Automatic Identification of User Interest For Personalized Search

Feng Qiu
University of California
Los Angeles, CA 90095
fqiu@cs.ucla.edu

Junghoo Cho
University of California
Los Angeles, CA 90095
cho@cs.ucla.edu

ABSTRACT

One hundred users, one hundred needs. As more and more topics are being discussed on the web and our vocabulary remains relatively stable, it is increasingly difficult to let the search engine know what we want. Coping with ambiguous queries has long been an important part of the research on Information Retrieval, but still remains a challenging task. *Personalized search* has recently got significant attention in addressing this challenge in the web search community, based on the premise that a user's general preference may help the search engine disambiguate the true intention of a query. However, studies have shown that users are reluctant to provide any explicit input on their personal preference. In this paper, we study how a search engine can learn a user's preference *automatically* based on her past click history and how it can use the user preference to personalize search results. Our experiments show that users' preferences can be learned accurately even from little click-history data and personalized search based on user preference yields significant improvements over the best existing ranking mechanism in the literature.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*search process*

General Terms

Algorithms, Human Factors, Experimentation

Keywords

Web search, Personalized search, User profile, User search behavior

1. INTRODUCTION

A number of studies have shown that a vast majority of queries to search engines are short and under-specified [1] and users may have completely different intentions for the *same* query [2]. For example, a real-estate agent may issue the query "office" to look for a vacant office space, while an IT specialist may issue the same query to look for popular Microsoft productivity software.

WWW 2006, May 23–26, 2006, Edinburgh, Scotland.
ACM 1-59593-323-9/06/0005.

To address these differences among the users, there has been a recent surge of interest in *personalized search* to customize search results based on a user's interest. Given the large and growing importance of search engines, personalized search has the potential to significantly improve user experience. For example, according to recent statistics [3] if we can reduce the time users spend on searching for results on Google by a mere 1% through effective personalization, over 187,000 person-hours (21 years!) will be saved each month.

Unfortunately, studies have also shown that the vast majority of users are reluctant to provide any explicit feedback on search results and their interest [4]. Therefore, a personalized search engine intended for a large audience has to *learn* the user's preference *automatically* without any explicit input from the users. In this paper, we study the problem of how we can learn a user's interest automatically based on her past click history and how we can use the learned interest to personalize search results for future queries.

To realize this goal, there exist a number of important technical questions to be addressed. First, we need to develop a reasonable *user model* that captures how a user's click history is related to her interest; a user's interest can be learned through her click history only if they are correlated. Second, based on this model we need to design an effective *learning method* that identifies the user's interest by analyzing the user's click history. Finally, we need to develop an effective *ranking mechanism* that considers the learned interest of the user in generating the search result.

Our work, particularly our ranking mechanism, is largely based on a recent work by Haveliwala on *Topic-Sensitive PageRank* [5]. In this work, instead of computing a *single* global PageRank value for every page, the search engine computes *multiple* Topic-Sensitive PageRank values, one for each topic listed in the Open Directory[1]. Then during the query time, the search engine picks the most suitable Topic-Sensitive PageRank value for the given query and user, hoping that this *customized* version of PageRank will be more relevant than the global PageRank. In fact, in a small-scale user study, Haveliwala has shown that this approach leads to notable improvements in the search result quality if the appropriate Topic-Sensitive PageRank value can be selected by the search engine, but he posed the problem of automatic learning of user interest as an open research issue. In this paper, we try to plug in this final missing piece for an automatic personalized search system and make the following contributions:

[1]http://www.dmoz.org

- We provide a formal framework to investigate the problem of learning a user's interest based on her past click history. As part of this framework, we propose a simple yet reasonable model on how we can succinctly represent a user's interest and how the interest affects her web click behavior.

- Based on the formal user model, we develop a method to estimate her *hidden interest* automatically based on her observable past click behavior. We provide theoretical and experimental justification of our estimation method.

- Finally, we describe a ranking mechanism that considers a user's hidden interest in ranking pages for a query based on the work in [5]. We conduct a user survey to evaluate how much the search quality improves through this personalization. While preliminary, our survey result indicates significant improvement in the search quality — we observe about 25% improvement over the best existing method — demonstrating the potential of our approach in personalizing web search.

The rest of this paper is organized as follows. We first provide an overview of Topic-Sensitive PageRank in Section 2, on which our work is mainly based. In Section 3 we describe our models and methods. Experimental results are presented in Section 4. We finally review related work in Section 5 and conclude the paper in Section 6.

2. BACKGROUND

In this section we briefly present some basic background for our work. We will start by presenting the PageRank algorithm in Section 2.1, then we will proceed to Topic-Sensitive PageRank in Section 2.2.

2.1 PageRank

The key idea behind PageRank is that, when a page p_0 links to a page p, it is probably because the author of page p_0 thinks that page p is important. Thus, this link adds to the importance score of page p. How much score should be added for each link? Intuitively, if a page itself is very important, then its author's opinion on the importance of other pages is more reliable; and if a page links to many pages, the importance score it confers to each of them is decreased. This simple intuition leads to the following formula of computing PageRank: for each page p, let $\mathcal{A}_p$ denote the set of pages linking to p, l_{p_0} denote the number of out-links on page p_0, and $PR(p)$ denote the PageRank of page p, then

$$PR(p) = \sum_{p_0 \in \mathcal{A}_p} PR(p_0)/l_{p_0} \tag{1}$$

Another intuitive explanation of PageRank is based on the *random surfer model* [6], which essentially models a user doing a random walk on the web graph. In this model, a user starts from a random page on the web, and at each step she randomly chooses an out-link to follow. Then the probability of the user visiting each page is equivalent to the PageRank of that page computed using the above formula.

However, in real life, a user does not follow links all the time in web browsing; she sometimes types URL's directly and visits a new page. To reflect this fact, the random surfer model is modified such that at each step, the user follows one of the out-links with probability d, while with the remaining probability $1-d$ she gets bored and jumps to a new page. Under the standard PageRank, the probability of this jump to a page is uniform across all pages; when a user jumps, she is likely to jump to any page with equal probability. Mathematically, this is described as $E(p) = 1/n$ for all p, where $E(p)$ is the probability to jump to page p when she gets bored and n is the total number of web pages. We call such a jump a *random jump*, and the vector $\mathbf{E} = [E(1), \ldots, E(n)]$ a *random jump probability vector*. Thus we have the following modified formula[2]:

$$PR(p) = d * \sum_{p_0 \in \mathcal{A}_p} PR(p_0)/l_{p_0} + (1-d) * E(p) \tag{2}$$

The computed PageRank values are used by search engines during query processing, such that the pages with high PageRank values are generally placed at the top of search results.

