quality of their rankings. This is explained with the following example. Top k pages in $\mathcal{C}^{\mathcal{R}}(k)$ can be ordered in $k!$ different ways (i.e., it has $k!$ permutations). $\mathcal{C}^{\mathcal{R}}(k)$ values for all those orders are equal. Therefore, for a given $\mathcal{C}^{\mathcal{R}}(k)$ value, it is impossible to make any assumptions about the quality of the ranking of top k pages, just by looking at $\mathcal{C}^{\mathcal{R}}(k)$. To this end, we devise another quality metric $\phi^{\mathcal{R}}$ that uses $\mathcal{C}^{\mathcal{R}}$ and is able to quantitatively report both the ranking quality and the importances of the top k pages in $\mathcal{R}$:

$$\phi^{\mathcal{R}}(k) = \begin{cases} 0, & \text{if } k=0; \\ \phi^{\mathcal{R}}(k-1) + \mathcal{C}^{\mathcal{R}}(k-1) + \frac{I(\mathcal{R}_k)}{2}, & \text{if } 1\le k\le|\mathcal{R}|; \\ \phi^{\mathcal{R}}(k-1) + \mathcal{C}^{\mathcal{R}}(k-1), & \text{if } k>|\mathcal{R}|. \end{cases}, \tag{6.3}$$

In order to explain the idea behind the $\phi^{\mathcal{R}}$ metric, we visualize it in a two-dimensional graph.

For a given ranking $\mathcal{R}$, we define a two-dimensional graph in which k is plotted on the X axis and $\mathcal{C}^{\mathcal{R}}(k)$ is plotted on the Y axis. For every possible k ($0\le k\le|\mathcal{R}|$), $<k,\mathcal{C}^{\mathcal{R}}(k)>$ pair corresponds to a single point (denoted as p_k) in the two-dimensional graph. Here, if $I(\mathcal{R}_k)=0$, then the point p_k is to the **east** of the point p_{k-1}, because $\mathcal{C}^{\mathcal{R}}(k)=\mathcal{C}^{\mathcal{R}}(k-1)$. Similarly, if $I(\mathcal{R}_k)\neq 0$, then the point p_k is to the **northeast** of the point p_{k-1}, because $\mathcal{C}^{\mathcal{R}}(k)>\mathcal{C}^{\mathcal{R}}(k-1)$. Fig. 6.1 shows the two-dimensional graph and corresponding points for the $\mathcal{R}$ obtained using the sample ranker ρ_1 given in Table 6.1. Next, for every k ($1\le k\le|\mathcal{R}|$), we connect two points p_{k-1} and p_k with straight line. As a result, we obtain a curve that starts at p_0 and ends at $p_{|\mathcal{R}|}$. This is visualized in Fig. 6.2. Finally, $\phi^{\mathcal{R}}(k)$ equals to the area under the curve created by connecting consecutive points starting from p_0 and finishing at p_k.

We note that, the best possible curve (which yields the largest $\phi^{\mathcal{R}}$ value) can be obtained from the ground-truth ranking $\mathcal{R}^*$. Therefore, we assume that the oracle ranker produces a ranking identical to $\mathcal{R}^*$. In the rest of this work, $\mathcal{R}^*$ stands for both ground-truth ranking and the ranking obtained from the oracle ranker.

Before analyzing the effectiveness of this metric, we define the relative quality $\Phi^\rho(k)$ of a given ranking $\mathcal{R}^\rho$ at rank k with respect to the best possible ranking $\mathcal{R}^*$ as

$$\Phi^\rho(k) = \frac{\phi^{\mathcal{R}^\rho}(k)}{\phi^{\mathcal{R}^*}(k)}. \tag{6.4}$$

Here, Φ^ρ is the normalized version of ϕ^ρ. This is necessary, because it is more convenient to produce numerical evaluation results in the $[0, 1]$ range.

Next, we briefly explain how Φ handles all of the four aspects stated above. The devised Φ metric emphasizes the discovery of important pages (i.e., with high click counts) at early ranks, as the $\mathcal{C}^{\mathcal{R}}(k)$ continues to contribute to the value of the metric at all ranks following k. This property handles the aspect (i). The aspect (ii) is already handled by $\mathcal{C}^{\mathcal{R}}$. The problem (iii) of handling long tails of rankings is also resolved by ϕ, because the curve of the shorter ranking is extended in the horizontal direction in order to catch the longer rankings's size. Regarding the last aspect about the implementation simplicity and the computational time complexity, calculation of ϕ requires a simple linear pass over the ranking $\mathcal{R}$ and simple computations. The functioning of ϕ resembles the ROC analysis and the area under the curve metric [50].

Fig. 6.3, 6.4, 6.5 and 6.6 visualize the $\phi^{\mathcal{R}}$ calculations for the rankings given in Table 6.1. Fig. 6.7 plots the $\phi^{\mathcal{R}}$ calculations of all four rankings on the same plot.

Table 6.1: Rankings of four different rankers and their evaluations. The ground-truth ranking use is $\mathcal{R}^* = <a, b, c, d, g>$, where ground-truth importances of pages a, b, c, d and g are 100, 60, 30, 5 and 2, respectively.

ρ	$\mathcal{R}^\rho$	$\phi^{\mathcal{R}}$	
ρ_1	$\mathcal{R}^{\rho_1} = \mathcal{R}^1 = <a, b, e, f, c, d>$	$\phi^{\mathcal{R}_1} = 867.5$	$\Phi^{\mathcal{R}_1} = 0.925$
ρ_2	$\mathcal{R}^{\rho_2} = \mathcal{R}^2 = <a, d, b, f, e, c>$	$\phi^{\mathcal{R}_2} = 797.5$	$\Phi^{\mathcal{R}_2} = 0.851$
ρ_3	$\mathcal{R}^{\rho_3} = \mathcal{R}^3 = <e, f, d, c, b, a>$	$\phi^{\mathcal{R}_3} = 232.5$	$\Phi^{\mathcal{R}_3} = 0.248$
ρ^*	$\mathcal{R}^* = <a, b, c, d, e, f>$	$\phi^{\mathcal{R}^*} = 937.5$	$\Phi^{\mathcal{R}^*} = 1.000$

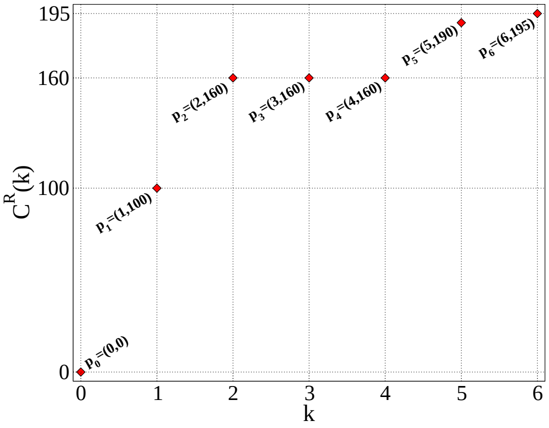


Figure 6.1: Two-dimensional graph and points for the $\mathcal{R}$ obtained using ρ_1.

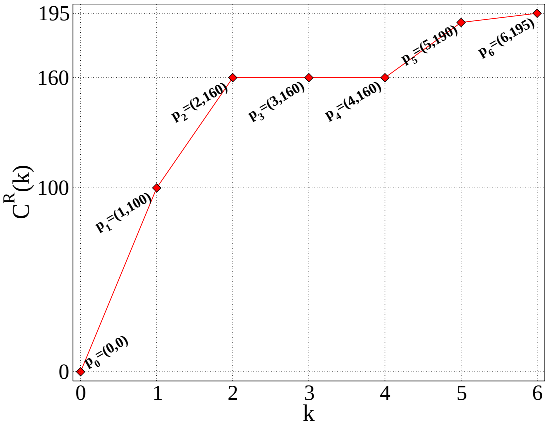


Figure 6.2: Two-dimensional graph and curve for the $\mathcal{R}$ obtained using ρ_1

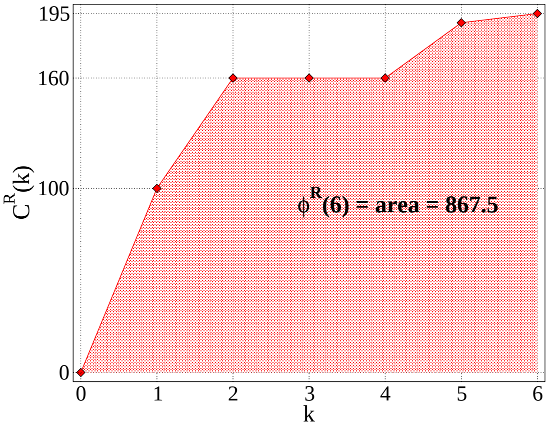


Figure 6.3: $\phi^{\mathcal{R}}$ calculation for ρ_1.

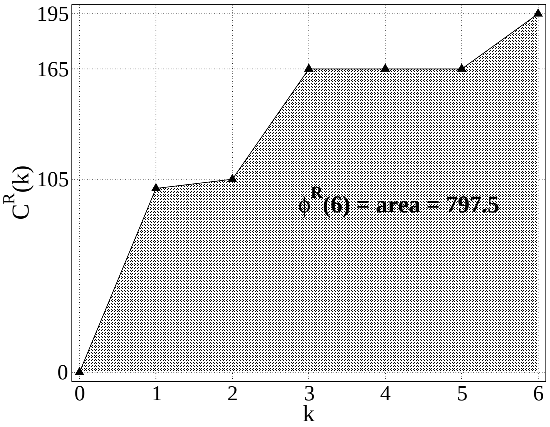


Figure 6.4: $\phi^{\mathcal{R}}$ calculation for ρ_2.

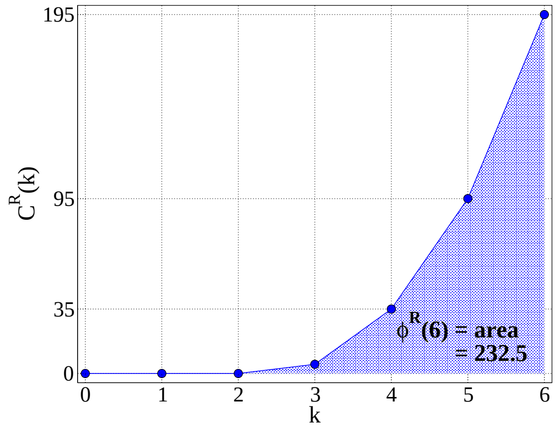


Figure 6.5: $\phi^{\mathcal{R}}$ calculation for ρ_3.

