Personalizing PageRank
Based on Domain Profiles

Mehmet S. Aktas, Mehmet A. Nacar, and Filippo Menczer
Computer Science Department
Indiana University
Bloomington, IN 47405 USA
{maktas,mnacar,fil}@indiana.edu

Abstract

Personalized versions of PageRank have been proposed to rank the results of a search engine based on a user's topic or query of interest. This paper introduces a methodology for personalizing PageRank vectors based on URL features such as Internet domains. Users specify interest profiles as binary feature vectors where a feature corresponds to a DNS tree node. Given a profile vector, a weighted PageRank can be computed assigning a weight to each URL based on the match between the URL and the profile features. We present promising preliminary results from a small experiment in which users were allowed to select among nine URL features combining the top two levels of the DNS tree, leading to 2^9 pre-computed PageRank vectors from a Yahoo crawl. Personalized PageRank performed favorably compared to pure similarity based ranking and traditional PageRank.

Key Words: personalized search, link analysis, PageRank, internet domain profiles, web search, personalized PageRank vectors

1. Introduction

The Web is a highly distributed and heterogeneous information environment. The immense number of Web documents presents various challenges for search engines. Storage space, crawling speed, and computational speed are some of these challenges. This paper deals with the retrieval of the most relevant documents. Recent search engines rank pages by combining traditional information retrieval techniques based on page content, such as the word vector space [1, 2], with link analysis techniques based on the hypertext structure of the Web, such as PageRank [3] and HITS [4].

The PageRank algorithm provides a global ranking of Web pages based on their importance estimated from hyperlinks [5, 3, 6]. For instance, a link from page "A" to page "B" is considered as if page "A" is voting for the importance of page "B". So, as the number of links to page "B" increases, its importance increases as well. In PageRank, not only the number of inlinks but their sources decide the importance of a page. In this scenario the global ranking of pages is based on the Web graph structure. Search engines such as Google[1] utilize the link structure of the Web to calculate the PageRank values of the pages. These values are then used to rank search results to improve precision. Comprehensive reviews of the issues related to PageRank can be found in [7, 8, 9].

The PageRank algorithm [5, 3] attempts to provide an objective global estimate of Web page importance. However, the importance of Web pages is subjective for different users and thus can be better determined if the PageRank algorithm takes into consideration user preferences. The importance of a page depends of the different interests and knowledge of different people; a global ranking of a Web page might not necessarily capture the importance of that page for a given individual user. Here we explore how to personalize PageRank based on features readily available from page URLs. For instance a user might favor pages from a specific geographic region, as may be revealed by Internet (DNS) domains. Likewise, topical features of Internet domains might also reflect user preferences. A user might prefer pages that are more likely to be monitored

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

by experts for accuracy and quality, such as pages published by academic institutions. Current search engines cannot rank pages based on individual user needs and preferences.[2]

In order to address the above limitations of global PageRank, we introduce a methodology to personalize PageRank scores based on URL features such as Internet domains. In this scenerio, users specify interest profiles as binary feature vectors where a feature corresponds to a DNS tree node or node set. We pre-compute PageRank scores for each profile vector by assigning a weight to each URL based on the match between the URL and the profile features. A weighted PageRank vector is then computed based on URL weights, and used at query time to rank results. We present promising preliminary results from a small experiment in which users were allowed to select among nine URL features combining the top two levels of the DNS tree, leading to 2^9 pre-computed PageRank vectors.

In the next section we discuss work relevant to PageRank computation and personalizing PageRank. Section 3. presents our method of computation for personalized PageRank vectors and outlines how user profiles are created based on Internet domains. Section 4. details the design and architecture of our implementation as well as a user study conducted to evaluate our methodology. Experimental results are presented in Section 5.

2. Background

The idea of a personalized PageRank was first introduced in [5] and has been studied by various researchers [10, 11, 12] as a query-dependent ranking mechanism. If personal preferences are based on n binary features, there are 2^n different personalized PageRank vectors for all possible user preferences. This requires an enourmous amount of computation and storage facilities. In an attempt to solve this problem, a method was introduced that computes only a limited amount of PageRank vectors offline [12]. This method provides for a methodology where personalized PageRank vectors can be computed at query time for all other possible user preferences. The main concern of the work presented here is to introduce a methodology for personalizing PageRank vectors based on URL features. To this end, we limit the choices of user preferences to topical and geographic features of Internet domains.