Note that under the standard PageRank, every page is assigned a *single global score* independently of any query. Therefore, PageRank can be deployed very efficiently for online query processing because it can be *precomputed* offline before any query. However, because all pages are given a single global rank, this standard PageRank cannot incorporate the fact that the relative importance of pages may change depending on a query.

2.2 Topic-Sensitive PageRank

The *Topic-Sensitive PageRank* scheme (TSPR) proposed in [5] is an interesting extension of PageRank that can potentially provide different rankings for different queries, while essentially retaining the efficiency advantage of the standard PageRank. In the TSPR scheme, multiple scores, instead of just one, are computed for each page, one for every topic that we consider. More precisely, to compute TSPR with respect to topic t, we first define a *biased random jump probability vector* with respect to topic t, $\mathbf{E}_t = [E_t(1), \ldots, E_t(n)]$, as

$$E_t(p) = \begin{cases} 1/n_t & \text{if page } p \text{ is related to topic } t \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where n_t is the total number of pages related to topic t. The set of pages that are considered related to topic t (i.e., the pages whose $E_t(p)$ value is non-zero) are referred to as the *bias set* of topic t. Then the TSPR score of page p with respect to t is defined as

$$TSPR_t(p) = d * \sum_{p_0 \in \mathcal{A}_p} TSPR_t(p_0)/l_{p_0} + (1-d) * E_t(p). \tag{4}$$

Note the similarity of Equations 2 and 4. Under the random surfer model interpretation, the biased vector $\mathbf{E}_t$ means that when a user jumps, she jumps only to the pages related to topic t. Then $TSPR_t(p)$ is equivalent to the probability of the user's arriving at v when her random jumps are biased this way.

Assuming that we consider m topics, m TSPR scores are computed for each page, which can be done offline. Then during online query processing, given a query, the search engine figures out the most appropriate TSPR score and uses it to rank pages.

[2]Here we skip the discussion on how we deal with dangling pages for brevity. See [6] for more details

In [5], Haveliwala computed TSPR values considering the first-level topics listed in the Open Directory and showed that TSPR provides notable performance improvement over the standard PageRank if the search engine can effectively estimate the topic to which the user-issued query belongs. In the paper, the author also posed the problem of automatic identification of user preference as an open research issue, which can help the search engine pick the best TSPR for a given query and user.

3. PERSONALIZED SEARCH BASED ON USER PREFERENCE

We now discuss how we build upon the result of TSPR to personalize search results. In particular, we describe our framework on how we can learn a user's personal preference automatically based on her past click history and how we can use the preference during query time to rank search results for the particular user.

In Section 3.1 we first describe our representation of user preferences. Then in Section 3.2 we propose our user model that captures the relationship between users' preferences and their click history. In Section 3.3 we proceed to how we learn user preferences. Finally in Section 3.4 we describe how to use this preference information in ranking search results.

3.1 User Preference Representation

Given the billions of pages available on the web and their diverse subject areas, it is reasonable to assume that an average web user is interested in a limited subset of web pages. In addition, we often observe that a user typically has a small number of *topics* that she is primarily interested in and her preference to a page is often affected by her general interest in the topic of the page. For example, a physicist who is mainly interested in topics such as science may find a page on video games not very interesting, even if the page is considered to be of high quality by a video-game enthusiast. Given these observations, we may represent a user's preference at the granularity of either *topics* or individual *web pages* as follows:

Definition 1 (Topic Preference Vector) A user's topic preference vector is defined as an m-tuple $\mathbf{T} = [T(1), \ldots, T(m)]$, in which m is the number of topics in consideration and $T(i)$ represents the user's degree of interest in the i^{th} topic (say, "Computers"). The vector $\mathbf{T}$ is normalized such that $\sum_{i=1}^{m} T(i) = 1$. □

Example 1 Suppose there are only two topics: "Computers" and "News," and a user is interested in "Computers" three times as much as she is interested in "News," then the topic preference vector of the user is $[0.75, 0.25]$. □

Instead of the above, we may represent a user's interest at the level of web pages.

Definition 2 (Page Preference Vector) A user's page preference vector is defined as an n-tuple $\mathbf{P} = [P(1), \ldots, P(n)]$, in which n is the total number of web pages and $P(i)$ represents the user's degree of interest in the i^{th} page. The vector $\mathbf{P}$ is normalized such that $\sum_{i=1}^{n} P(i) = 1$. □

In principle, the page preference vector may capture a user's interest better than the topic preference vector, because her interest is represented in more detail. However, we note that our goal is to learn the user's interest through the analysis of her past click history. Given the billions of pages available on the web, a user can click on only a very small fraction of them (at most hundreds of thousands), making the task of learning the page preference vector very difficult; we have to learn the values of a billion-dimension vector from hundreds of thousands data points, which is bound to be inaccurate. Due to this practical reason, we use the topic preference vector as our representation of user interest in the rest of this paper.

We note that the this choice of preference representation is valid only if a user's interest in a page is mainly driven by the topic of the page. We will try to check the validity of this assumption later in the experiment section — even though in an indirect way — by measuring the effectiveness of our search personalization method based on topic preference vectors.

In Table 1, we summarize the symbols that we use throughout this paper. The meaning of some of the symbols will be clear as we introduce our user model.

Symbol	Meaning
n	The total number of web pages
m	The number of topics in consideration
$T(i)$	A user's topic preference on the i^{th} topic
$P(i)$	A user's page preference on the i^{th} page
$V(p)$	A user's probability of visiting page p
$\mathbf{E_t}$	Biased random jump probability vector with respect to topic t
$PR(p)$	PageRank of page p
$TSPR_t(p)$	Topic-Sensitive PageRank of page p on topic t
$PPR_{\mathbf{T}}(p)$	Personalized PageRank of page p for the user whose topic preference vector is $\mathbf{T}$

Table 1: Symbols used throughout this paper and their meanings

3.2 User Model

To learn the topic preference vector of a user from her past click history, we need to understand how the user's clicks are related to her preference. In this section, we describe our user model that captures this relationship. As a starting point, we first describe the *topic-driven random surfer* model.

