

Adaptive Web Search Based on User Profile Constructed without Any Effort from Users

Kazunari Sugiyama^{*}
Nara Institute of Science and
Technology
8916-5 Takayama, Ikoma,
Nara 630-0192, Japan
kazuna-s@is.naist.jp

Kenji Hatano
Nara Institute of Science and
Technology
8916-5 Takayama, Ikoma,
Nara 630-0192, Japan
hatano@is.naist.jp

Masatoshi Yoshikawa
Nagoya University
Furo, Chikusa, Nagoya,
Aichi 464-8601, Japan
yosikawa@itc.nagoya-
u.ac.jp

ABSTRACT

Web search engines help users find useful information on the World Wide Web (WWW). However, when the same query is submitted by different users, typical search engines return the same result regardless of who submitted the query. Generally, each user has different information needs for his/her query. Therefore, the search results should be adapted to users with different information needs. In this paper, we first propose several approaches to adapting search results according to each user's need for relevant information without any user effort, and then verify the effectiveness of our proposed approaches. Experimental results show that search systems that adapt to each user's preferences can be achieved by constructing user profiles based on modified collaborative filtering with detailed analysis of user's browsing history in one day.

Categories and Subject Descriptors

H.1.2 [User/Machine Systems]: Human factors; H.3.3 [Information Search and Retrieval]: Information filtering, Relevance feedback, Search process; H.5.4 [Hypertext/Hypermedia]: Navigation, User issues

General Terms

Algorithms, Experimentation, Human factors, Performance

Keywords

WWW, Information retrieval, User modeling

1. INTRODUCTION

It has become increasingly difficult for users to find information on the WWW that satisfies their individual needs since information resources on the WWW continue to grow. Under these circumstances, Web search engines help users find useful information on the WWW. However, when the same query is submitted by different users, most search engines return the same results regardless of who submits the query. In general, each user has different information needs for his/her query. For example, for the query "Java," some users may be interested in documents dealing with the programming language, "Java," while other users may want documents

^{*}He is currently working for HITACHI, Ltd., Software Division.

related to "coffee." Therefore, Web search results should adapt to users with different information needs. In order to predict such information needs, there are several approaches applying data mining techniques to extract usage patterns from Web logs [40, 8, 11, 47]. However, the discovery of patterns from usage data by itself is not sufficient for performing the personalization tasks. Furthermore, Shahabi and Chen [37] have pointed out that the item association generated from Web server logs might be wrong because Web usage data from the server side are not reliable. Therefore, these techniques are not so appropriate for Web personalization. Another novel information systems designed to realize such adaptive systems have been proposed that personalize information or provide more relevant information for users. As far as we know, three types of Web search systems provide such information: (1) systems using relevance feedback [10], (2) systems in which users register their interest or demographic information, and (3) systems that recommend information based on users' ratings. In these systems, users have to register personal information such as their interests, age, and so on, beforehand, or users have to provide feedback on relevant or irrelevant judgements, ratings on a scale from 1 (very bad) to 5 (very good), and so on. These types of registration, feedback, or ratings can become time consuming and users prefer easier methods. Therefore, in this paper, we propose several approaches that can be used to adapt search results according to each user's information need. We then compare the retrieval accuracy of our proposed approaches. Compared with our prior works [41, 44], we scrutinize user's browsing history in one day closely and it allows each user to perform more fine-grained search by capturing changes of each user's preferences without any user effort. Such a method is not performed in typical search engines.

This paper is organized as follows: In Section 2, we review related work focusing on personalized search systems. In Section 3, we propose novel approaches to providing relevant information that satisfies each user's information need by capturing changes in user's preferences without user's effort. In Section 4, we present the experimental results for evaluating our proposed approaches. Finally, we conclude the paper with a summary and directions for future work in Section 5.

2. RELATED WORK

As described in Section 1, there are several types of search systems that provide users with information more relevant to their individual needs. For example, we review hyperlink-based personalized Web search, personalized Web sites, and recommender systems.

2.1 Hyperlink-Based Personalized Web Search

The field of Web information retrieval focuses on hyperlink structures of the Web, for example with Web search engines such as Google¹ [6] and the CLEVER project [18]. To address several problems with these engines, i.e., (1) the weight for a Web page is merely defined, and (2) the relativity of contents among hyperlinked Web pages is not considered, we proposed several approaches to refining the TF-IDF scheme for Web pages using their hyperlinked neighboring pages [42, 43]. In personalized Web searches, the hyperlink structures of the Web are also becoming important. The use of personalized PageRank to enable personalized Web searches was first proposed in [32], where it was suggested as a modification of the global PageRank algorithm, which computes a universal notion of importance of a Web page. The computation of (personalized) PageRank scores was not addressed beyond the original algorithm. Haveliwala [13] used personalized PageRank scores to enable “topic sensitive” Web searches. Experiments in this work concluded that the use of personalized PageRank scores can improve a Web search. However, no experiments based on a user’s context such as browsing patterns, bookmarks, and so on were conducted. Therefore, it is not clear if search results obtained using this approach actually satisfy information needs that is different user by user. In addition, the number of hub vectors used was limited to 16 due to the computational requirements. In order to address this problem, Jeh and Widom [20] proposed an approach that can scale well with the large size of hub vectors to realize personalized Web searches. On the other hand, Chang et al. [7] proposed algorithms for creating “personally customized *authority* documents” to correspond more closely to the user’s internal model following the conventions of Kleinberg’s HITS algorithm [23].