Techniques for efficient and scalable calculation of PageRank scores are an area of very active reseach [13, 14, 15, 16]. While this is important and relevant to the issue of personalized PageRank discussed here, it is outside the scope of the present paper. For the experiments presented here we use a collection from a relatively small crawl ($\sim 10^5$ pages, cf. Section 4.), and it is not necessary to recompute PageRank frequently. Therefore scalability is not discussed further.

Google has recently started beta-testing a personalized Web search service based on topical user profiles.[3] It appears that user profiles are based on hierarchical topic directories (a'la Open Directory Project[4]), however due to lack of documentation we are unable to discuss the similarities or differences between this work and the methodology proposed here.

"Topic-sensitive" web search, introduced by Haveliwala [10], is similar to our work. The method suggests pre-computation of topical PageRank vectors prior to query time. The idea is to minimize the jumping probability to pages that are considered as irrelevant to the topic. Topic-sensitive PageRank vectors are then combined at query time based on the similarity between topics and query. In our approach we personalize PageRank scores by assigning weights to URLs based on matched URL features. At query time the user's profile is matched with the corresponding personalized PageRank vector. As in the traditional PageRank, our method does not require the content of pages since we are only interested in URLs.

[2] See Section 2. for an exception to this, currently under beta-testing by Google.
[3] http://labs.google.com/personalized
[4] http://dmoz.org

3. Domain-Based Personalized PageRank

3.1 Personalized PageRank Vectors

Personalized PageRank vectors provide a ranking mechanism which in turn creates a personalized view of the Web for individual users. The computation of personalized PageRank vectors is done prior to search time. When calculating the PageRank vectors, predefined user profiles are taken into consideration.

We use the following recursive definition for personalized PageRank computation:

$$R_U(p) = (1-d) + d \cdot \sum_{\{q:q\rightarrow p\}} \frac{W_U(q) \cdot R_U(q)}{|s : q \rightarrow s|}$$

where U is the user profile, d is the traditional jump probability (or damping factor), the sum over pages q that link to p has each element normalized by the number of outlinks from page q, and $W_U(q)$ is the weight of page q based on profile U. The reader will immediately note that it is the weight vector that generalizes PageRank to the personalized case, and that the definition readily reverts to the traditional PageRank in the special case where $W_U(q) = 1$ irrespective of users or pages.

3.2 User Profile from Internet Domains

In this paper we study user profiles based on URL features. Profiles can be based on any URL features such as path keywords, protocols, host names, etc. Let us focus on Internet (DNS) domains. A user is expected to input his/her interests as a set of domain features, before query time. When a query is submitted by the user, we retrieve the personalized PageRank vector corresponding to his/her profile in order to rank the hits satisfying the query.

A domain profile is a binary feature vector. Domain features are divided into N groups or categories, such as geographic or topical features (N is a parameter). When assigning a weight to a URL based on its features we use the following algorithm, which takes a URL in input and returns a corresponding normalized weight. We first analyze the fully qualified domain name of the server host. This domain analysis creates a URL feature vector. Let n be the number of matched feature groups between the user profile vector U and the feature vector of page p's URL. The normalized weight for this URL and user profile is then defined by $W_U(p) = 2^{n-N}$.

Let us illustrate the above algorithm with an example. Consider a site p that belongs to the United Kingdom's government, `http://www.direct.gov.uk`; and a user profile U with *geographic* domain features `(America, Europe)` and *topical* domain features `(Educational, Commercial)`. Let us also assume that $N = 2$, i.e. we consider only the two groups of geographic and topical domain features. In this example the domain analysis yields a URL feature vector `(Europe, Government)` from the domains `uk` and `gov`. As a result $n = 1$ feature groups are matched, namely the geographic feature `Europe`, and therefore $W_U(p) = 0.5$.

4. Evaluation

To evaluate our methodology we carried out a Web crawl and implemented an extension of the Nutch[5] open-source search engine to combine similarity and PageRank computations. We then conducted a user study to explore the improvement in precision/recall when applying our idea of personalizing PageRank based on URL features.