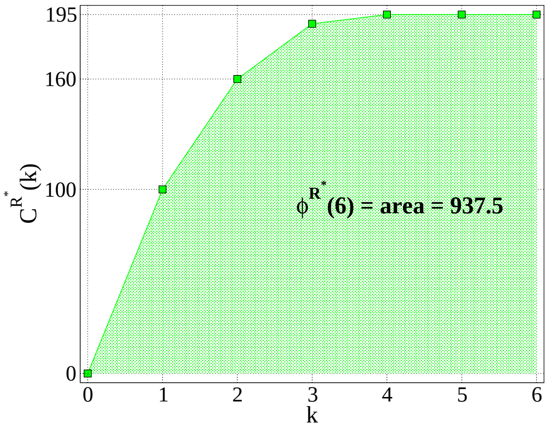


Figure 6.6: $\phi^{\mathcal{R}^*}$ calculation for the oracle ranker.

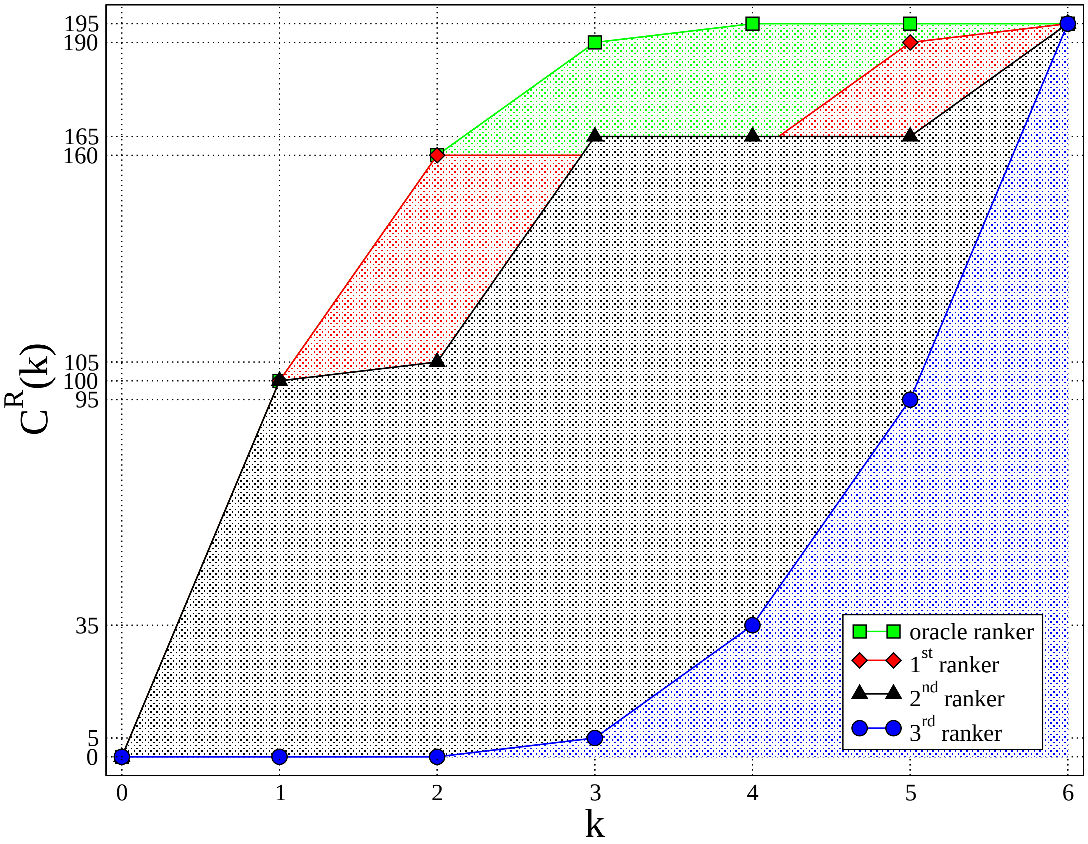


Figure 6.7: $\phi^{\mathcal{R}}$ calculations for all rankers.

Chapter 7

Data and Setup

In this chapter, we describe the computing platform, the URL normalization technique and the datasets we use for experiments.

7.1 Computing platform

Our experimental framework consists of various computational steps such as data preprocessing, scoring computations and performance evaluation. Every step requires a large computational work because of the size of the datasets. To this end, all steps of the experiments are carried out on a large computer cluster composed of thousands of processors, running Linux. Specifically, we use Apache Hadoop framework[1] for distributed computations and HDFS[2] for distributed data storage.

The codes are written in Pig Latin scripting language [51] on top of the Apache Hadoop framework. Pig Latin is a scripting language that plays a similar role over Hadoop as SQL plays over relational databases. Moreover, Pig Latin is widely used for large scale data analysis purposes both in industry and academia. Thus,

[1] Apache Hadoop, `http://hadoop.apache.org`.

[2] Hadoop Distributed File System, `http://hadoop.apache.org`.

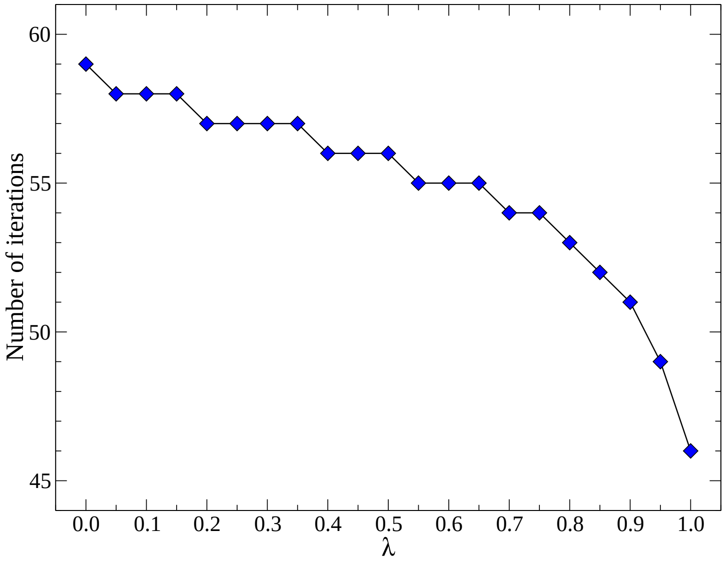


Figure 7.1: Number of power iterations before convergence for varying values of λ.

it provides a perfect environment for our computations on large-scale datasets.

As mentioned in Chapter 3, for scoring computations we use the power method [34] and iterate until the convergence to a solution. We define the convergence rule as follows. Let $\mathbf{p}^k$ denote the score vector at iteration k and p_i^k denote the score of page i at iteration k. The score difference between two consecutive iterations is defined as

$$\Delta_k = \sum_{i=1}^{N} |p_i^k - p_i^{k-1}|. \tag{7.1}$$

We assume that the iterations converge when $\Delta_k < 10^{-6}$ holds. In other words, iterations converge when the L1-norm of the PageRank vector is less than 10^{-6}.

Even though we have not paid special attention to optimize the execution time of scoring computations, the iterations converge to a solution in several hours in the worst case. Fig. 7.1 displays the number of iterations needed before convergence for varying λ values (see Eq. 5.1). In general, we observe faster convergence as λ increases.

Table 7.1: URL Normalization examples.

$\mathcal{U}$	$\tilde{\mathcal{U}}$
`ftp://127.127.127.0/index.html`	ignored
`ftp://www.abc.com/index.html`	ignored
`http://www.abc.com:80/index.html`	`http://abc.com/index.html`
`https://www.abc.com:9401/index.php`	`https://abc.com:9401/index.php`
`http://com`	ignored
`http://www.abc.def.co.uk`	`http://abc.def.co.uk`
`http://www.abc.com/path/name.php`	`http://abc.com/path/name.php`

7.2 Handling of URLs

To be consistent, we use the same URL normalization technique in preprocessing of all types of datasets. For a given URL string $\mathcal{U}$, we obtain the normalized form of $\mathcal{U}$, denoted as $\tilde{\mathcal{U}}$, by obeying the following rules.

- We do not consider $\mathcal{U}$ if it does not conform to the URL definition specified by RFC 1738.[3] We ensure the RFC 1738 compliance by checking if the constructor of the `java.net.URL` class throws a `MalformedURLException` for the given URL.
- We do not consider $\mathcal{U}$ if it is represented in IP address format.
- We do not consider $\mathcal{U}$ if its length is less than 11 characters or greater than 5,000 characters. The lower limit ensures that the domain name contains at least 1 character, because, in the worst case, the protocol (e.g., "`http://`") and the shortest possible domain extension (e.g., "`.ca`") occupy 10 characters and leave 1 last character for domain name. On the other hand, the upper limit prevents us from spammy URLs (usually extracted from HTML contents of the crawled web pages).
- We remove the `www` prefixes from $\mathcal{U}$.
- We remove the default port 80 (if present) from the host part of $\mathcal{U}$.
- We consider $\mathcal{U}$ only if it served by the HTTP and HTTPS protocols.

Table 7.1 presents some examples for the URL normalization we use.

[3] RFC 1738, `http://www.ietf.org/rfc/rfc1738.txt`.

Table 7.2: Size of the browsing data.

Total page visits	1,919,657,987
Page visits by following a link (i.e., transportation)	1,189,735,491
Page visits by typing the URL (i.e., teleportation)	729,922,496

7.3 Web page collection

The web page collection we use is crawled in late 2011 by a commercial web search engine. The compressed version of this web snapshot occupies around 50 terabytes and contains around 6.5 billion web pages. We use the JSOUP HTML parser library to parse the content of each web page.[4] Then, from the parsed content we extract the links located inside the HTML `<a>` tags with `href` attributes.

Due to the difficulties involved in parsing web pages written in the CJK (Chinese, Japanese and Korean) languages, we exclude such pages from further consideration. Moreover, self-links are removed and identical out-links in a page are contracted into a single out-link. We convert the remaining pages and links into a web graph and further compress this graph to obtain a host-level graph of the Web. For simplicity, we limit the maximum number of out-links of a host to five million. If the links (corresponding edges in the constructed graphs) are weighted, we consider five million out-links with the largest weights. In the rest of the thesis, we use this host-level graph, which includes about 230 million unique web hosts with 1.5 billion inter-host links.

7.4 Browsing data

We obtain the web browsing data through Yahoo! toolbar. In our experiments, we use only the browsing data acquired from users who explicitly gave permission for their page views to be logged. In total, our data contains around two billion page visits (exact size is given in Table 7.2), performed by users all around the

[4] JSOUP homepage, `http://jsoup.org`.

Table 7.3: Sample user browsing history used by PBRank.

	$VISIT_1$	$VISIT_2$	$VISIT_3$
URL	`www.aaa.com`	`www.bbb.com`	`www.ddd.com`
REF. URL	-	`www.ccc.com`	-
TIME	2013-08-01, 17:30:05	2012-08-01, 17:30:05	2011-08-01, 17:30:05
COUNTRY	us	tr	uk
OS	OS X 10.8.4	WINDOWS XP	UBUNTU 11.10
BROWSER	Safari 6.0.5	IE 9.0	-
⋮	⋮	⋮	⋮

world. The browsing data is obtained in a period right after the web collection is crawled.