Definition 3 (Topic-Driven Random Surfer Model) Consider a user with topic preference vector $\mathbf{T}$. Under the topic-driven random surfer model, the user browses the web in a two-step process. First, the user chooses a topic of interest t for the ensuing sequence of random walks with probability $T(t)$ (i.e., her degree of interest in topic t). Then with equal probability, she jumps to one of the pages on topic t (i.e, pages whose $E_t(p)$ values are non-zero). Starting from this page, the user then performs a random walk, such that at each step, with probability d, she randomly follows an out-link on the current page; with the remaining probability $1-d$ she gets bored and picks a new topic of interest for the next sequence of random walks based on $\mathbf{T}$ and jumps to a page on the chosen topic. This process is repeated forever. □

Example 2 Suppose there are only two topics: "Computers" and "News," and a user's topic preference vector is $[0.7, 0.3]$. Under the topic-driven random surfer model, this means that 70% of the the user's "random-walk sessions" (the set of pages that the user visits by following links) start from computer-related pages and 30% start from news-related pages. Note the difference of this model from the standard random surfer model, where the user's random walk may start from any page with equal probability. □

Given this model and the discussion in Section 2.2, we can see that , $TSPR_t(p)$, the Topic-Sensitive PageRank of page p with regard to topic t, is equivalent to the probability of a user's visiting page p when the user is only interested in topic t (i.e., $T(i)$ is 1 for $i = t$ and 0 otherwise).

In general, we can derive the relationship between a user's visit to a page and her topic preference vector $\mathbf{T}$. To help the derivation, we first define a user's *visit probability vector*

Definition 4 (Visit Probability Vector) A user's visit probability vector is defined as an n-tuple $\mathbf{V} = [V(1), \ldots, V(n)]$, in which n is the total number of pages and $V(i)$ represents the user's probability to visit the page i. We assume the vector $\mathbf{V}$ is normalized such that $\sum_{i=1}^{n} V(i) = 1$. □

Given this definition, it is straightforward to derive the following relationship between a user's visit probability vector $\mathbf{V}$ and the user's topic preference vector $\mathbf{T}$ based on the linearity property of TSPR [5]:

Lemma 1 *Consider a user with the topic preference vector* $\mathbf{T} = [T(1), \ldots, T(m)]$ *and the visit probability vector* $\mathbf{V} = [V(1), \ldots, V(n)]$. *Under the topic-driven random surfer model,* $V(p)$*, the probability to visit page* p*, is given by*

$$V(p) = \sum_{i=1}^{m} T(i) * TSPR_i(p) \tag{5}$$

□

Due to space constraints, we provide all detailed proofs in the extended version of this paper [7].

Note that Equation 5 shows the relationship between a user's visits and her topic preference vector if the user follows the topic-driven random surfer model. In practice, however, the search engine can only observe the user's clicks *on its search result*, not the general web surfing behavior of the user. That is, the user clicks that the search engine observes is not based on the topic-driven random surfer model; instead the user's clicks are heavily affected by the rankings of search results. To address this discrepancy, we now extend the topic-driven random-surfer model as follows:

Definition 5 (Topic-Driven Searcher Model) Consider a user with topic preference vector $\mathbf{T}$. Under the topic-driven searcher model, the user always visits web pages through a search engine in a two-step process. First, the user chooses a topic of interest t with probability $T(t)$. Then the user goes to the search engine and issues a query on the chosen topic t. The search engine then returns pages ranked by $TSPR_t(p)$, on which the user clicks. □

To derive the relationship between a user's visit probability vector $\mathbf{V}$ and her topic preference vector $\mathbf{T}$ under this new model, we draw upon the result from a recent study by Cho and Roy [8]. In [8], the authors have shown that when a user's clicks are affected by search results ranked by $PR(p)$, the user's visit probability to page p, $V(p)$, is proportional to $PR(p)^{9/4}$, as opposed to $PR(p)$ as predicted by the random-surfer model. Given this new relationship, it is relatively straightforward to prove the following:

Theorem 1 *Consider a user with the topic preference vector* $\mathbf{T}$ *and the visit probability vector* (V). *Then under the topic-driven searcher model,* $V(p)$ *is given by*[3]

$$V(p) = \sum_{i=1}^{m} T(i) * [TSPR_i(p)]^{9/4} \tag{6}$$

□

Note that Equation 6 is equivalent to Equation 5 except that the term $TSPR_i(p)$ is replaced with $[TSPR_i(p)]^{9/4}$, a replacement coming from the result in [8].

In the rest of this paper, we will assume the above topic-driven searcher model as our main user model and use Equation 6 as the relationship between a user's visit probability vector and her topic preference vector.

3.3 Learning Topic Preference Vector

We now turn our attention to how we can learn a user's topic preference vector from the user's past click history on search results. Roughly, this learning can be done based on our user model, in particular Equation 6, which describes the relationship between a user's visits, her topic preference, and the Topic-Sensitive PageRank values of the pages. Note that among the variables in the equation we can measure $V(p)$ values experimentally by observing the user's clicks on search results as we illustrate in the following example:

Example 3 Suppose there exist 10 pages on the web and through the past searches, the user has visited the first page twice, the second page once and none of the other pages. The user's visit probability vector is then can be estimated as $\mathbf{V} = [\frac{2}{3}, \frac{1}{3}, 0, \ldots, 0]$ given the click history. □

Also, the $TSPR_i(p)$ values in Equation 6 can be computed from Equation 4 based on the hyperlink structure of the web and a reasonable choice of the bias set for each topic. The only unknown variables in Equation 6 are $T(i)$ values, which can be derived from other variables in the equation. Formally, this learning process can be formulated as the following problem:

Problem 1 Given the user visit probability vector $\mathbf{V} = [V(1), \ldots, V(n)]$, and m Topic-Sensitive PageRank vectors $\mathbf{TSPR}_i = [TSPR_i(1), \ldots, TSPR_i(n)]$ for $i = 1, \ldots, m$, find the user's topic preference vector $\mathbf{T} = [T(1), \ldots, T(m)]$ that satisfies

$$\mathbf{V} = \sum_{i=1}^{m} T(i) \cdot \mathbf{TSPR}_i^{9/4} \tag{7}$$

Here, we use the notation $\mathbf{A}^k$ to denote a vector whose elements are the elements of $\mathbf{A}$ raised to the kth power. □

There exist a number of ways to approach this problem. In particular, regression-based and maximum-likelihood methods are two of the popular techniques used in this context:

[3]Even if the $TSPR_i(j)$ values are normalized such that $\sum_{j=1}^{n} TSPR_i(j) = 1$, it may be that $TSPR_i(j)^{9/4}$ values do not satisfy $\sum_{j=1}^{n} TSPR_i(j)^{9/4} = 1$. In Equation 5, we assume that we have already renormalized $TSPR_i(j)$ values such that $\sum_{j=1}^{n} TSPR_i(j)^{9/4} = 1$ for every i.