4.1 Design and Architecture

For our experiment we used a collection of pages obtained by crawling the Web in April 2004, starting from three seed categories ("Education," "Region," and "Government") of the Yahoo Directory[6]. The resulting

[5] `http://www.nutch.org`
[6] `http://dir.yahoo.com`

Table 1. Domain features used in the profiles.

Number	Feature	Category	Domains
1	Commercial	Topical	`com`
2	Military	Topical	`mil`
3	Government	Topical	`gov`
4	Non-Profit Organizations	Topical	`org`
5	Network Organizations	Topical	`net`
6	Educational	Topical	`edu`
7	America	Geographic	`ca,us,...`
8	Asia	Geographic	`jp,tw,...`
9	Europe	Geographic	`it,uk,...`

crawl data consists of 107,890 URLs and 468,410 links forming a Web graph. When calculating the PageRank scores, one must deal carefully with the problem of danglink links — nodes that don't have known outlinks — as explained in [5, 17]. It has also been showed that it is possible to compute PageRank scores with missing outlink information and keep PageRank errors under control [18]. To minimize error rate in PageRank calculations and maximize the size of our Web graph, we used an additional imaginary node to distribute the PageRank from danglink links back to the graph. Each dangling link node was linked to the imaginary node, and this was linked to all of the nodes without known inlinks. This approach is similar to the one described in [14, 17].

Offline, we pre-computed $2^9 - 1 = 511$ personalized PageRank vectors including a plain PageRank vector (the case where all features are selected and the case where no features are selected are considered identical and equivalent to plain PageRank). Personalized PageRank vectors were computed based on predefined domain profiles. In our design, a domain profile may consist of 9 features: 6 topical and 3 geographic domain features as illustrated in Table 1. PageRank vectors are computed once and stored prior to query time. We used the compressed sparse row (CSR) data structure to store the adjacency matrix representation of our Web graph. The CSR data structure stores its row and column index for each entry. Entries are listed one row after another. This is simply done by a data structure which is a triplet `(i,j,value)`. We defined a Java object to represent a triplet and a global array to store the triplet objects. This way we do not store non-zero values unnecessarily. Also, to avoid increasing the online query time, we updated the Nutch index system so that it can also accommodate PageRank scores along with the existing information such as anchor text, keywords, and similarity score. This prevents the heavy I/O overhead of reading the PageRank scores from an external database or file store. We used global parallel arrays for vertices and PageRank vectors.

For online query processing and to manage our user study, we implemented various user interfaces using Java Server Pages. When a query is submitted, we use the Nutch search mechanism to retrieve the hits. Nutch uses a TFIDF based similarity metric [1, 2] to rank hits satisfying a query, returning a similarity score with each hit. We reorder the hits based on plain and personalized PageRank scores — the latter based on the profile of the user who submitted the query. We use three global arrays to store the ranking scores of the hits based on these three different ranking mechanism. We then multiply the similarity-based Nutch score by the plain PageRank score to obtain the final ranking score of each hit for ordinary PageRank. Likewise, we compute the final ranking score for personalized PageRank by multiplying the Nutch score by the weighted PageRank scores corresponding to the user profile.

4.2 User Study

We conducted a preliminary user study to compare the performance of the three ranking methods based on pure similarity, plain PageRank and weighted (personalized) PageRank. We asked each volunteer to use our personalized search facility after they input their domain profiles into our system. There were 5 human subjects who contributed to our small user study with a total of 10 queries. We realize that recall and precision values are dependent on whether the human subjects in a study are experienced searchers or not. An experienced searcher may bias recall and precision by composing queries that result in very many or very few relevant results. To this end, we conducted our user study with a group of graduate students and did not give out any information about the main goal of the search engine. Volunteers were only expected to select relevant URLs satisfying their choice of preferences.

