

Using Hyperlink Features to Personalize Web Search

Mehmet S. Aktas[†], Mehmet A. Nacar[†], and Filippo Menczer^{†‡}

[†] Computer Science Department
[‡] School of Informatics
Indiana University
Bloomington, IN 47405 USA
{maktas,mnacar,fil}@indiana.edu

Abstract. Personalized search has gained great popularity to improve search effectiveness in recent years. The objective of personalized search is to provide users with information tailored to their individual contexts. We propose to personalize Web search based on features extracted from hyperlinks, such as anchor terms or URL tokens. Our methodology personalizes PageRank vectors by weighting links based on the match between hyperlinks and user profiles. In particular, here we describe a profile representation using Internet domain features extracted from URLs. Users specify interest profiles as binary vectors where each feature corresponds to a set of one or more DNS tree nodes. Given a profile vector, a weighted PageRank is computed assigning a weight to each URL based on the match between the URL and the profile. We present promising results from an 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.

1 Introduction

The explosive growth of documents in the Web makes it difficult to determine which are the most relevant documents for a particular user, given a general query. 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 [3, 4].

Personalized search has gained great popularity to improve search effectiveness in recent years [10, 12, 29]. The objective of personalized search is to provide users with information tailored to their individual contexts. We propose to personalize Web search based on features extracted from hyperlinks, such as anchor terms or URL tokens. Our methodology personalizes PageRank vectors by weighting links based on the match between hyperlinks and user profiles. In particular, here we describe a profile representation using Internet domain features extracted from URLs.

We identify two aspects of link analysis. One is the global importance of pages as estimated from analyzing the Web link graph structure. There is a major body of research exploring retrieval techniques based on link popularity such as PageRank [3] and HITS [4]. Another aspect of link analysis is the structure of the hyperlinks themselves. For example, anchor text has been shown to be a very good predictor of content of the linked page [27, 28]. One can expect that keywords in the anchor text of a link might be highly related with the content of that page. The accuracy and quality of a page can also be estimated by looking at its URL. Web pages published under an educational institution Web site might be deemed to have higher prestige compared to those published under free Web hosting sites. In this research, we combine these two aspects of link analysis: PageRank and hyperlink structure to improve search effectiveness through personalized search.

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 Google¹ 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 [7–9].

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 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 hyperlink features such as Internet domains. In this scenario, 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 hyperlink and the profile features. A

¹ <http://www.google.com>

² See Section 2 for an exception to this, currently under beta-testing by Google.

weighted PageRank vector is then computed based on URL weights, and used at query time to rank results. We present promising results from an experiment in which users were allowed to select among nine hyperlink 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 [10–12] 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 enormous 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 hyperlink 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 research [13–16]. 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.

Web personalization and customization are related, but distinct concepts. The objective of personalization is to provide users with information tailored to their individual contexts. Mostafa [21] defines customization systems as involving more functionalities than personalization. This includes factors such as location and time to identify the structure and presentation of information. Methodologies for the system to learn about the user’s informational context can be defined as information request, use and demand patterns. Our system requires that the user provides personal information by creating an individual profile based on domain profiles. Therefore we view our approach as an example of personalization and not customization.

Personalization can be achieved in various ways. Erinaki and Vazirgiannis [20], categorize personalization into four groups. In content based filtering the recommendation is done based on the user’s individual preferences. In collaborative filtering the information is sorted based on personal preferences of like-

minded individuals [22]. In rule-based filtering the users are asked questions by the system. The system extracts rules from the answers and then applies these rules to deliver appropriate content to the users. At last, in Web usage mining personalized recommendation takes place by applying data mining techniques to web usage data [22–26]. Our methodology for personalization falls into the first category as we utilize static profiles defined by users for re-sorting search results.

Search can be personalized in two ways: query augmentation and result processing [19]. In the former method the query is compared against user preferences. If the similarity between the query and user preferences is above a threshold the query is augmented with metadata and submitted to the search engine to obtain more precise results. In the latter method the results that are returned by a search engine are re-ranked based on the user’s profile. Here, we focus on personalizing search hits through result processing.