1. *Linear regression*: Because the topic-driven searcher model is an approximation of what users do and because a search engine can observe a relatively small number of user clicks, the $\mathbf{V}$ vector measured from user clicks cannot be identical to the one prescribed by the model. Linear regression is a popular method used in this scenario. It picks the unknown parameter values such that the difference between what we observe and what the model prescribes is minimized. In our context, it determines the $T(i)$ values that minimize the *square-root error* $\left|\mathbf{V} - \left(\sum_{i=1}^{m} T(i) \cdot \mathbf{TSPR}_i^{9/4}\right)\right|^2$. Assuming that $\mathbf{TSPR}$ is an $n \times m$ matrix whose (i,j) entry is $TSPR_j(i)^{9/4}$, the linear regression method asserts that this error is minimum when the topic preference vector is

$$\mathbf{T} = \left(\mathbf{TSPR}^{\mathrm{t}}\mathbf{TSPR}\right)^{-1}\mathbf{TSPR}^{\mathrm{t}}\mathbf{V}. \tag{8}$$

   Here, $\mathbf{TSPR}^{\mathrm{t}}$ is the transpose of $\mathbf{TSPR}$.

2. *Maximum-likelihood estimator*: The core principle behind a maximum-likelihood estimator is that the unknown parameters of the model should be chosen such that the observed set of events are the most likely outcome from the model. To apply this method, we introduce some new notation. We assume that the user has visited k pages and use p_i to denote the i^{th} visited page. We also define $V_{\mathbf{T}}(p)$ as $\sum_{i=1}^{m} T(i) \cdot \mathbf{TSPR}_i^{9/4}$, the probability that the user visits page p under the topic-driven searcher model when her topic preference vector is $\mathbf{T} = [T(1), \ldots, T(m)]$. Assuming that all user visits are independent, the probability that the user visits the pages $p_1, \ldots, p_k$ is then $\prod_{i=1}^{k} V_{\mathbf{T}}(p_i)$. Then the values of the vector $\mathbf{T}$ is chosen such that this probability is maximized. That is, under the maximum-likelihood estimator,

$$\mathbf{T} = \arg\max_{\mathbf{T}} \left(\prod_{i=1}^{k} V_{\mathbf{T}}(p_i)\right) \tag{9}$$

In our experiments, we have applied both methods to the learning task and find that the maximum-likelihood estimator performs significantly better than the linear regression method. This is in large part due to, again, the sparsity of the observed click history. Since the user does not visit the vast majority of pages, most of the entries in the visit probability vector are zero. Therefore, even though two users may visit completely different sets of pages, the difference between their visit probability vectors is small because the two vectors have the same zero values for most entries. This problem turns out to be not very significant for the maximum likelihood method because it only considers the pages that the user visited during estimation as we can see from Equation 9. In our experiment section, therefore, we report only the results from the maximum-likelihood estimator.

3.4 Ranking Search Results Using Topic Preference Vectors

In this section we describe how we rank search results based on the user's topic preference vector we have learned. Our approach is based on the framework proposed in [5]: Given a query q, we identify the most likely topic of the query q and use the TSPR values corresponding to the identified topic. More precisely, given q, we estimate $PR(T(i)|q)$, the probability that q is related to topic i, and compute the ranking of page p as follows:

$$\sum_{t=1}^{m} Pr(T(i)|q) \cdot TSPR_i(p) \tag{10}$$

That is, we compute the sum of TSPR values weighted by the likelihood that the query is related to each topic. Note that this ranking degenerates into $TSPR_t(p)$ when the topic of the query q is clearly t (i.e., $Pr(T(t)|q) = 1$ only for $i = t$).

In deciding the likely topic of q, we may rely on two potential sources of information: the *user's preference* and the *query itself*.

Example 4 For simplicity, suppose we consider only two topics, "Business" and "Computers." For the query *C++ programming*, it is clear from the query that this is on "Computers" not "Business." For another query *office*, however, its topic is difficult to judge from the query itself. It is only when we consider the preference of the user, — say, the user is an IT specialist who is mostly interested in "Computers" — that the likely topic of the query becomes clear. □

We can design a mechanism that considers both sources of information in identifying the likely topic of a query based upon the Bayesian framework [9, 5]. According to the Bayes' theorem, $P(T(i)|q)$ can be rewritten as

$$\begin{aligned} Pr(T(i)|q) &= \frac{Pr(q, T(i))}{Pr(q)} \\ &= \frac{Pr(T(i)) * Pr(q|T(i))}{Pr(q)} \\ &\propto Pr(T(i)) * Pr(q|T(i)) \end{aligned} \tag{11}$$

Here, note that $Pr(T(i))$ is the probability that the user is interested in topic i, which is captured in the i^{th} term of the user's topic preference vector $\mathbf{T} = [T(1), \ldots, T(m)]$. That is, $Pr(T(i)) = T(i)$. $Pr(q|T(i))$ is the probability that the user issues the query q when she is interested in topic i. As in [5], this probability may be computed by counting the total number of occurrences of terms in query q in the web pages listed under the topic i in the Open Directory.

Given Equations 10 and 11, we can then compute $PPR_{\mathbf{q},\mathbf{T}}(p)$, the *personalized ranking* of page p for query q issued by the user of topic preference vector $\mathbf{T}$, as follows:

$$PPR_{\mathbf{T}}(p) = \sum_{t=1}^{m} T(i) \cdot Pr(q|T(i)) \cdot TSPR_i(p) \tag{12}$$

Note that in the above equation, the term $T(i)$ is the personalization factor based on the user's preference. $Pr(q|T(i))$ is the term that identifies the topic based on the query itself. Later in our experiment section, we will evaluate the effectiveness of these two individual terms by using only one of them during ranking.

3.5 Evaluation Metrics

How can we evaluate the effectiveness of our proposed methods? Given that our primary goal is to learn the user's topic preference vector from her past click history and use this vector to personalize search ranking, we may consider one of the following evaluation metrics:

1. *Accuracy of topic preference vector*: One natural way to evaluate our learning method is to directly measure

the difference between the user's actual topic preference vector and the learned vector. For this purpose, we may use the *relative error* between the two:

$$E(\mathbf{T_e}) = \frac{|\mathbf{T_e} - \mathbf{T}|}{|\mathbf{T}|} \quad (13)$$

Here $\mathbf{T}$ denotes the user's actual topic preference vector and $\mathbf{T_e}$ denotes our estimation based on the user's click history.