After submitting a query, a volunteer was shown a single screen with the search results from the three ranking mechanisms mixed together. For each query, the top 10 results from each ranking method were merged and then randomly shuffled before being shown to the volunteer. As an example, suppose that Nutch returns at least 30 results satisfying a query. These hits are reranked based on the two PageRank methods (each combined with similarity). If the top 10 hits from the three ranking mechanisms turn out to have no overlap with each other, then the volunteer would be shown 30 hits in random order as a result of his/her query. If the top hits from the different ranking mechanisms overlap with each other, then the number of results shown to the user would range from 10 to 30.

The Web-based user study interface was designed to be easy to use and reduce possible mistakes in user evaluations. Our interface consists of three stages. In the first stage, users provide identification information (to associate users with queries across sessions) and choices of interests in topical and geographic domains. This is illustrated in Figure 1. The second stage is the Web search facility, through which users are expected to submit their queries. The third stage of the user interface is where the shuffled top hits of the three ranking mechanisms are displayed to the user. Here we also provide facilities for displaying the hit pages and selecting relevant pages satisfying the user query. The third stage of our user interface is also shown in Figure 1.

5. Results

Once a user submits the evaluation for the results of a query, we calculate precision/recall pairs for that query as follows. For each hit h we have the rank from each of the three ranking scores, and the user's binary (0/1) relevance assessment u. Therefore for each ranking mechanism r and query q we compute precision and recall at rank i:

$$\begin{aligned} precision_r(i,q) &= \frac{1}{i}\sum_{j=1}^{i} u(h(r,j,q)) \\ recall_r(i,q) &= \frac{1}{|h : u(h(q)) = 1|}\sum_{j=1}^{i} u(h(r,j,q)) \end{aligned}$$

where $h(r, j, q)$ is the hit ranked j by ranking mechanism r for query q.

Precision-recall plots for the three ranking mechanism and for $i = 1, \ldots, 10$ are shown in Figure 2. The measurements are averaged across the 10 queries posed by the users.

Both PageRank based ranking methods outperform pure similarity based ranking; this is not surprising — it is quite established that link analysis helps to identify important pages. The more important question here is the difference between the two PageRank based methods. The plots suggest that personalized PageRank vectors can help improve the quality of results returned by a search engine. At very low recall level, the two perform the same. At higher recall levels, personalized PageRank achieves better precision. While these preliminary results are not highly significant statistically given the very small user study, they are promising. Domain-based personalization seems to provide us with a mechanism to adjust the estimated importance of pages based on user preferences.

6. Conclusions and Future Work

In this paper we introduced a metodology for personalizing PageRank based on user profiles built from URL features such as server host domains. We outlined the implementation of a simple personalized Web search