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 (à la Open Directory Project⁴), however due to lack of documentation we are unable to discuss the similarities or differences between that 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 hyperlink 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.

We also note that Baeza-Yates and Davis [30] introduce a variant of PageRank that gives weights to link based on a) relative position in the page, b) tag where the link is contained and c) length of the anchor text. Here, we focus on personalizing PageRank scores based on hyperlink features such as Internet domain features extracted from URLs.

3 Personalized PageRank Based on Hyperlink Features

3.1 Traditional PageRank Computation

PageRank is one of the most well known algorithm that is based on global popularity of web pages. It was first introduced by Brin and Page [3]. The computation of plain PageRank vectors is done as described below.

³ <http://labs.google.com/personalized>

⁴ <http://dmoz.org>

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

where d is the traditional jump probability (or damping factor) and the sum over pages q that link to p has each element normalized by the number of outlinks from page q .

3.2 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.

3.3 User Profile from Internet Domains

In this paper we study user profiles based on hyperlink features. Profiles can be based on any hyperlink 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.

Alternatively, one could extend the profile by adding features associated with the protocols (say, if a user was interested mainly in FTP sites) or with keywords (say, if a user was interested in certain organizations such as webMD). Figure 1 shows the hyperlink feature space.

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

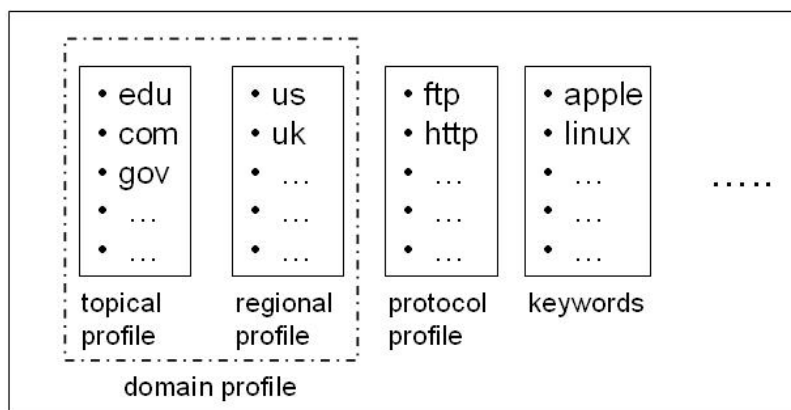


Fig. 1. Various kinds of profiles can be used on hyperlink features. Here we focus on domain-based features.

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 Nutch⁵ 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 hyperlink 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 Directory⁶. The resulting crawl data consists of

⁵ <http://www.nutch.org>

⁶ <http://dir.yahoo.com>

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].

Table 1. Domain features used in the profiles.

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

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 mechanisms. 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 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 30 human subjects who contributed to our user study with a total of 30 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 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 2. 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 2.