2.2 Personalized Web Sites

Link topology and the structure and contents of Web pages are often used in the construction of a personalized Web site. In this section, we review the framework of these systems with regard to “Link Personalization,” and “Content Personalization.”

2.2.1 Link Personalization

This scheme involves selecting the links that are more relevant to the user and changing the original navigation space by reducing or improving the relationships between Web pages. E-commerce applications use link personalization to recommend items based on the buying history of clients or some categorization of clients based on ratings and opinions. Users who give similar ratings to similar objects are presumed to have similar preferences, so when a user seeks recommendations about a certain product, the site suggests those recommendations that are most popular for his/her class or those that best correlate with the given product for that class. At the E-commerce site for Amazon.com², this approach has been taken to an extreme by constructing a “New for you” home page and presenting it to each user, with new products that the user may be interested in. Additionally, Amazon.com uses implicit recommendations via purchase history and/or explicit recommendations via “rate it” features to generate recommendations of products to purchase. In a recent study, Tsandilas and Schraefel [46] proposed a system that automatically adapts links in the browsed pages based on their relevance to the weighted topics specified by sliders that users can manipulate.

¹<http://www.google.com/>

²<http://www.amazon.com/>

2.2.2 Content Personalization

In general, content personalization is done when pages present different information to different users. The difference between this and “Link Personalization” described in Section 2.2.1 is subtle because part of the contents (i.e., the link anchors) presents different information when links are personalized. However, content personalization is referred to when substantial information in a Web page is personalized, unlike link anchors. For example, Bharat et al. [4] presented “*Krakatoa Chronicle*”, an interactive personalized newspaper on the WWW that allows for interactive personalization, browsing and layout control. Moreover, My Yahoo!³ [28] or My Netscape⁴ filters the information that is relevant to the user, showing only sections and details in which the user may be interested. The user may explicitly indicate his/her preferences, or preferences may be inferred (semi-) automatically from his/her profile or from his/her navigation activity. At these sites, users choose a set of “modules” from a large set including weather, news, music and so on, and further personalize these modules by choosing a set of attributes of the module to be perceived. The approach followed in these applications is that the users should be able to “construct” their own pages and even the layout may be customized. However, users have to input their preferences or demographic information based on the prior questionnaire. Systems related to personalization on the Web seems to be mainly based on text retrieval. However, personalized systems in the field of multimedia are also being developed [16, 30, 25, 39, 9, 17]. These systems also require explicit users’ inputs to obtain relevant information. In summary, the aforementioned systems have two problems: (1) the users’ loads become high because these systems heavily rely on the users’ inputs; (2) these sites cannot adapt to the changes in users’ preferences unless the users change their previously registered preferences by themselves.

2.3 Recommender Systems

It has become increasingly difficult to search for useful information on the Web because the amount of information on the Web continues to grow. Therefore, we get the feeling of being overwhelmed by the number of choices. This situation is often referred to as “information overload.” As one of the most promising approaches to alleviate this overload, recommender systems have emerged in domains such as E-commerce, digital libraries, and knowledge management. These systems provide personalized suggestions based on user preferences. Recommender systems collect user feedback in the form of ratings for items in a given domain and exploit similarities and differences among profiles of several users in determining how to recommend an item. There are two prevalent approaches to constructing recommender systems – collaborative filtering-based and content-based recommendation.

2.3.1 Collaborative Filtering-Based Recommendation

Collaborative filtering-based recommendation is the most successful recommendation technique to date. The term collaborative filtering was coined by Goldberg et al. [12]. Collaborative filtering means that people collaborate to help one another perform filtering by recording their reactions to documents they read. Based on this concept, Goldberg et al. developed a system called Tapestry that is one of the earliest implementations of collaborative filtering-based recommendation. This system is used to filter email and it allows users to annotate messages. The collaborative filtering provided by Tapestry was not automated, and users were required to formulate

³<http://www.my.yahoo.com/>

⁴<http://my.netscape.com/>

complex queries in a special query language designed for the task. In addition, this system relied on explicit opinions of people from a close-knit community, such as a group of office workers. However, recommender systems for large communities generally cannot depend on everyone knowing each other. Therefore, the framework in Tapestry is not appropriate to systems for large communities.