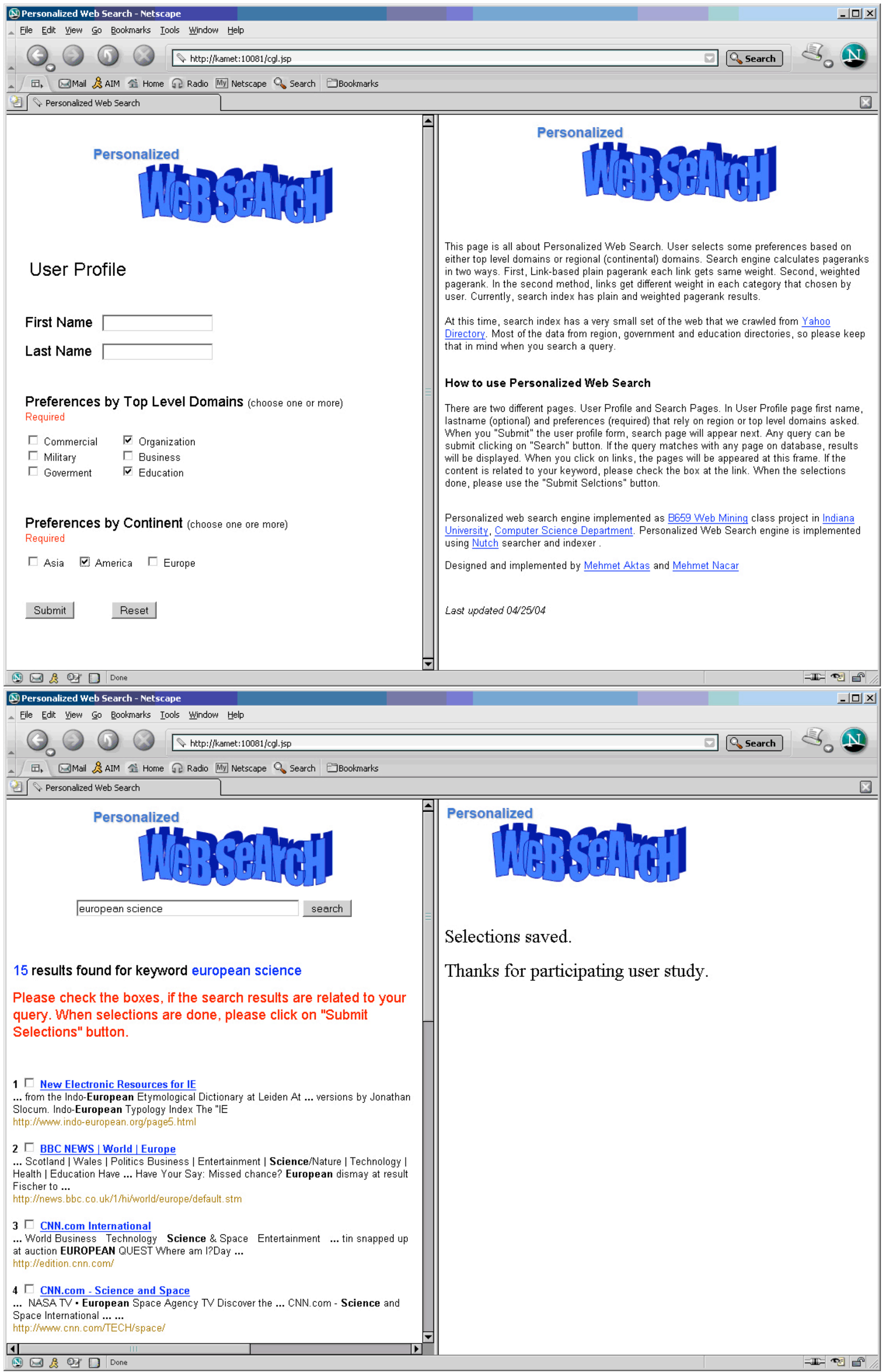

Figure 1. Web-based interface to conduct our user study. Top: User Profile Page. The user enters his/her identification information and choices of topical and geographic domain interests to create a user profile. Bottom: Web Search Page. The user submits a query and selects any relevant results, which are saved with each query.