2. *Accuracy of personalized ranking*: Our final goal is to personalize search results based on the user's preference. Therefore, we may use the accuracy of the final personalized ranking (as opposed to the accuracy of the user's topic preference vector) as our evaluation metric. We illustrate why this metric may be more useful using the following example.

Example 5 We consider only two topics and 5 pages. Suppose that the two topics are closely related, and their TSPR values are identical. That is, for example,

$$\mathbf{TSPR}_1 = \mathbf{TSPR}_2 = [0.8, 0.1, 0, 0, 0.1].$$

Now consider two users whose topic preference vectors are $\mathbf{T}_1 = [1, 0]$ and $\mathbf{T}_2 = [0, 1]$, respectively. In this scenario, note that even though the users' topic preference vectors are completely different, their visit probability vectors are identical:

$$\begin{aligned} \mathbf{V}_{\mathbf{T}_1} &= 1 \cdot \mathbf{TSPR}_1 + 0 \cdot \mathbf{TSPR}_2 \\ &= 0 \cdot \mathbf{TSPR}_1 + 1 \cdot \mathbf{TSPR}_2 = \mathbf{V}_{\mathbf{T}_2} \end{aligned}$$

Therefore, it is not possible to learn the users' exact topic preference vectors from their visit history.

Fortunately in this case, the difference in the weights of the topic preference vectors do not matter; the personalized ranking for the two users are always the same

$$Pr(T(1)|q) \cdot \mathbf{TSPR}_1 + Pr(T(2)|q) \cdot \mathbf{TSPR}_2 = \mathbf{TSPR}_1.$$

□

The above example illustrates that even though our estimated topic preference vector may not be the same as the user's actual vector, the final personalized ranking can still be accurate, making our learning method useful. To evaluate how well we can compute the user's personalized ranking, we use the *Kendall's τ distance* [10] between our *estimated* ranking and the *ideal* ranking. That is, let σ_k be the sorted list of top-k pages based on the estimated personalized ranking scores. Let σ'_k be the sorted top-k list based on the the personalized ranking computed from the user's true preference vector. Let S be the union of σ_k and σ'_k. Then the Kendall τ distance between σ_k and σ'_k is computed as

$$\tau(\sigma_k, \sigma'_k) = \frac{|(i,j) : i,j \in S, \sigma_k(i) < \sigma_k(j), \sigma'_k(i) > \sigma'_k(j)|}{|S| * |S-1|} \quad (14)$$

The value of Kendall's τ distance ranges between 0 and 1, taking 0 when the two ranks are identical. Given that most users never look beyond the top 20 entries in search results [1], using a k value between 10 and 100 may be a good choice to compare the difference in the ranks perceived by the users.

3. *Improvement in search quality*: The ultimate success of a personalized ranking scheme can be measured by the quality of the search results determined by the users. Given that an effective ranking scheme should place relevant pages close to the top of the search results, we may use the *average rank* of relevant pages in the search result as a measure of its quality if we know what pages users deem relevant to each query.

In the next section, we evaluate the effectiveness of our personalized search framework using all three metrics described above.

4. EXPERIMENTS

In this section we discuss various experiments we have done to evaluate our proposed methods and show the results. We first describe our experimental setup in Section 4.1. Then in Section 4.2 we describe a simulation-based experiment to measure the accuracy of our learning method. Finally in Section 4.3 we present the results from our user survey that measures the perceived quality of our personalized ranking method.

4.1 Experimental Setup

In order to apply the three evaluation metrics described in Section 3.5, we need the following three datasets: (1) users' click history, (2) the set of pages that are deemed relevant to the queries that they issue, and (3) the Topic-Sensitive PageRank values for each page.

To collect these data, we have contacted 10 subjects in the UCLA Computer Science Department and collected 6 months of their search history by recording all the queries they issued to Google and the search results that they clicked on. Table 2 shows some high-level statistics on this query trace.

Statistics	Value
# of subjects	10
Collection period	04/2004 – 10/2004
Avg # of queries per subject	255.6
Ave # of clicks per query	0.91

Table 2: Statistics on our dataset

To identify the set of pages that are relevant to queries, we carried out a human survey. In this survey, we first picked the most frequent 10 queries in our query trace, and for each query, each of the 10 subjects were shown 10 randomly selected pages in our repository that match the query. Then the subjects were asked to select the pages they found relevant to their own information need for that query. On average 3.1 (out of 10) search results were considered relevant to each query by each user in our survey.

Finally, we computed TSPR values for 500 million web pages collected in a large scale crawl in 2005. That is, based on the link structure captured in the snapshot, we computed the original PageRank and the Topic-Sensitive PageRank values for each of the 16 first-level topic listed in the Open Directory. The computation of these values was performed on a workstation equipped with a 2.4GHz Pentium 4 CPU and 1GB of RAM. The computation of 500 million TSPR values roughly took 10 hours to finish for each topic on the workstation.

4.2 Accuracy of Learning Method

In this section we first try to measure the accuracy of our learning method. Here, we are concerned with both the *accuracy* of our method and the *size* of the click history necessary for accurate estimation. Even if a user's preference can be learned accurately in principle, it may not be possible in practice if it requires a sample size significantly larger than what we can actually collect.

The best way of measuring the accuracy of our method is to estimate the users' topic preferences from the real-life data we have collected, and ask the users how accurate our results are. The problem with this method is that, although users could tell which are the topics they are most interested in, it tends to be very difficult for them to assign an accurate weight to each of these topics. For example, if a user is interested in "Computers" and "News," is her topic preference vector $[0.5, 0.5]$ or $[0.4, 0.6]$? This innate inaccuracy in users' topic preference estimations makes it difficult to investigate the accuracy of our method using real-life data.

Thus, we will use a synthetic dataset generated by simulation based on our topic-driven searcher model:

1. **Generation of topic preference vector.** In our implementation, the number of topics the user is interested in is fixed to K as an experimental parameter. Then we randomly choose K topics and assign random weights to each selected topic. The weights for other topics are set to 0. The vector is normalized to sum up to one.

2. **Generation of click history.** Once we generate a user's topic preference vector, we generate a sequence of L clicks done by the user using the visit probability distribution dictated by our model (Equation 6 in Section 3.2).