In our toolbar logs, each page visit is stored with some meta-data related to the page and the user who visited the page, including the fields *URL*, *REFERRER URL*, *TIME*, *COUNTRY*, *OS* and *BROWSER*. Here, the field *URL* contains the URL of the visited page, the field *TIME* contains the time at which the page is visited and the field *COUNTRY* contains the name of the country where the user is physically located. Moreover, the fields *OS* and *BROWSER* contain information about the Operating System and the Internet Browser which are used during the page visit, respectively. Finally, if the user has reached the page by clicking a link in another page (i.e., user has transported to the current page), the field *REFERRER URL* contains the URL of the referrer page. If the *REFERRER URL* is not available, this indicates that the user manually typed the URL into the address bar of the browser or clicked a bookmark link (i.e., user has teleported to the current page). Table 7.2 gives the number of page visits occurred by following a link (i.e., *REFERRER URL* is available) and typing the URL (i.e., *REFERRER URL* is not available). Sample toolbar logs are shown in Table 7.3.

At this point, we note that the format of our user browsing data differs from the format of the user browsing data used by BrowseRank (explained in Chapter 4). In our case, toolbar logs contain the actual referrer URL which directly gives us the edges of the user browsing graph. Unfortunately, BrowseRank estimates the edges of the user browsing graph by constructing user browsing sessions.

Table 7.4: Sample query log.

	$QUERY_1$	$QUERY_2$	$QUERY_3$
QUERY STRING	"metu"	"bilkent"	"odtu"
TIME	08-01-13,13:00	08-01-12,05:00	08-01-11,17:34
CLICKED URLS	metu.edu.tr odtu.edu.tr	bilkent.edu.tr bilkenthotel.com.tr metu.edu.tr	metu.edu.tr odtu.edu.tr

Table 7.5: Ground-truth ranking obtained from sample query log.

URL	click count
metu.edu.tr	3
odtu.edu.tr	2
bilkent.edu.tr	1
bilkenthotel.com.tr	1

To this end, the user browsing graph we construct is more accurate than the one constructed by BrowseRank.

7.5 Click data

Due to the reasons explained in Section 6.1, as a ground-truth in evaluation of PBRank, we use large-scale user feedback in the form of clicks issued on web search results. To this end, we use a random sample of over 700,000 clicks obtained from the query logs of a commercial web search engine in a time period that follows the acquisition of the browsing data. Out of those 700,000 clicks, we extract around 170,000 unique URLs. The query log contains information about the query string, the time when the query is submitted to the search engine, the URLs clicked by the user who submitted the query, and some profile information about the user. Here, the user may have clicked to multiple URLs from the search result list returned for a single search query. Finally, we aggregate the click counts of all web pages over all clicks in the data, and obtain a ground-truth importance ranking of web pages by sorting them in decreasing order of their click counts. Table 7.4 shows sample query logs and Table 7.5 presents the

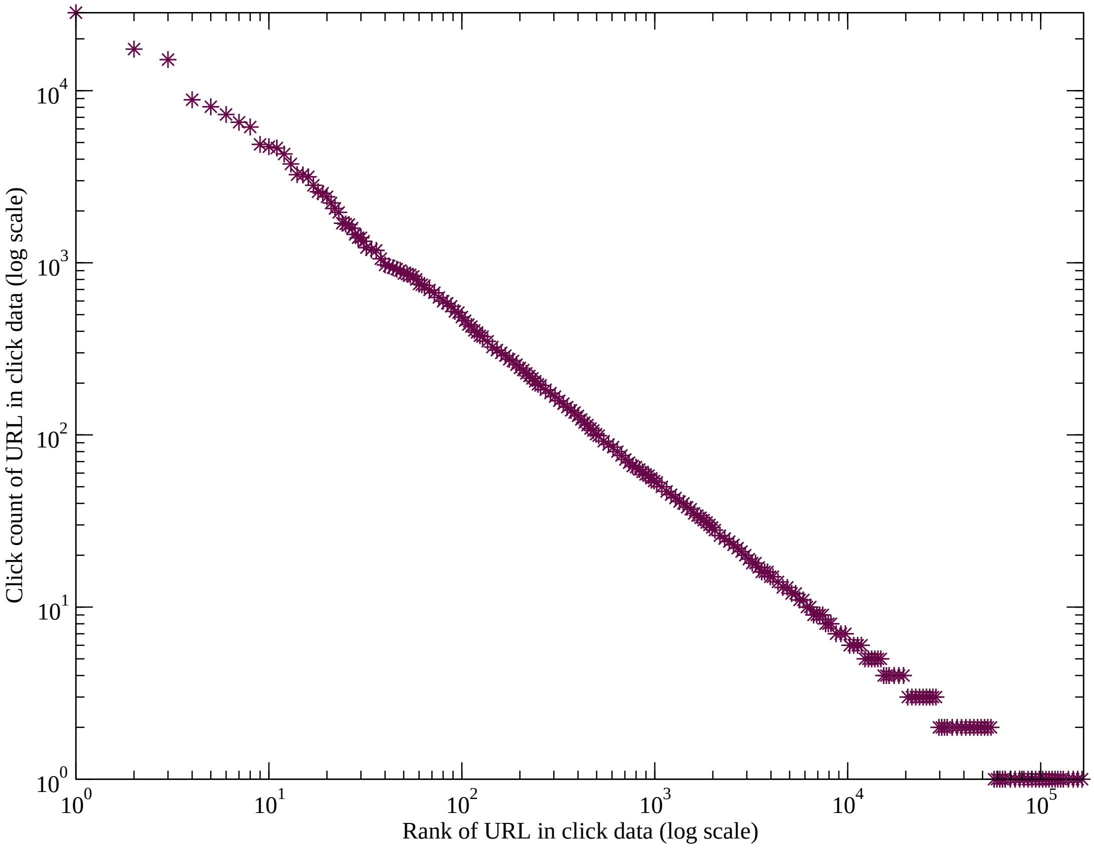


Figure 7.2: Distribution of URLs' clicks counts in web search results.

ground-truth ranking obtained after aggregating the clicks given in Table 7.4.

Next, we give some insights about the click data, browsing data and their correlation. Fig. 7.2 displays the distribution of click counts in our sample. As expected, the click counts follow a power-law distribution. Namely, there are few highly clicked URLs and many URLs with very few clicks. The scatter plot in Fig. 7.3 shows the correlation between the visit counts of URLs in the browsing data and their click counts. According to this figure, there is a partial correlation between the browsing data and click data. We observe a large number of URLs that are highly visited by web users in browsing data, but not received many clicks from search engine users when displayed in web search results. However, the opposite is not true, i.e., highly clicked web pages tend to be visited by many web users. This observation provides us enough motivation to use the click data as a ground-truth for representing page importance.

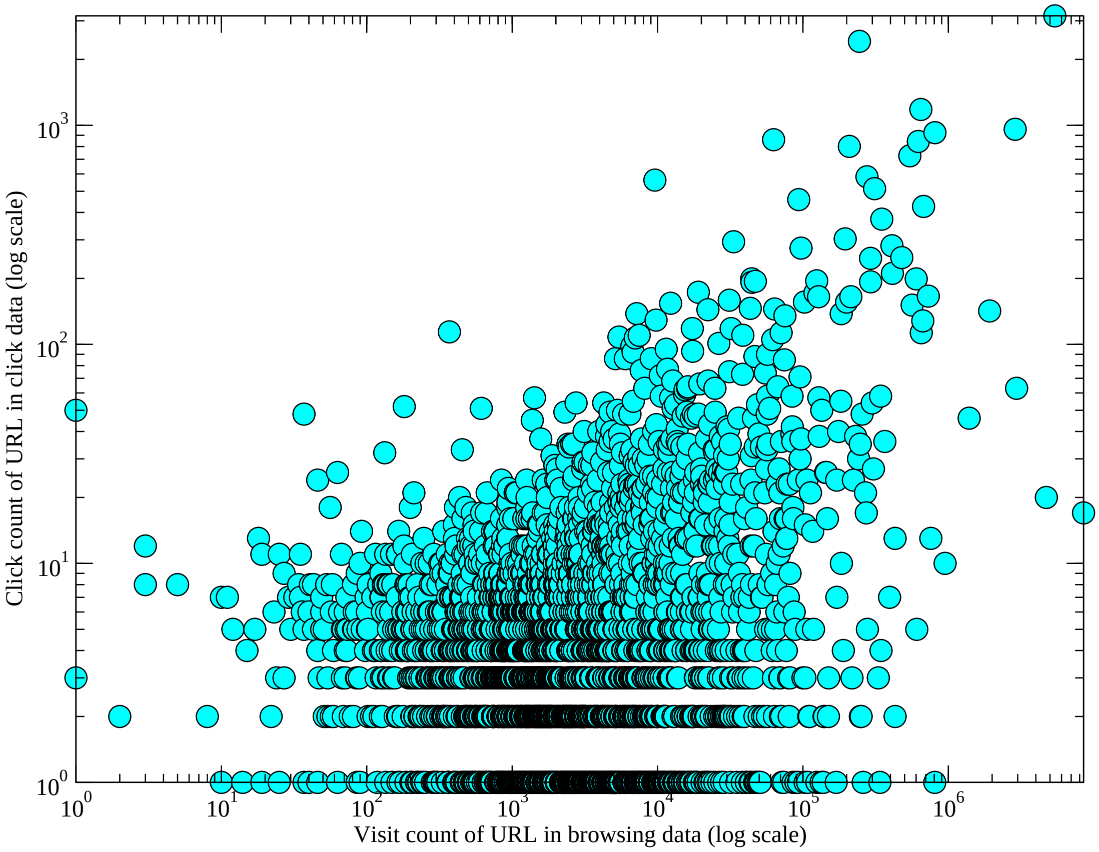


Figure 7.3: Visit count of a URL in the browsing data versus its click count in search results.

Chapter 8

Experiments

In this chapter, we give various statistics about our datasets, so that the reader can gain a deep understanding of the insights of data we use. Moreover, we present the results of various experiments conducted using our data.

8.1 Optimizing β

The β parameter, defined in Eq. 5.4, indicates the probability of the random surfer to follow a hyperlink while surfing on the web. In order to devise a realistic random surfer and make PBRank computations based on it, it is important to accurately adjust the β constant. Here, we can directly use the Eq. 5.4 which adjusts the β constant by measuring the ratio between the numbers of visits initiated by following an out-link in a page and the total number of visits in the browsing data. Using the Eq. 5.4 and the numbers given in Table 7.2, we have

$$\beta = \frac{1,189,735,491}{1,919,657,987} = 0.619764301 \approx 0.62, \tag{8.1}$$

In the rest of this thesis, we set $\beta = 0.62$ for all experiments. This value indicates that pages are slightly more likely to be visited by clicking on the hyperlinks.