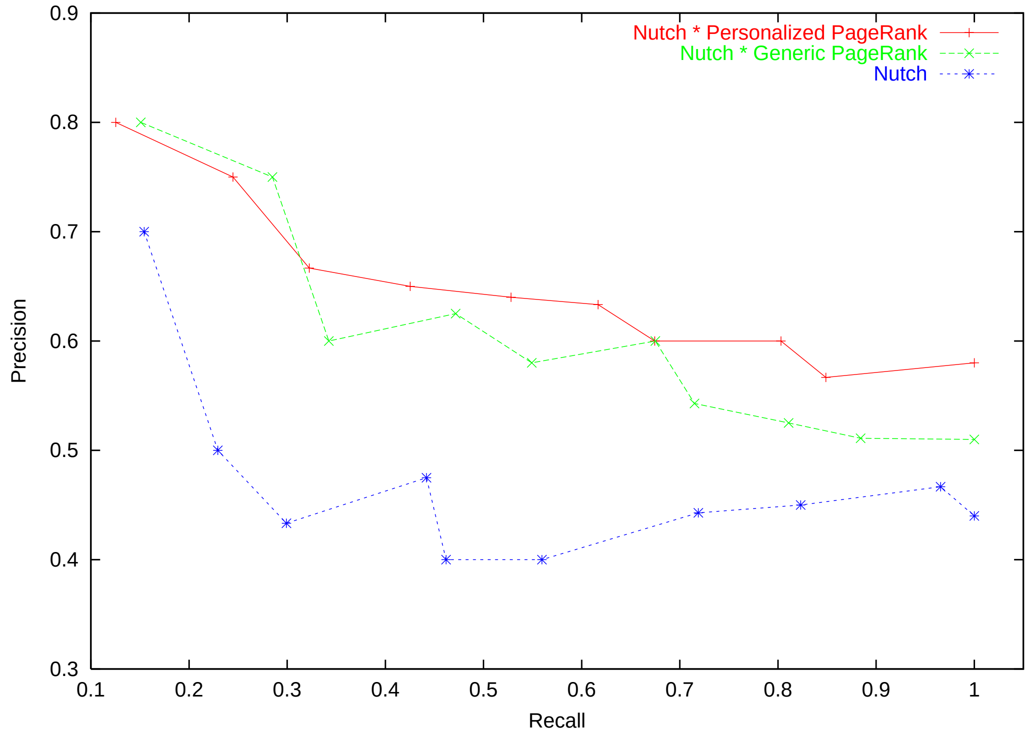

Figure 2. Precision-recall plots of three different ranking mechanisms.

engine based on these ideas, and on a small set of URL domain features. Preliminary results based on a limited Web crawl and a small user study suggest that personalized PageRank vectors can improve the quality of results.

In future work we will explore more features of URLs in personalizing PageRank. We also plan to study efficient ways of calculating PageRank scores in order to enable our personalized search approach to scale with larger user profiles and Web crawls.

Acknowledgments

We are grateful to the 5 volunteers who helped with the user study. FM is partially supported by NSF award N. IIS-0348940. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] van Rijsbergen, C.: Information Retrieval. Butterworths, London (1979) Second edition.

[2] Salton, G., McGill, M.: An Introduction to Modern Information Retrieval. McGraw-Hill, New York, NY (1983)

[3] Brin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. Computer Networks **30** (1998) 107–117

[4] Kleinberg, J.: Authoritative sources in a hyperlinked environment. Journal of the ACM **46** (1999) 604–632

[5] Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: Bringing order to the Web. Technical report, Stanford University Database Group (1998)

[6] Brin, S., Motwani, R., Page, L., Winograd, T.: What can you do with a Web in your pocket. IEEE Data Engineering Bulletin **21** (1998) 37–47

[7] Arasu, A., Cho, J., Garcia-Molina, H., Paepcke, A., Raghavan, S.: Searching the web. ACM Trans. Inter. Tech. **1** (2001) 2–43

[8] Langville, A.N., Meyer, C.D.: Deeper inside PageRank. Internet Mathematics (Forthcoming)

[9] Langville, A.N., Meyer, C.D.: A survey of eigenvector methods of Web information retrieval. SIAM Review (Forthcoming)

[10] Haveliwala, T.: Topic-sensitive PageRank. In Lassner, D., De Roure, D., Iyengar, A., eds.: Proc. 11th International World Wide Web Conference, ACM Press (2002)

[11] Richardson, M., Domingos, P.: The intelligent surfer: Probabilistic combination of link and content information in PageRank. In: Advances in Neural Information Processing Systems 14, Cambridge, MA, MIT Press (2002) 1441–1448

[12] Jeh, G., Widom, J.: Scaling personalized Web search. In: Proc. 12th International World Wide Web Conference. (2003)

[13] Haveliwala, T.: Efficient computation of pagerank. Technical report, Stanford Database Group (1999)

[14] Kamvar, S.D., Haveliwala, T.H., Manning, C.D., Golub, G.H.: Exploiting the block structure of the Web for computing PageRank. Technical report, Stanford University (2003)

[15] Kamvar, S.D., Haveliwala, T.H., Manning, C.D., Golub, G.H.: Extrapolation methods for accelerating the computation of pagerank. In: Proc. 12th International World Wide Web Conference. (2003)

[16] Kamvar, S.D., Haveliwala, T.H., Golub, G.H.: Adaptive methods for the computation of PageRank. Technical report, Stanford University (2003)

[17] Eiron, N., McCurley, K., Tomlin, J.: Ranking the Web frontier. In: Proc. 13th conference on World Wide Web, ACM Press (2004) 309–318

[18] Acharyya, S., Ghosh, J.: Outlink estimation for pagerank computation under missing data. In: Alt. Track Papers and Posters Proc. 13th International World Wide Web Conference. (2004) 486–487