4.2.1 Accuracy of estimated topic preference vector

In order to measure the accuracy of the estimated *topic preference vector* we generate synthetic user click traces as described above and measure the relative error of our estimated vector compared to the true preference vector (Equation 13). Figure 1 shows the results from this experiment. In the graph, the height of a bar corresponds to the average relative error at the given K and L values.

From the figure, we see that at the same K value, as the sample size L increases, the relative error of our method generally decreases. For instance, when $K = 4$, the relative error at $L = 10$ is 0.66, but the error goes down to 0.46 when $L = 1000$. For most K values, we also observe that beyond the sample size $L = 100$, the decreasing trend of the relative error becomes much less significant. For example, when $K = 4$, the relative error at $L = 100$ is 0.4, but the error actually gets slightly larger to 0.41 when $L = 500$. In the graph, an interesting number to note is when $K = 3$ and $L = 100$, because this is the setting that we expect to see in practical scenarios.[4] At this parameter setting ($K = 3$, $L = 100$) the average relative error from our estimation is 0.3, indicating that we can learn the user's topic preference vector with 70% accuracy. The graph also shows that as the user gets interested in more topics (i.e., as K increases), the accuracy of our method generally gets worse. This trend is expected because when K is large the user visits pages on diverse topics. Therefore, we need to collect a larger number of overall clicks, to obtain a similar number of clicks on the pages on a particular topic.

[4]This statement is based on the fact that we were able to collect an average of 200 clicks per user in 7 months and that the analysis of this trace indicates that a typical user is interested in 3 to 4 topics out of 16.

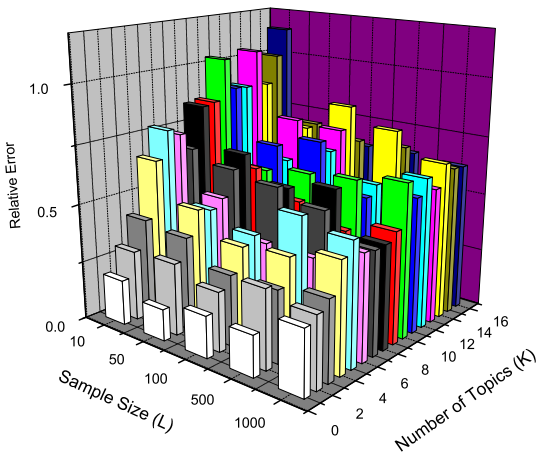

Figure 1: Relative errors in estimated weights

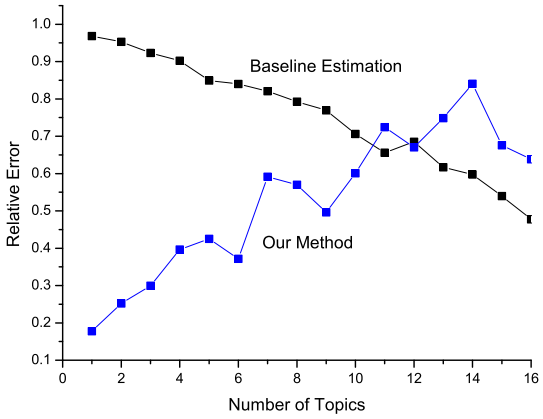

Figure 2: Comparison of relative errors in estimated weights on sample size 100

In Figure 2, we try to see whether our estimation is meaningful at all by comparing it against the baseline estimation, which assumes an equal weight for every topic (i.e., $T(i) = \frac{1}{16}$ for $i = 1, \ldots, 16$). The graph for our method is based on when $L = 100$. From the graph, we can see that when the users are interested in a relatively few number of topics ($1 \leq K \leq 6$), our estimate can be meaningful. The relative error is significantly smaller than for the baseline estimation, indicating that we do get to learn the user's general preference. However, when the user interest is very diverse, the graph shows that we need to collect much more user-click-history data in order to obtain a meaningful esti-

mate for the user preference.

4.2.2 Accuracy of Personalized PageRank

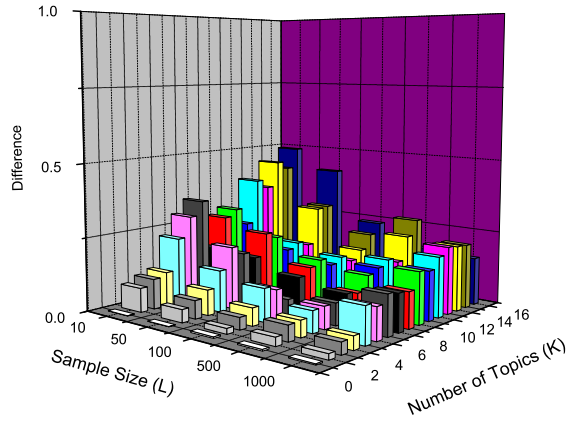

Figure 3: Differences in rankings of top 20 pages

We now investigate the accuracy of our *personalized ranking*. To measure this accuracy, we again generate synthetic user click data based on our model and estimate the user's topic preference vector. We then compute the Kendall's τ distance between the ranking computed from the *estimated* preference vector and the ranking computed from the true preference vector. Figure 3 shows the results when the distance is computed for the top 20 pages.[5] We can see from the figure that, in spite of the relatively large relative error reported in the previous section, our method produces pretty good rankings. For example, when $K = 3$ and $L = 100$, the distance is 0.05, meaning that only 5% of the pairs in our ranking are reversed compared to the true ranking.

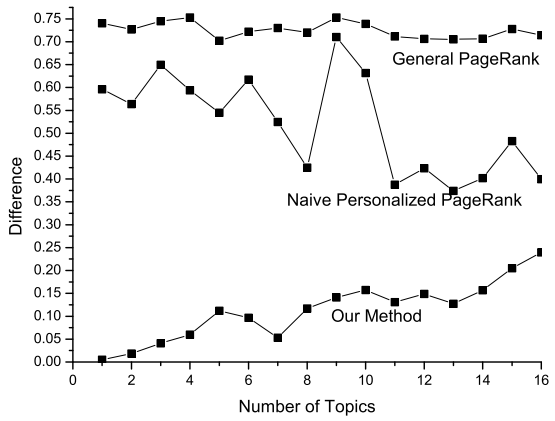

Figure 4: Comparison of differences in rankings of top 20 pages on sample size 100

[5]We also computed the distance for top-k pages for larger k values and the results were similar.