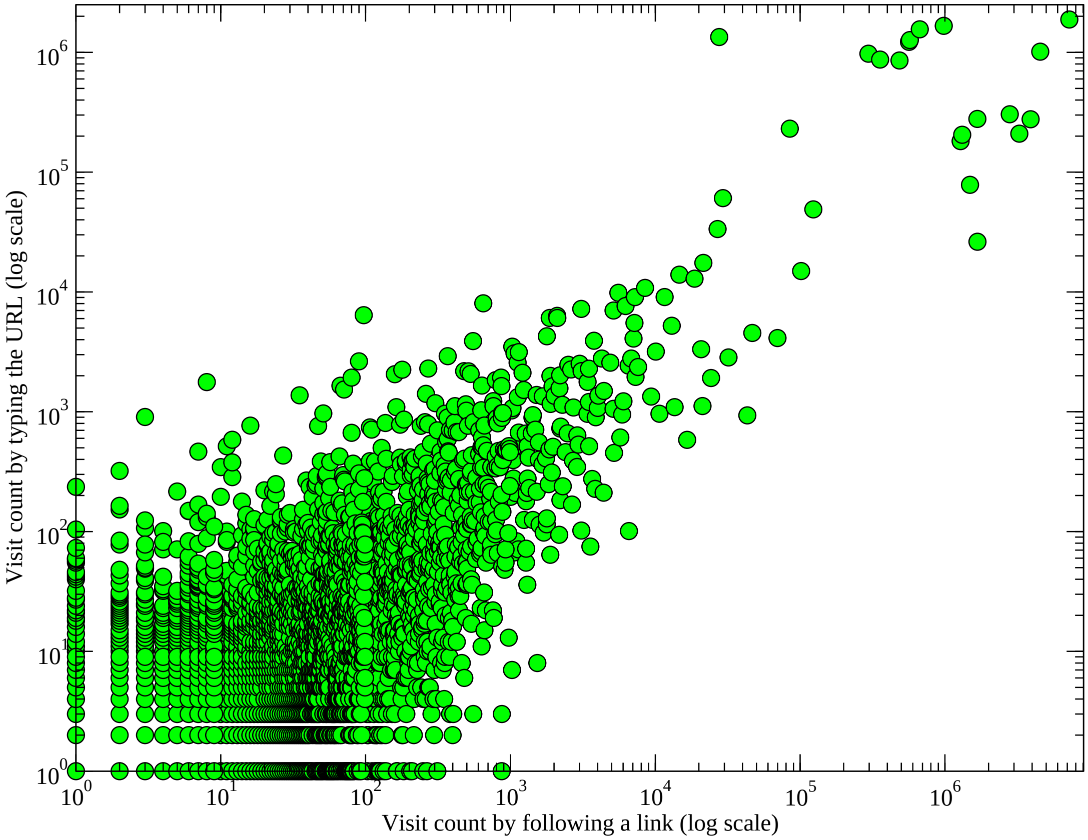


Figure 8.1: Number of times a URL is visited by following a link versus typing in the navigation bar.

The obtained number is consistent with the earlier observation in [4], where the β for surfing the entire web is estimated to be between 0.6 and 0.725.

According to Fig. 8.1, we observe in our browsing data that there is a positive correlation between visiting a page by following a link and visiting a page by typing its URL. That is, a page which is frequently visited by following links, tends to be frequently visited by typing its URL. Similarly, if a page is rarely visited by following links, then it is also rarely visited by typing its URL.

Finally, Fig. 8.2 displays the distribution of URL visit counts in the browsing data. We observe a power-law distribution. Moreover, counts of the visits by following links is slightly higher than the counts of the visits by typing URLs.

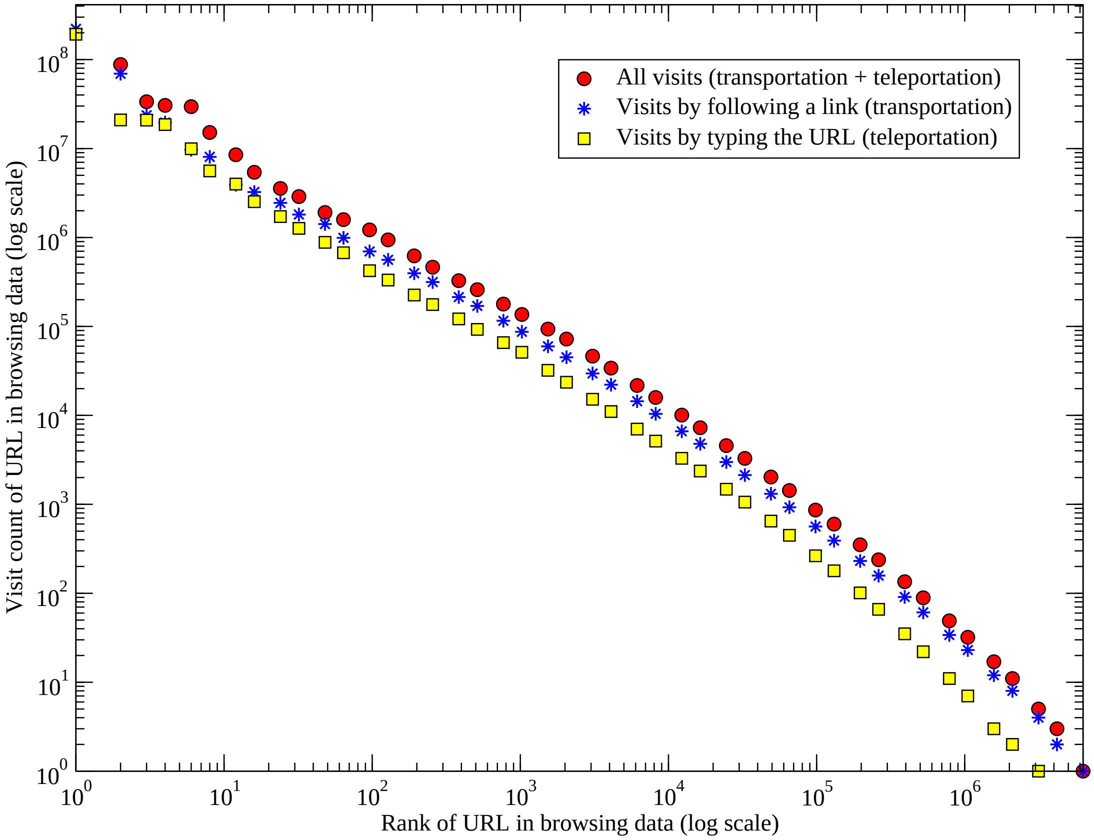


Figure 8.2: Distribution of URL visit counts in toolbar data.

8.2 In-link versus visit counts

A significant part of the page importance assigned in web data is affected by the hyperlinks referring to that page (i.e., in-links). Similarly, a significant part of the page importance assigned in browsing data is affected by the visit count of that page. In order to verify whether the web data and browsing data can be two complementary data sources in assessing the importance of a page, the scatter plot in Fig. 8.3 presents the visit counts and in-link counts of web pages. Looking at the figure, there is no clearly visible correlation between two data sources. This means that web data and browsing data can be two complementary data sources.

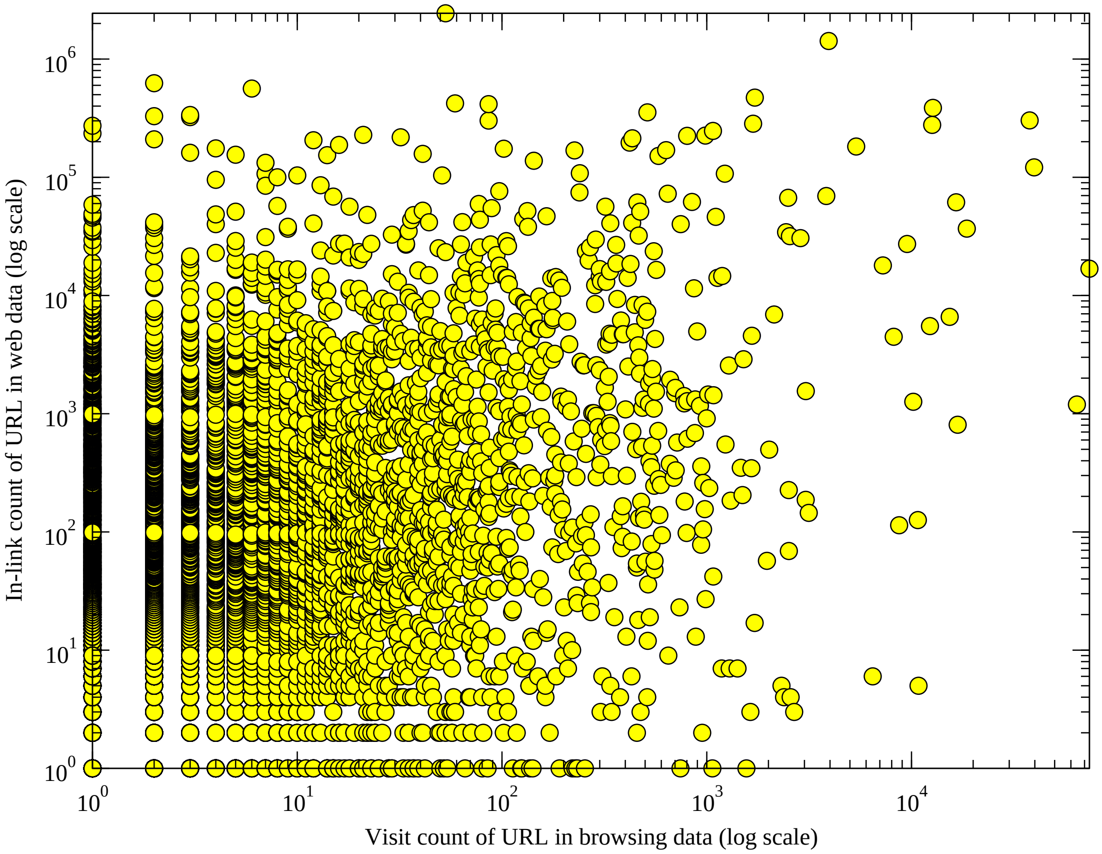


Figure 8.3: Number of times a URL is visited by a user versus it is linked by another URL.