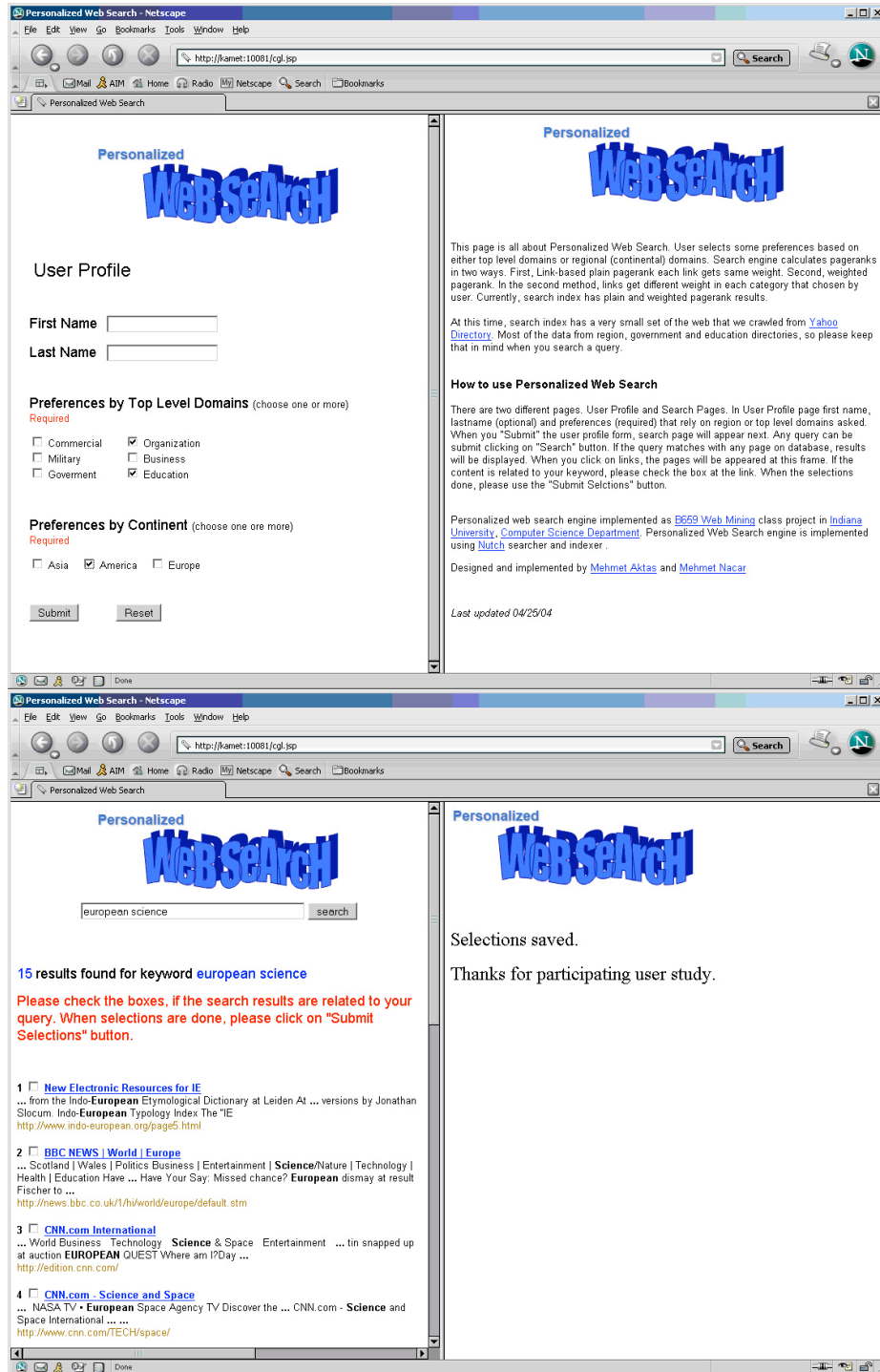


Fig. 2. Web-based interface to conduct our user study. Top: User Profile Page. The user enters his/her identification information and choices of topical and geographic domain interests to create a user profile. Bottom: Web Search Page. The user submits a query and selects any relevant results, which are saved with each query.

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 :

$$precision_r(i, q) = \frac{1}{i} \sum_{j=1}^i u(h(r, j, q))$$

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

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 $i = 1, \dots, 10$ are shown in Figure 3. The measurements are averaged across the 30 queries posed by the users.