We now compare our personalized ranking against two other baseline rankings: (1) a ranking based on general PageRank and (2) a ranking based on Equation 12, but assuming that all topic weights are equal (i.e., $T(i) = \frac{1}{16}$ for $i = 1, \ldots, 16$). This comparsion will indicate *how much improvement* we get from our topic preference estimation. In Figure 4 we compare the Kendall's τ distance of the three ranking methods for our synthetic data. From the graph, we can see that our method consistently produces much smaller distances than the two baseline rankings, indicating the effectiveness of our learning method.

4.3 Quality of Personalized Search

We now try to measure how much our personalization method improves the overall quality of search results based on our user survey. To measure this improvement we compare the following four ranking mechanisms:

- *General PageRank*: Given a query, we rank the pages that match the query based on their global PageRank values.
- *Topic-Sensitive PageRank*: We rank pages assuming that the user is interested in all topics. That is, we rank pages based on $PPR_{\mathbf{T}}(p)$ of Equation 12, but assuming that $T(i) = \frac{1}{16}$ for $i = 1, \ldots, 16$. This represents a ranking method that does not take the user's preference into account.
- *Personalized Topic-Sensitive PageRank*: We rank pages based on Equation 12 using the estimated topic preference vector, but excluding the second term $Pr(q|T(i))$. That is, we rank pages by $\sum_{i=1}^{m} T(i) \cdot TSPR_i(p)$. This represents a ranking method that uses the user preference, but not the query in identifying the likely topic of the query.
- *Query-Biased Personalized Topic-Sensitive PageRank*: We rank pages based on Equation 12 without omitting any terms. This represents a ranking method that uses both the user preference and the query to identify the likely topic of the query.

In order to measure the quality of a ranking method, we use the data collected from the user survey described in Section 4.1. In the survey, for each of the 10 most common queries from our query trace, we asked our subjects to select the pages that they consider relevant to their own information need among 10 random pages in our repository that match the query. Considering that the queries chosen by us might not be the ones the users would issue in real life (and thus users may not care about the performance of our method on the queries), we also asked each subject to give the probability of her issuing each query in real life. Then for each of our subjects and queries, we compute the weighted average rank of the selected pages using the following formula under each ranking mechanism:

$$AvgRank(u, q) = \sum_{p \in S} Pr(q|u) * R(p) \qquad (15)$$

Here S denotes the set of pages the user u selected for query q, $Pr(q|u)$ denotes the probability that user u issues query q, and $R(p)$ denotes the rank of page p by the given ranking mechanism. Smaller AvgRank values indicate better placements of relevant results, or better result quality.

In Figure 5 we aggregate the results by queries to show how well our methods perform for different queries. The horizontal dotted lines in the graph show the average *AvgRank* over all queries. We can see that our personalization scheme is very effective and achieves significant performance improvement over traditional methods for most queries. The average improvements of Personalized PageRank and Query-Biased Personalized PageRank over Topic-Sensitive PageRank, over all queries, are 32.8% and 24.9%, respectively.[6]

We note that, from our experimental results, although the Query-Biased Personalized PageRank takes more information (both the user's topic preference and the query she issues) into consideration, it performs worse than Personalized PageRank in most cases. This result is quite surprising, but the difference is too small to draw a meaningful conclusion. It will be an interesting future work to see whether the Personalized PageRank indeed performs better than the Query-Biased Personalized PageRank in general and if it does, investigate the reason behind it.

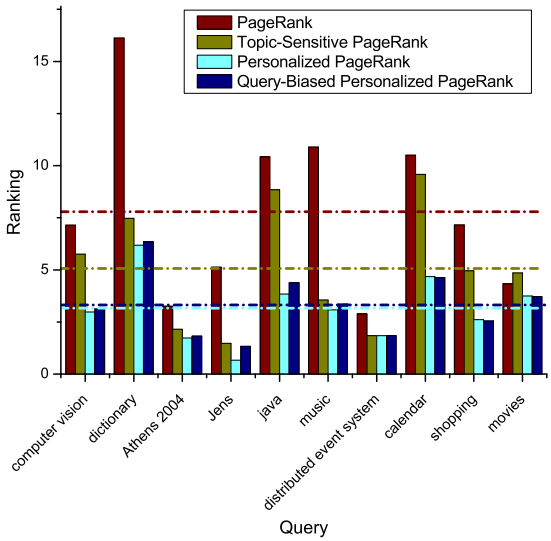

Figure 5: Comparison of weighted average rankings of selected pages by queries (Lower is better)

In Figure 6 we aggregate the results by participants to show the overall effectiveness of our methods for each user. The horizontal dotted lines in the graph show the average results of all participants. We can see that the two personalization-based methods outperform the traditional methods in all cases. For example, the Personalized PageRank method outperforms Topic-Sensitive PageRank by more than 70% for the 2^{nd} participant, while for the 6^{th} participant the improvement is only around 7%. This demonstrates that the effectiveness of personalization depends on the user's search habits. For example, if the user is not likely to issue ambiguous queries, then Topic-Sensitive PageRank should be able to capture her search intentions pretty well, thus decreasing the improvement our methods could achieve. Nevertheless, our methods still greatly improve over the traditional ones overall. The average improvements of Personalized PageRank and Query-Biased Personalized PageRank over Topic-Sensitive PageRank, over all participants, are 24.2% and 15.2%, respectively.

[6]These improvements are statistically significant under the two-sample t-test at the 0.05 level.

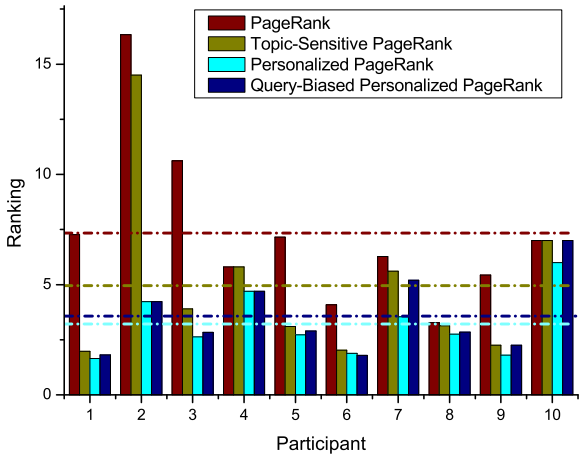

Figure 6: Comparison of weighted average rankings of selected pages by participants (Lower is better)

5. RELATED WORK

After the original PageRank paper [11] was published, there has been considerable research on Personalized PageRank [5, 12, 13, 14, 15]. A large body of this work studies the scalability and performance issues because computing Personalized PageRank for every user may not scale to billions of users. For example, [5, 13, 14] provide a framework to limit the bias vector space during the computation of PageRanks, so that acceptable performance can be achieved. Other than the scalability studies, [12] tries to tailor the PageRank vectors based on query terms (but not by individual users). In [15] Personalized PageRanks are computed based on the user profiles explicitly specified by the users. Our work is different from this body of work in that we focus on developing an automatic learning mechanism for user preferences, so that they can be used to compute Personalized PageRank.