8.3 Overlap among data sources

Fig. 8.4 displays the overlap between the three different data sources. While Fig. 8.4a) shows the overlap using URL counts, Fig. 8.4b) gives the percentages. As expected, the web data is larger than the browsing and click data. Among all available URLs, only 0.706% is not present in the web data. According to Fig. 8.5a) and Fig. 8.5b), 20.63% of the URLs (i.e., 1,628,058 URLs) in the browsing data are not present in the other two data sources. This shows that new URLs can be discovered through the browsing data [29]. However, the share of these URLs in the entire set of URLs is only 0.703% due to the large size of the web data (see Fig. 8.4b)). According to Fig. 8.5c) and Fig. 8.5d), a large portion of the ground-truth click data (96.68%) is available in either the web data or the browsing data. More than three-fourth of the URLs (i.e, 78.72%) in the click

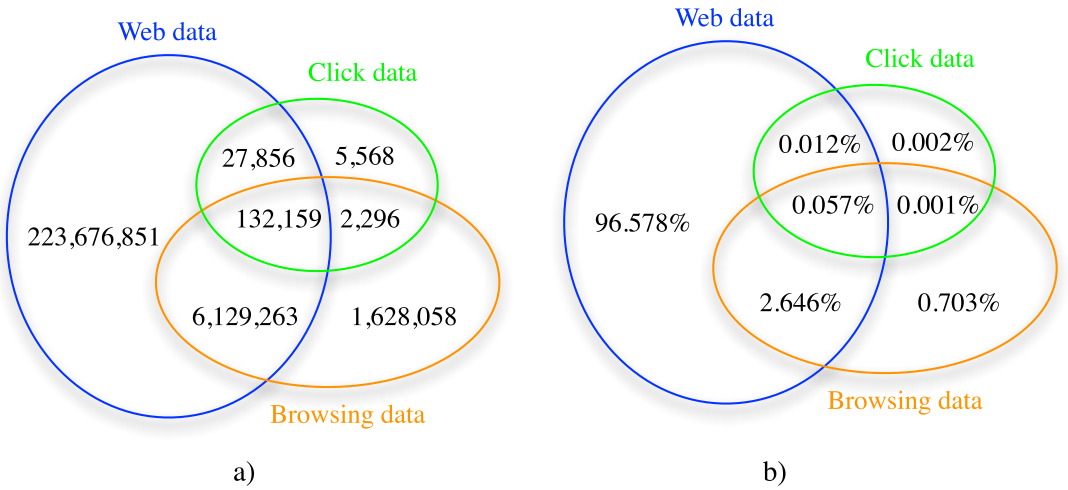


Figure 8.4: Distribution of all available URLs in the web data, browsing data, and click data. a) using unique URL count, b) using percentages.

Table 8.1: Coverage of URLs in the ground-truth click data when different data sources are used in ranking.

Data used in ranking	Coverage (χ)
Only browsing data	80.1%
Only web data	95.3%
Both web and browsing data	96.7%

data are available in both web data and browsing data. Finally, Fig. 8.5e) and Fig. 8.5f), presents the distribution of URLs that are available in the web data. As mentioned above, due to the large size of the web data, only a small portion of the web data is available in other two data sources.

8.4 Coverage

Based on the numbers in Fig. 8.5c) and Fig. 8.5d), we can compute the coverage metric, χ (see Eq. 6.1), as follows. Assume that there are three rankers. One of them uses only browsing data, the other one uses only web data, and the third one exploits both data sources. Table 8.1 shows the coverage values of those rankers. Using both web and browsing data at the same time provides a coverage increase of 16.6% over using only the browsing data. Although it is relatively minor,

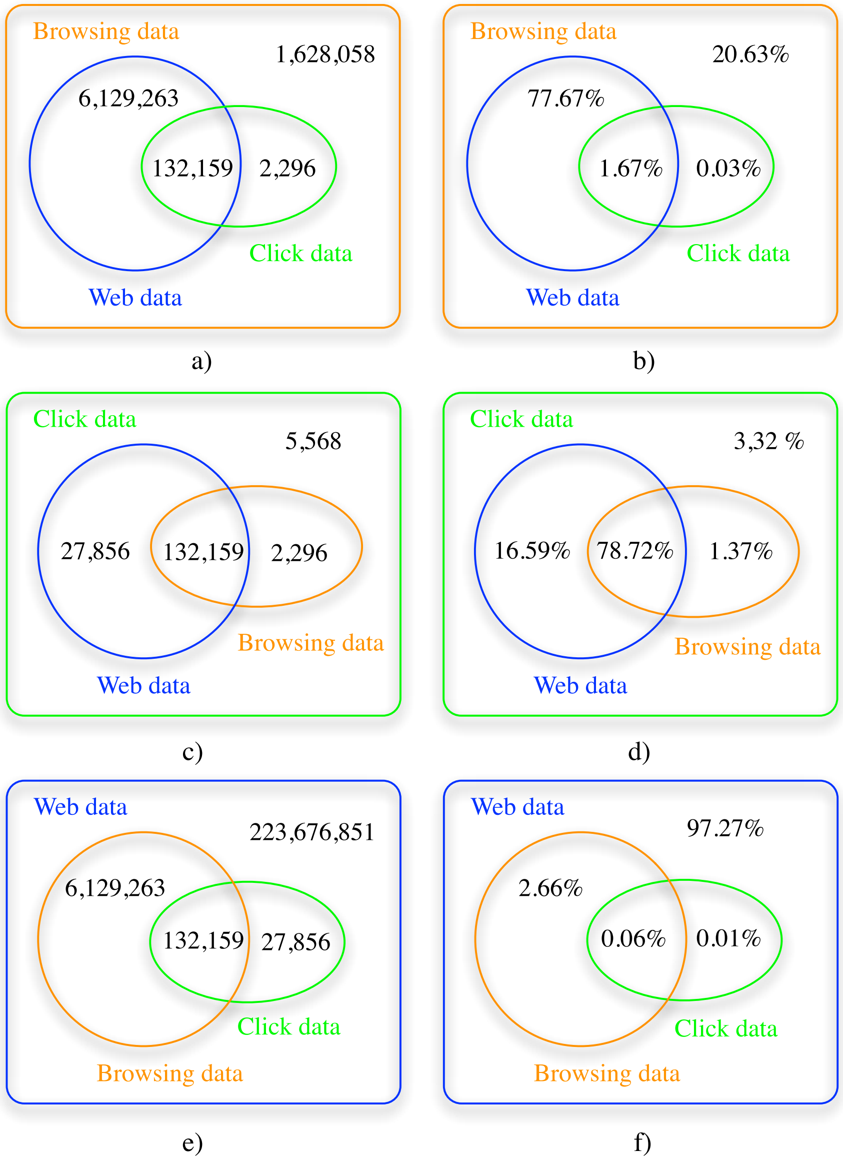


Figure 8.5: Distribution of URLs in different types of data: a-b) distribution of URLs that are available in the browsing data, c-d) distribution of URLs that are available in the click data, and e-f) distribution of URLs that are available in the web data.

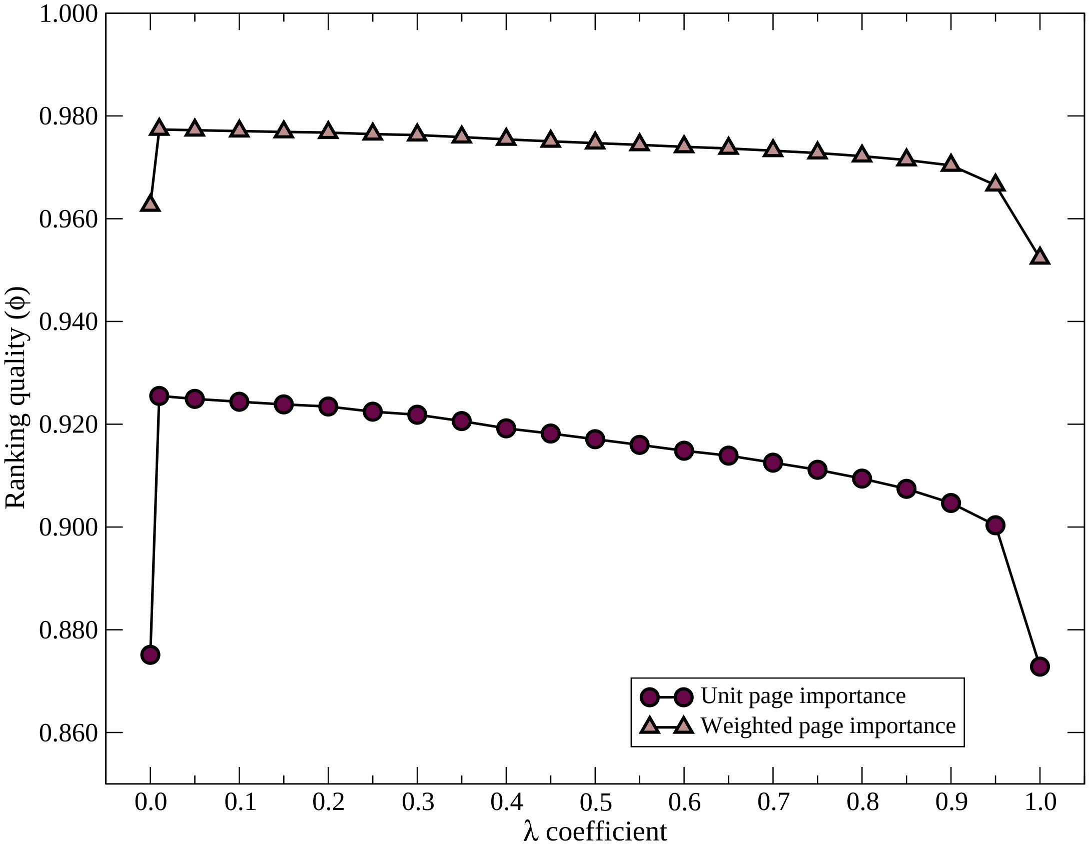


Figure 8.6: The variation in ranking quality (Φ) for different values of λ.

using both data sources improves the coverage metric by 1.4% over using only the web data. In either case, this result indicates that PBRank can produce non-zero importance scores for a larger number of URLs than both BrowseRank and PageRank. Therefore, the coverage values calculated above support our initial motivation of improving the coverage qualite of ranking by combining two data sources.

8.5 Optimizing λ

The λ parameter, defined in Eq. 5.1, indicates the weight of the web data in the hybrid ranking. We aim to find the λ value that optimizes the ranking quality metric defined (Φ) in Eq. 6.4. This means that for some λ value we expect an

Table 8.2: The ranking quality metric (Φ) for varying values of λ.