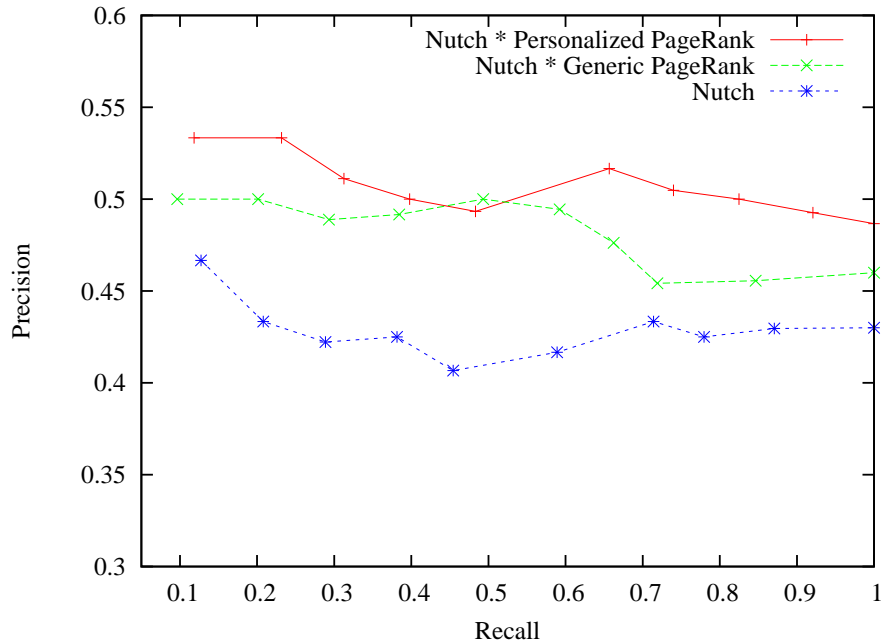


Fig. 3. Precision-recall plots of three different ranking mechanisms.

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. While these results are not highly significant statistically, 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

In this paper we introduced a methodology for personalizing PageRank based on user profiles built from hyperlink features such as server host domains. We outlined the implementation of a simple personalized Web search engine based on these ideas, and on a small set of URL domain features. Results based on a limited Web crawl suggest that personalized PageRank vectors can improve the quality of results.

Acknowledgments

We are grateful to the 30 volunteers who helped with the user study. FM is partially supported by NSF award N. IIS-0348940.

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
19. Pitkow, J., Schutze, H., Cass T., Cooley R., Turnbull D., Edmonds A., Adar E., Breuel T.: Personalized Search. Vol. 42, No. 9 Communication of ACM. (2002)
20. M. Eiriraki, M. Vazirgiannis.: Web Mining for Web Personalization ACM Transactions on Internet Technologies (ACM TOIT). Vol.3. Issue 1
21. Javed Mostafa: Information Customization IEEE Intelligent Systems Vol 17.6 (2002)
22. Sung Ho Ha: Helping Online Customers Decide through Web Personalization IEEE Intelligent Systems Vol 17.6 (2002)
23. M. Jenamani, P. Mohapatra, and S. Ghose: Online Customized Index Synthesis in Commercial Web Sites IEEE Intelligent Systems Vol 17.6 (2002)
24. Nasraoui O., Petenes C.: Combining Web Usage Mining and Fuzzy Inference for Website Personalization. in Proc. of WebKDD 2003 - KDD Workshop on Web mining as a Premise to Effective and Intelligent Web Applications, Washington DC, August 2003, p. 37
25. Mobasher B., Dai H., Luo T., and Nakagawa M.: Effective personalization based on association rule discovery from Web usage data. ACM Workshop on Web information and data management, Atlanta, GA
26. Li J. and Zaiane O.: Using Distinctive Information Channels for a Mission-based Web-Recommender System. In Proc. of "WebKDD-2004 workshop on Web Mining and Web Usage Analysis", part of the ACM KDD: Knowledge Discovery and Data Mining Conference, Seattle, WA.
27. B.D. Davison: Topical locality in the Web. In Proceedings of the 1st International World Wide Web Conference, Geneva, 1994. www1.cern.ch/PapersWWW94/reinpost.ps
28. Shannon Bradshaw and Kristian Hammond: Automatically Indexing Research Papers Using Text Surrounding Citations. In Working Notes of the Workshop on Intelligent Information Systems, Sixteenth National Conference on Artificial Intelligence, Orlando, FL, July 18-19
29. Fang Liu, Clement Yu, Weiyi Meng: Personalized Web Search For Improving Retrieval Effectiveness. IEEE Transactions on Knowledge and Data Engineering, January 2004
30. Ricardo BaezaYates and Emilio Davis: Web Page Ranking using Link Attributes. WWW2004, May 17-22, 2004, New York, New York, USA.