Researchers have also proposed ways to personalize web search based on ideas other than PageRank [16, 17, 18]. For example, [16] extends the well-known HITS algorithm by artificially increasing the authority and hub scores of the pages marked "relevant" by the user in previous searches. [17] explores ways to consider the topic category of a page during ranking using user-specified topics of interest. [18] does a sophisticated analysis on the correlation between users, their queries and search results clicked to model user preferences, but due to the complexity of the analysis, we believe this method is difficult to scale to general search engines.

There also exists much research on learning a user's preference from pages she visited [19, 20, 21]. This body of work, however, mainly relies on *content analysis* of the visited pages, differently from our work. In [19], for example, multiple TF-IDF vectors are generated, each representing the user's interests in one area. In [20] pages visited by the user are categorized by their similarities to a set of precategorized pages, and user preferences are represented by the topic categories of pages in her browsing history. In [21]

the user's preference is learned from both the pages she visited and those visited by users similar to her (collaborative filtering). Our work differs from these studies in that pages are characterized by their Topic-Sensitive PageRanks, which are based on the *web link structure*. It will be an interesting future work to develop an effective mechanism to combine both the content and the web link structure for personalized search.

Finally, Google[7] has started a beta-testing of a new personalized search service[8], which seems to estimate a searcher's interests from her past queries. Unfortunately, the details of the algorithm are not known at this point.

6. CONCLUSION

In this paper we have proposed a framework to investigate the problem of personalizing web search based on users' past search histories without user efforts. In particular, we first proposed a user model to formalize users' interests in web pages and correlate them with users' clicks on search results. Then, based on this correlation, we described an intuitive algorithm to actually learn users' interests. Finally, we proposed two different methods, based on different assumptions on user behaviors, to rank search results based on the user's interests we have learned.

We have conducted both theoretical and real-life experiments to evaluate our approach. In the experiment on synthetic data, we found that for a reasonably small user search trace (containing 100 past clicks on results), the user interests estimated by our learning algorithm can be used to pretty accurately predict her view on the importance of web pages, which is expressed by her Personalized PageRank, showing that our method is effective and easily applicable to real-life search engines. In the real-life experiment, we applied our method to learn the interests of 10 subjects we contacted, and compared the effectiveness of our method in ranking future search results for them to traditional PageRank and Topic-Sensitive PageRank. The results showed that, on average, our method performed between 25%–33% better than Topic-Sensitive PageRank, which turned out to be much better than PageRank.

In the future we plan to expand our framework to take more user-specific information into consideration. For example, users' browsing behaviors, email information, etc. The difficulties in doing this include integration of different information sources, modeling of the correlation between various information and the user's search behavior, and efficiency concerns. We also plan to design more sophisticated learning and ranking algorithms to further improve the performance of our system.

7. REFERENCES

[1] B.J. Jansen, A. Spink, and T Saracevic. Real life, real users, and real needs: A study and analysis of user queries on the Web. *Information Processing and Management*, 36(2):207 – 227, 2000.

[2] Robert Krovetz and W. Bruce Croft. Lexical ambiguity and information retrieval. *Information Systems*, 10(2):115–141, 1992.

[3] Nielsen netratings search engine ratings report. http://searchenginewatch.com/reports/article.php/2156461, 2003.

[4] J. Carroll and M. Rosson. The paradox of the active user. *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*, 1987.

[5] T. Haveliwala. Topic-Sensitive PageRank. In *Proceedings of the Eleventh Intl. World Wide Web Conf.*, 2002.

[6] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. In *Proc. of WWW '98*, 1998.

[7] F. Qiu and J. Cho. Automatic identification of user preferences for personalized search. Technical report, UCLA Computer Science Department, 2005.

[8] J. Cho and S. Roy. Impact of Web search engines on page popularity. In *Proc. of WWW '04*, 2004.

[9] Dimitri P. Bertsekas and John N. Tsitsiklis. *Introduction to Probability*. Athena Scientific, 2002.

[10] M. Kendall and J. Gibbons. *Rank Correlation Methods*. Edward Arnold, London, 1990.

[11] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the Web. Technical report, Stanford Digital Library Technologies Project, 1998.

[12] M. Richardson and P. Domingos. The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2002.

[13] G. Jeh and J. Widom. Scaling personalized web search. In *Proceedings of the Twelfth Intl. World Wide Web Conf.*, 2003.

[14] S. Kamvar, T. Haveliwala, C. Manning, and G. Golub. Exploiting the block structure of the web for computing PageRank. Technical report, Stanford University, 2003.

[15] M. Aktas, M. Nacar, and F. Menczer. Personalizing PageRank based on domain profiles. In *Proc. of WebKDD 2004: KDD Workshop on Web Mining and Web Usage Analysis*, 2004.

[16] F. Tanudjaja and L. Mui. Persona: A contextualized and personalized web search. In *Proc. of the 35th Annual Hawaii International Conference on System Sciences*, 2002.

[17] P. Chirita, W. Nejdl, R. Paiu, and C. Kohlschuetter. Using ODP metadata to personalize search. In *Proceedings of ACM SIGIR '05*, 2005.

[18] J. Sun, H. Zeng, H. Liu, Y. Lu, and Z. Chen. CubeSVD: A novel approach to personalized web search. In *Proceedings of the Fourteenth Intl. World Wide Web Conf.*, 2005.

[19] L. Chen and K. Sycara. Webmate: a personal agent for browsing and searching. *Proc. 2nd Intl. Conf. on Autonomous Agents and Multiagent Systems*, pages 132–139, 1998.

[20] A. Pretschner and S. Gauch. Ontology based personalized search. In *ICTAI*, pages 391–398, 1999.

[21] K. Sugiyama, K. Hatano, , and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of the Thirteenth Intl. World Wide Web Conf.*, 2004.

[7] http://www.google.com

[8] http://www.google.com/psearch