	Φ	
λ	Unit weight	Weighted
(only browsing data) 0	0.87512	0.96259
0.00001	0.92265	0.97637
0.0001	0.92390	0.97679
0.001	0.92496	0.97716
0.005	0.92536	0.97731
0.01	**0.92550**	**0.97738**
0.015	0.92541	0.97735
0.02	0.92531	0.97733
0.03	0.92525	0.97730
0.04	0.92504	0.97725
0.05	0.92494	0.97723
0.1	0.92437	0.97705
0.15	0.92387	0.97689
0.2	0.92345	0.97676
0.25	0.92244	0.97646
0.3	0.92186	0.97628
0.35	0.92062	0.97588
0.4	0.91919	0.97544
0.45	0.91818	0.97506
0.5	0.91709	0.97471
0.55	0.91599	0.97437
0.6	0.91484	0.97399
0.65	0.91390	0.97368
0.7	0.91252	0.97323
0.75	0.91114	0.97279
0.8	0.90943	0.97218
0.85	0.90743	0.97143
0.9	0.90467	0.97038
0.95	0.90034	0.96653
(only web data) 1	0.87283	0.95232

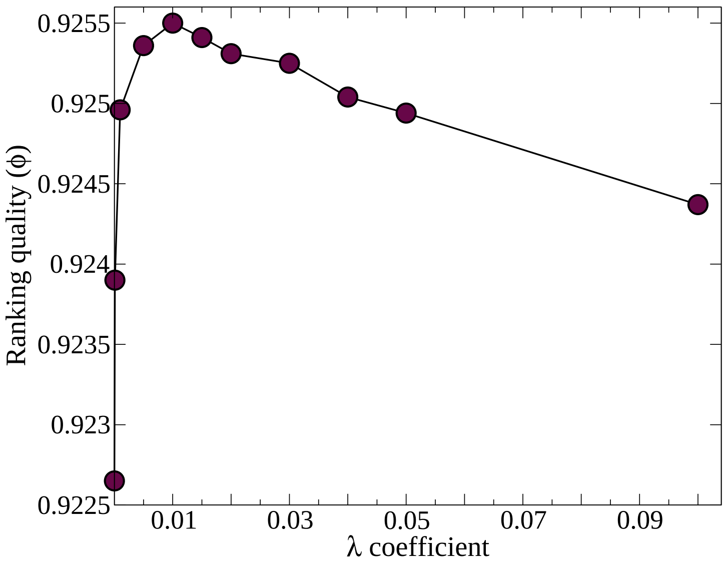


Figure 8.7: The variation in ranking quality (Φ) for the values of λ near 0, using unit page importance.

optimal ranking. To this end, we compute the value of the ranking quality metric for different PBRank rankings that are obtained by varying λ through parameter sweeping. Fig. 8.6 shows the values of the metric with λ increased between zero and one at increments of 0.1. As mentioned before, $\lambda = 0$ corresponds to our BrowseRank variant, which uses only the browsing data, and $\lambda = 1$ corresponds to PageRank, which uses only the web data. According to the figure, any λ value between zero and one yields a superior ranking performance than either baseline. We observe better performance as λ is closer to zero. Hence, we perform another parameter sweep for λ values near zero. Fig. 8.7 shows the values of the metric for λ near zero and when unit page importance is used. Similarly, Fig. 8.8 shows the values of the metric for λ near zero and when weighted page importance is used. All results of this experiment are displayed in Table 8.2. We observe that the optimum λ value is somewhere between 0.005 and 0.015. This indicates that the the browsing data should have a much higher influence than the web data in assessing URL importance. Specifically, we set $\lambda = 0.01$ in the rest of the experiments, because for $\lambda = 0.01$ we observe the best ranking quality. According to the ratio 0.99/0.01, the feedback obtained from the browsing data has 99 times more influence on the ranking quality than the feedback coming from the web data. To sum up, the best hybrid ranking is observed when 99% comes from

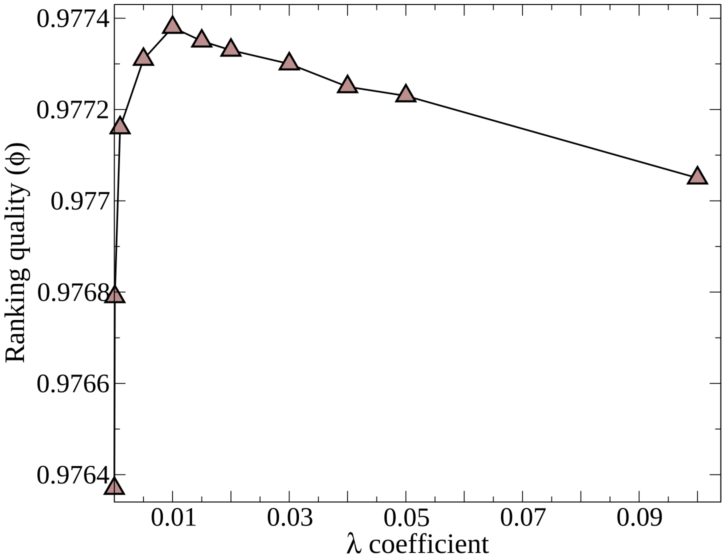


Figure 8.8: The variation in ranking quality (Φ) for the values of λ near 0, using weighted page importance.

browsing data, and the rest 1% comes from the web data.

8.6 Comparison of rankings

Next, in Fig. 8.9, we compare the distributions of URL importance scores generated by PBRank for three different values of λ: $\lambda \in \{0, 0.01, 1\}$. Since, the ranking generated by PBRank for $\lambda = 0.01$ is highly affected by the browsing data, we expect to obtain extremely similar rankings for $\lambda = 0$ and $\lambda = 0.01$. As expected, the score distributions for $\lambda = 0$ and $\lambda = 0.01$ are very similar to each other and different than the score distribution in case of $\lambda = 1$. Another observation is that the distribution for $\lambda = 0$ is shorter than the other two because fewer URLs (only those in the web browsing data) are ranked.

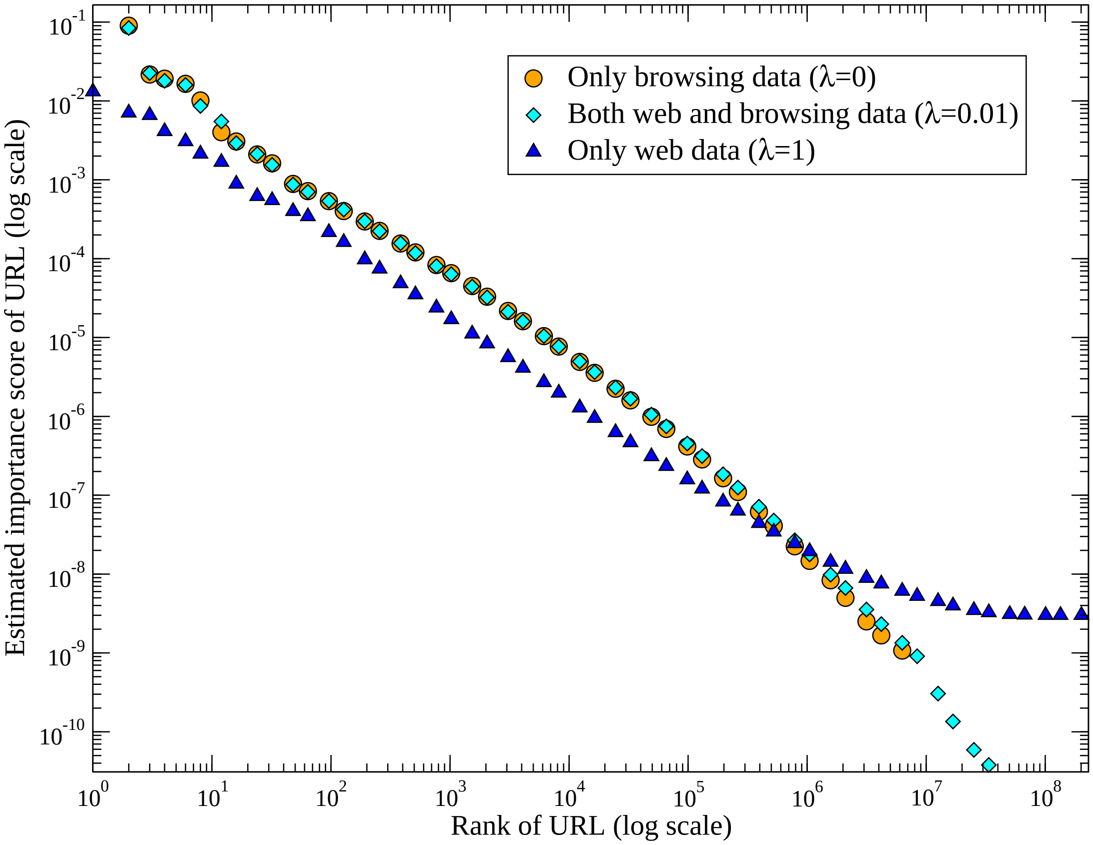


Figure 8.9: Distribution of URL importance scores.

8.7 Contribution of data sources to top k ranks

In this experiment, we analyze how data sources contribute to top k ranks of the ranking generated by PBRank for $\lambda = 0.01$. Table 8.3 reports the contribution of each data source to top k ranks. Here, the top $k = 100$ URLs of the ranking come from both web and browsing data. We observe URLs that are available only in the browsing data as k increases to 1,000 (i.e., 27 URLs out of top 1,000 URLs are available only in the browsing data). The URLs that are available only in the web data become visible after the top 1,000 ranks. This result indicates that the URLs in the very top ranks are mainly determined by the feedback obtained from the browsing data. On the other hand, looking at the entire ranking (i.e., $k = 10^8$), we observe a large contribution (i.e., around 92%) from URLs that are available only in web data. This means that the tail of the ranking comes from

Table 8.3: Contribution of different data sources to the top k URLs in PBRank with $\lambda = 0.01$.

k	Only web data	Both web and browsing data	Only browsing data
1	0	1	0
10	0	10	0
100	0	100	0
1,000	0	973	27
10,000	15	9,297	688
100,000	1,646	88,415	9,939
1,000,000	77,866	804,424	117,710
10,000,000	2,575,717	5,969,132	1,455,151
100,000,000	92,108,224	6,261,422	1,630,354

the web data, which supports our initial motivation of increasing the coverage of rankings. The numbers in Table 8.3 are visualized in Fig. 8.10 as contribution percentages.

8.8 Spatio-temporal user context

Next, we investigate if PBRank can be customized for the spatio-temporal context of users. The initial motivation for spatio-temporal customization is to check whether PBRank can generate specific rankings when the browsing data is customized. To clarify, we look for the answers of the following type of questions:

- If the browsing data is obtained only from US users, will PBRank generate a US-specific importance ranking?
- If the browsing data contains browsing logs with timestamps that correspond to morning hours (or night hours), will PBRank boost up the importances of the news web sites (or the web sites with adult content). Here, we assume that the news web sites are frequently visited in the morning, and the adult web sites are frequently visited during the night.
- If the browsing data is obtained only from users with Linux OS, will PBRank boost up the importances of the web sites related to the open source software

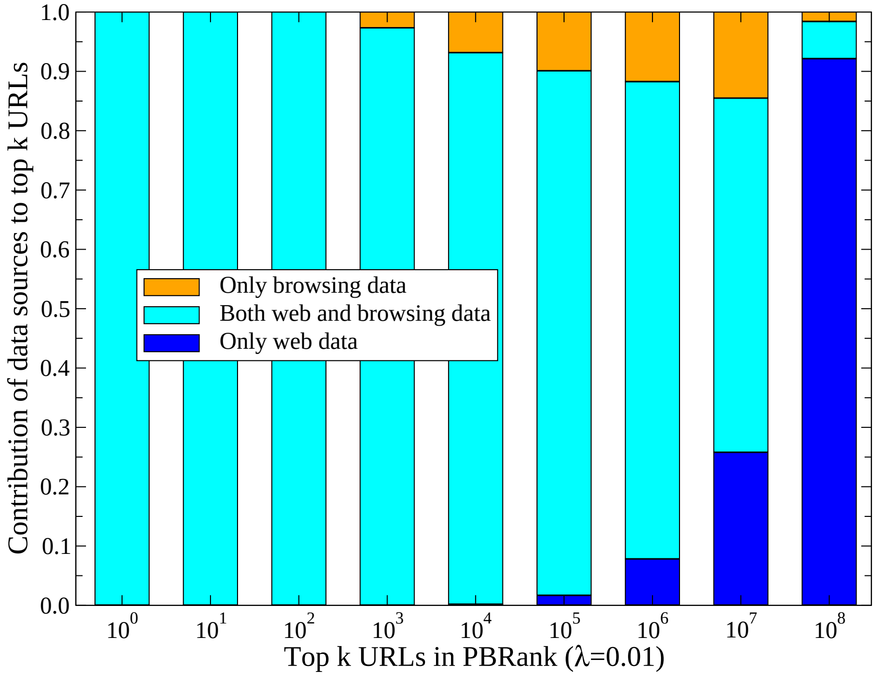


Figure 8.10: Contribution of different data sources to the top k URLs in PBRank with $\lambda = 0.01$.

community.

- If the browsing data contains only browsing logs from weekends, will PBRank generate a weekend-specific importance ranking?

In order to answer those questions, we conducted experiments by incorporating the meta-data available in browsing logs.

First, we check if PBRank can generate country-specific importance rankings. To this end, we build two separate PBRank models ($\lambda = 0.01$) using the browsing data obtained from the users located in the US or those in the UK. We refer to the URL rankings generated by these two models as PBRank-US and PBRank-UK, respectively. We then compute the quality of these two rankings against two different ground-truth click data: clicks issued by the users located in the US or the UK. These two sets of ground-truth data are referred to as G-US and G-UK, respectively. Table 8.4 shows the ranking quality for different combinations. With respect to G-US, better ranking quality is achieved when PBRank-US is used. The same holds for G-UK. This indicates the potential for location-specific

Table 8.4: Ranking quality when the ranking model uses browsing data belonging to users in different countries (United States and United Kingdom).

	Ground-truth click data	
Custom PBRank	G-US	G-UK
PBRank-US	**0.97833**	0.97478
PBRank-UK	0.96270	**0.97863**

URL rankings.

Next, we experiment with the time aspect and customized PBRank using browsing data obtained from particular hours of the day or days of the week. However, experiment results did not report much improvement in ranking qualities of hour-specific or day-specific rankings. Thus, we exclude those results. One reason for this can be the small size of browsing logs when customized for the spatio-temporal context. Although the size of the entire browsing data is large, its size decreases seriously when customized for the spatio-temporal context. Another reason can be the natural absence of hour-specific or day-specific importance rankings, i.e., the importances of adult web sites do not depend on the hours of the day.

8.9 Top 40 hosts

Tables 8.5, 8.6, and 8.7 give the top 40 hosts ranked by PBRank with $\lambda = 0$, $\lambda = 0.01$ and $\lambda = 1$, respectively. In general, for $\lambda = 1$, the `godaddy.com` subdomains dominate the top rankings (these domains are known to be supported by link farms). For λ values close to zero, all `godaddy.com` subdomains are pushed down to lower ranks. This is because the feedback from the browsing data makes it clear that these hosts are not important enough to appear at the top ranks. Another interesting point is the high rank of `bobparsons.me`, which is Bob Parsons's (the CEO of godaddy.com) personal blog. We believe that many pages in godaddy.com's link farms give links to `bobparsons.me`, artificially increasing its importance. This example clearly demonstrates one of the drawbacks of using only web data for page importance estimations. On the other side, $\lambda = 0$

Table 8.5: The top 40 web hosts ranked by using only browsing data ($\lambda=0$).

Rank	$\lambda=0$ (only browsing data)
1	http://facebook.com
2	http://apps.facebook.com
3	http://google.com
4	http://mail.google.com
5	http://youtube.com
6	http://accounts.google.com
7	http://yahoo.com
8	http://accounts.youtube.com
9	http://search.yahoo.com
10	http://login.yahoo.com
11	http://login.live.com
12	http://mail.yahoo.com
13	http://google.com.vn
14	http://twitter.com
15	http://google.co.in
16	http://tagged.com
17	http://us.lrd.yahoo.com
18	http://online.wellsfargo.com
19	http://google.ro
20	http://translate.google.com
21	http://bing.com
22	http://bankofamerica.com
23	http://us.mg5.mail.yahoo.com
24	http://chaseonline.chase.com
25	http://get.adobe.com
26	http://docs.google.com
27	http://tw.yahoo.com
28	http://paypal.com
29	http://us.mg4.mail.yahoo.com
30	http://google.fr
31	http://msn.com
32	http://tw.rd.yahoo.com
33	http://plus.google.com
34	http://google.co.id
35	http://adobe.com
36	http://edit.yahoo.com
37	http://google.com.eg
38	http://sitekey.bankofamerica.com
39	http://amazon.com
40	http://ebay.com

Table 8.6: The top 40 web hosts ranked by PBRank with $\lambda = 0.01$.

Rank	$\lambda = 0.01$
1	`http://facebook.com`
2	`http://apps.facebook.com`
3	`http://google.com`
4	`http://youtube.com`
5	`http://mail.google.com`
6	`http://accounts.google.com`
7	`http://yahoo.com`
8	`http://accounts.youtube.com`
9	`http://search.yahoo.com`
10	`http://login.yahoo.com`
11	`http://twitter.com`
12	`http://login.live.com`
13	`http://mail.yahoo.com`
14	`http://google.com.vn`
15	`http://tagged.com`
16	`http://google.co.in`
17	`http://us.lrd.yahoo.com`
18	`http://adobe.com`
19	`http://google.ro`
20	`http://translate.google.com`
21	`http://online.wellsfargo.com`
22	`http://get.adobe.com`
23	`http://bing.com`
24	`http://bankofamerica.com`
25	`http://us.mg5.mail.yahoo.com`
26	`http://tw.yahoo.com`
27	`http://docs.google.com`
28	`http://chaseonline.chase.com`
29	`http://us.mg4.mail.yahoo.com`
30	`http://google.fr`
31	`http://tw.rd.yahoo.com`
32	`http://msn.com`
33	`http://paypal.com`
34	`http://edit.yahoo.com`
35	`http://google.co.id`
36	`http://amazon.com`
37	`http://google.com.eg`
38	`http://blogger.com`
39	`http://plus.google.com`
40	`http://sitekey.bankofamerica.com`

Table 8.7: The top 40 web hosts ranked by using only web data (λ=1).

Rank	λ=1 (only web data)
1	`http://godaddy.com`
2	`http://twitter.com`
3	`http://facebook.com`
4	`http://blogger.com`
5	`http://google.com`
6	`http://mya.godaddy.com`
7	`http://adobe.com`
8	`http://community.godaddy.com`
9	`http://wordpress.org`
10	`http://youtube.com`
11	`http://videos.godaddy.com`
12	`http://auctions.godaddy.com`
13	`http://addthis.com`
14	`http://maps.google.com`
15	`http://amazon.com`
16	`http://accounts.google.com`
17	`http://idp.godaddy.com`
18	`http://linkedin.com`
19	`http://validator.w3.org`
20	`http://statcounter.com`
21	`http://apple.com`
22	`http://networksolutions.com`
23	`http://macromedia.com`
24	`http://wordpress.com`
25	`http://ad.doubleclick.net`
26	`http://flickr.com`
27	`http://whoisprivacyprotect.com`
28	`http://securepaynet.net`
29	`http://myspace.com`
30	`http://buzz.blogger.com`
31	`http://acquirethisname.com`
32	`http://bobparsons.me`
33	`http://jigsaw.w3.org`
34	`http://w3.org`
35	`http://t.co`
36	`http://parallels.com`
37	`http://dcc.godaddy.com`
38	`http://namedrive.com`
39	`http://blog.twitter.com`
40	`http://quantcast.com`

Table 8.8: The change in the rankings of selected web hosts that are important according to the browsing data, but not the web data.

	Rank		
Web hosts	$\lambda=0$	$\lambda=0.01$	$\lambda=1$
`http://apps.facebook.com`	2	2	63
`http://mail.google.com`	4	5	109
`http://login.live.com`	11	12	678
`http://online.wellsfargo.com`	18	21	31,885
`http://search.yahoo.com`	9	9	281
`http://bankofamerica.com`	22	24	4,303
`http://chaseonline.chase.com`	24	28	58,463
`http://get.adobe.com`	25	22	49
`http://tagged.com`	16	15	25,079
`http://ecampus.phoenix.edu`	106	123	197,314

seems to boost the ranks of hosts that belong to banks due to the popularity of online banking. Due to similar reasons, commonly used web services (e.g., `apps.facebook.com` and `mail.google.com`) are also highly ranked by PBRank with λ close to zero.

8.10 Largest rank variations

Next, we analyze the large variation in the rankings of some selected web hosts. Table 8.8 shows ten manually selected web hosts that are important according to the browsing data, but not the web data. Here, we observe hosts that belong to popular banks (e.g., `bankofameria.com` and `chaseonline.chase.com`) and commonly used web services (e.g., `apps.facebook.com` and `mail.google.com`). The ranks of these hosts, which are not highly linked in the Web, are boosted when the influence of the browsing data is increased. Similarly, Table 8.9 shows ten web hosts that are important according to the web data, but not the browsing data. One observation is that the links from the social widgets present in a large number of web pages increase the ranks of social websites such as `twitter.com` and `digg.com`, when the rankings are influenced by the web data. It may be surprising that `en.wikipedia.org` appears at the 54th rank. We note that this host is the

Table 8.9: The change in the rankings of selected web hosts that are important according to the web data, but not the browsing data.

	Rank		
Web hosts	$\lambda=0$	$\lambda=0.01$	$\lambda=1$
`http://twitter.com`	14	11	2
`http://godaddy.com`	4,407	324	1
`http://adobe.com`	35	18	7
`http://blogger.com`	187	38	4
`http://linkedin.com`	180	98	18
`http://en.wikipedia.org`	55	54	46
`http://flickr.com`	203	141	26
`http://myspace.com`	290	142	29
`http://digg.com`	5,562	220	50
`http://wordpress.com`	4,201	999	24

English version of Wikipedia. In case of top level domains, `wikipedia.org` would be ranked much higher. It is interesting to observe that `myspace.com` loses ranks when the influence of the browsing data is increased. This is mainly due to the fading popularity of MySpace among users, despite the large number of MySpace links that are still present in the Web.

Chapter 9

Conclusion and Future Work

We proposed a novel model for computing web page importance scores by using a mixture of the feedback extracted from the hyperlink structure of the Web and the feedback obtained from the web browsing patterns of users. The first type of feedback serves as a remedy to the sparsity issue in the web browsing patterns while the latter helps improving the accuracy of computed importance scores. According to a quality metric using user clicks on web search results mined from a query log, the proposed hybrid model exploiting both the web structure and the navigation patterns of users lead to a better performance than using only a single type of feedback. We found that the optimum mixture is achieved when 99% of the score comes from the browsing feedback, and only 1% from the web feedback. Moreover, we demonstrated the spatial variation in user browsing behavior and exploited this variation to compute custom scores that depend on the current location of users.

As a future work, we consider to improve this work in the following four aspects. First issue is related with the page level experiments. In this work, all experiments are conducted at the host level. Although we expect the experiments on the page-level graphs to yield similar results (i.e., coverage and ranking quality), we left experiments at the page level as a future work. Second idea for further improvements of PBRank is to use different λ values for different hosts/pages. Every page has a different nature and the optimal λ value might differ depending on page's properties such as page content, page size, out-link amount. Another future work is to evaluate PBRank by comparing it with the real BrowseRank implementation. Last point is related with the proposed evaluation metric. The effectiveness of the our evaluation metric can be analyzed in more detail. Since evaluating the evaluation methods is a hard problem, we left this as a future work.

Bibliography

[1] M. Richardson, A. Prakash, and E. Brill, "Beyond PageRank: machine learning for static ranking," in *Proc. 15th Int'l Conf. World Wide Web*, pp. 707–715, 2006.

[2] B. B. Cambazoglu, H. Zaragoza, O. Chapelle, J. Chen, C. Liao, Z. Zheng, and J. Degenhardt, "Early exit optimizations for additive machine learned ranking systems," in *Proc. 3rd ACM Int'l Conf. Web Search and Data Mining*, pp. 411–420, 2010.

[3] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the Web," tech. rep., Stanford University, 1998.

[4] D. F. Gleich, P. G. Constantine, A. D. Flaxman, and A. Gunawardana, "Tracking the random surfer: empirically measured teleportation parameters in PageRank," in *Proc. 19th Int'l Conf. World Wide Web*, pp. 381–390, 2010.

[5] Z. Gyöngyi and H. Garcia-Molina, "Link spam alliances," in *Proc. 31st Int'l Conf. Very Large Data Bases*, pp. 517–528, 2005.

[6] Y. Liu, B. Gao, T.-Y. Liu, Y. Zhang, Z. Ma, S. He, and H. Li, "BrowseRank: letting web users vote for page importance," in *Proc. 31st Annual Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, pp. 451–458, 2008.

[7] Y. Ding, E. Yan, A. Frazho, and J. Caverlee, "PageRank for ranking authors in co-citation networks," *J. Am. Soc. Inf. Sci. Technol.*, vol. 60, no. 11, pp. 2229–2243, 2009.

[8] J. Cho, H. Garcia-Molina, and L. Page, "Efficient crawling through URL ordering," in *Proc. 7th Int'l Conference on World Wide Web*, pp. 161–172, 1998.

[9] Z. Gyongyi, P. Berkhin, H. Garcia-Molina, and J. Pedersen, "Link spam detection based on mass estimation," in *Proc. 32nd Int'l Conf. Very Large Data Bases*, pp. 439–450, 2006.

[10] R. Mihalcea, P. Tarau, and E. Figa, "PageRank on semantic networks, with application to word sense disambiguation," in *Proc. 20th Int'l Conf. Computational Linguistics*, 2004.

[11] J. Kleinberg, "Authoritive sources in a hyperlinked environment," *Journal of the ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[12] R. Lempel and S. Moran, "SALSA: The stochastic approach for link-structre analysis," *ACM Transactions on Information Systems*, vol. 19, no. 2, pp. 131–160, 2001.

[13] R. Baeza-Yates, P. Boldi, and C. Castillo, "Generalizing PageRank: damping functions for link-based ranking algorithms," in *Proc. 29th Annual Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, pp. 308–315, 2006.

[14] S. Kamvar, T. Haveliwala, and G. Golub, "Adaptive methods for computation of PageRank," in *Proc. Int'l Conf. Numerical Solution of Markov Chains*, 2003.

[15] S. Kamvar, T. Haveliwala, C. Manning, and G. Golub, "Exploiting the block structure of the Web for computing PageRank," tech. rep., Stanford University, 2003.

[16] A. N. Langville and C. D. Meyer, "Updating Markov chains with an eye on Google's PageRank," *SIAM J. Matrix Anal. Appl.*, vol. 27, no. 4, pp. 968–987, 2005.

[17] F. McSherry, "A uniform approach to accelerated PageRank computation," in *Proc. 14th Int'l Conf. World Wide Web*, pp. 575–582, 2005.

[18] A. Cevahir, C. Aykanat, A. Turk, and B. B. Cambazoglu, "Site-based partitioning and repartitioning techniques for parallel PageRank computation," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 5, pp. 786–802, 2011.

[19] D. Gleich, L. Zhukov, and P. Berkhin, "Fast parallel PageRank: A linear system approach," Tech. Rep. YRL-2004-038, Yahoo!, 2004.

[20] C. Kohlschütter, P.-A. Chirita, and W. Nejdl, "Efficient parallel computation of PageRank," in *Advances in Information Retrieval* (M. Lalmas, A. MacFarlane, S. Rüger, A. Tombros, T. Tsikrika, and A. Yavlinsky, eds.), vol. 3936 of *Lecture Notes in Computer Science*, pp. 241–252, Springer Berlin Heidelberg, 2006.

[21] P. Berkhin, "A survey on PageRank computing," *Internet Mathematics*, vol. 2, pp. 73–120, 7 2005.

[22] A. Langville and C. Meyer, "Deeper inside PageRank," *Internet Mathematics*, vol. 1, no. 3, pp. 335–380, 2005.

[23] K. Avrachenkov, N. Litvak, and K. S. Pham, "Distribution of PageRank mass among principle components of the Web," in *Proc. 5th Int'l Conf. Algorithms and Models for the Web-Graph*, pp. 16–28, 2007.

[24] P. Boldi, M. Santini, and S. Vigna, "PageRank as a function of the damping factor," in *Proc. 14th Int'l Conf. World Wide Web*, pp. 557–566, 2005.

[25] L. Pretto, "A theoretical analysis of Google's PageRank," in *Proc. 9th Int'l Symp. String Processing and Information Retrieval*, pp. 131–144, 2002.

[26] T. H. Haveliwala, "Topic-sensitive PageRank: A context-sensitive ranking algorithm for web search," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 4, pp. 784–796, 2003.

[27] G. Jeh and J. Widom, "Scaling personalized web search," in *Proc. 12th Int'l Conf. World Wide Web*, pp. 271–279, 2003.

[28] T. Haveliwala, "An analytical comparison of approaches to personalizing PageRank," tech. rep., Stanford University, 2003.

[29] X. Bai, B. B. Cambazoglu, and F. P. Junqueira, “Discovering URLs through user feedback,” in *Proc. 20th ACM Int’l Conf. Information and Knowledge Management*, pp. 77–86, 2011.

[30] E. Adar, J. Teevan, S. T. Dumais, and J. L. Elsas, “The web changes everything: understanding the dynamics of web content,” in *Proc. 2nd ACM Int’l Conf. Web Search and Data Mining*, pp. 282–291, 2009.

[31] E. Adar, J. Teevan, and S. T. Dumais, “Resonance on the Web: web dynamics and revisitation patterns,” in *Proc. 27th Int’l Conf. Human Factors in Computing Systems*, pp. 1381–1390, 2009.

[32] J. Huang and R. W. White, “Parallel browsing behavior on the Web,” in *Proc. 21st ACM Conf. Hypertext and Hypermedia*, pp. 13–18, 2010.

[33] R. Kumar and A. Tomkins, “A characterization of online browsing behavior,” in *Proc. 19th Int’l Conf. World Wide Web*, pp. 561–570, 2010.

[34] G. H. Golub and J. F. V. Loan, *Matrix Computation.* John Hopkins University Press, 3 ed., 1996.

[35] M. Bianchini, M. Gori, and F. Scarselli, “Inside PageRank,” *ACM Transactions on Internet Technology*, vol. 5, no. 1, 2005.

[36] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub, “Extrapolation methods for accelerating PageRank computations,” in *Proc. 12th Int’l Conf. World Wide Web*, pp. 261–270, 2003.

[37] R. Baeza-Yates and E. Davis, “Web page ranking using link attributes,” in *Proc. 13th Int’l World Wide Web Conf. (alternate track papers & posters)*, pp. 328–329, 2004.

[38] Y. Liu, T.-Y. Liu, B. Gao, Z. Ma, and H. Li, “A framework to compute page importance based on user behaviors,” *Information Retrieval*, vol. 13, no. 1, pp. 22–45, 2010.

[39] R. W. White, M. Bilenko, and S. Cucerzan, “Studying the use of popular destinations to enhance web search interaction,” in *Proceedings of the 30th*

annual international ACM SIGIR conference on Research and development in information retrieval, pp. 159–166, ACM, 2007.

[40] Z. Wang and X. Yang, *Birth and death processes and Markov chains.* Springer-Verlag Science Press, Beijing, 1992.

[41] W. J. Stewart, *Introduction to the numerical solution of Markov chains*, vol. 41. Princeton University Press Princeton, 1994.

[42] S. E. Robertson, "Overview of okapi projects," *Journal of Documentation*, vol. 53, no. 1, pp. 3–7, 1997.

[43] M. G. Kendal, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1-2, pp. 81–93, 1938.

[44] M. Melucci, "On rank correlation in information retrieval evaluation," *SIGIR Forum*, vol. 41, pp. 18–33, June 2007.

[45] C. Spearman, "The proof and measurement of association between two things," *The American Journal of Psychology*, vol. 15, no. 1, pp. 72–101, 1904.

[46] R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar, and E. Vee, "Comparing partial rankings," *SIAM Journal on Discrete Mathematics*, vol. 20, no. 3, pp. 628–648, 2006.

[47] R. Baeza-Yates, B. Ribeiro-Neto, *et al.*, *Modern information retrieval*, vol. 463. ACM press New York, 1999.

[48] K. Jarvelin and J. Kekalainen, "Cumulated gain-based evaluation of IR techniques," *ACM Transactions on Information Systems*, vol. 20, no. 4, pp. 422–446, 2002.

[49] W. R. Knight, "A computer method for calculating kendall's tau with ungrouped data," *Journal of the American Statistical Association*, vol. 61, no. 314, pp. 436–439, 1966.

[50] T. Fawcett, "An introduction to ROC analysis," *Pattern Recogn. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.

[51] C. Olston, B. Reed, U. Srivastava, R. Kumar, and A. Tomkins, “Pig Latin: a not-so-foreign language for data processing,” in *Proc. 2008 ACM SIGMOD Int’l Conf. Management of Data*, pp. 1099–1110, 2008.