Rating-based automated collaborative filtering is quickly becoming a popular approach to reducing information overload by providing personalized recommendations for information, products or services. For example, the k -nearest neighbor collaborative filtering-based systems are achieving widespread success on the Web. The GroupLens research system [33, 24], which filters Usenet news, first introduced an automated collaborative filtering system using the k -nearest neighbor-based algorithm. In this algorithm, a subset of appropriate k users is chosen based on their similarity to the active user, and a weighted aggregate of their rating is used to generate predictions for the active user. GroupLens then recommends Usenet news articles to these active users.

While the Tapestry and GroupLens mentioned above rely on explicit ratings, some systems rely on implicit ratings. For example, Morita and Shinoda [31] exploit “time-spent-reading” as a measure of implicit ratings. PHOAKS (People Helping One Another Know Stuff) [45] also uses implicit ratings to construct a recommender system by examining Usenet news postings to find “endorsements” of Web sites. It then creates a listing of the top Web sites endorsed in each newsgroup. Some recommender systems also explore user preferences transparently without any extra effort from the users like the recommender systems relying on implicit ratings described above. For example, Letizia [26, 27] and WebWatcher [21] infer user preferences by observing user-browsing behavior. However, the main shortcomings in Letizia and WebWatcher are that they maintain persistent and slowly-changing user models and overlook the fact that different browsing sessions by the same user or even a single session may involve different user interests and goals. Moreover, Kelly and Teevan [22] have published a nice summary with regard to the systems using implicit measures.

In addition, at the E-commerce sites such as Amazon.com, CD-now.com and MovieFinder.com, automated collaborative filtering systems have been used with considerable success. Moreover, in the field of audio, Ringo [38] uses collaborative filtering techniques to provide users with recommendations for music albums and artists.

2.3.2 Content-Based Recommendation

A content-based approach provides recommendations by comparing representations of content contained in an item with representations of content that the user is interested in. In this approach, a model of user ratings is first developed. Algorithms in this category use probabilities and envision the collaborative filtering process by computing the expected value of a user prediction given the user’s ratings on other items. The model building process is performed by three different machine learning algorithms: (1) Bayesian network [5], (2) clustering [3, 5], and (3) rule-based models [35].

The systems described in Section 2.3.1 only provide recommendations based on collaborative filtering. However, some systems provide better recommendations by combining collaborative filtering with content information. Fab [2] uses relevance feedback to simultaneously construct a personal filter along with a communal “topic” filter. Web pages are initially ranked by the topic filter and then sent to user’s personal filters. The user then provides relevance feedback for that Web page, and this feedback is used to modify both the personal filter and the originating topic filter. Basu et al. [3] integrate content and collaboration in a framework where

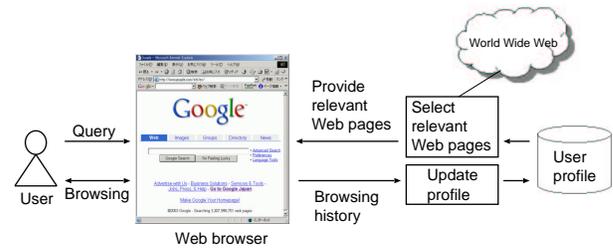


Figure 1: System overview.

they treat recommendation as a classification task. Melville et al. [29] overcome drawbacks of collaborative filtering systems in their recommender system by exploiting content information of items already rated. In recent study on recommender systems, Schafer et al. [36] introduce a new class of recommender system that provides users with personalized control over the generation of a single recommendation list formed from a combination of rich data using multiple information resources and recommendation techniques.

3. OUR PROPOSED METHOD

As we described in Section 2.1, hyperlink-based personalized search systems have a problem in that they do not clarify whether their search results actually satisfy each user’s information need. This is because personalization based on a user’s context, i.e., browsing patterns, bookmarks, and so on is not performed. The personalized Web sites described in Section 2.2 have the following shortcomings: (1) users have to rate items or adjust sliders to obtain relevant information in “Link Personalization” described in Section 2.2.1; and (2) in “Content Personalization” described in Section 2.2.2, the load on users becomes high because they have to answer questionnaires in advance to register their personal preferences or demographic information, and they have to change their registered information by themselves if their interests change. In addition, the recommender systems described in Section 2.3 have the potential to provide serendipitous recommendations if users are only willing to rate items. However, in actuality, most users are unwilling to rate items even though user’s ratings for items are key factors to achieving better recommendations. As a result, the accuracy of recommendations may be poor.

We do not necessarily believe that approaches based on user ratings provide users with more relevant information that satisfies each user’s information need. Therefore, search system should directly and exactly capture the changes in each user’s preferences without any user effort in order to provide more relevant information for each user. In order to construct such a system, we propose several approaches to adapting search results according to each user’s information need. Unlike the research studies described in the previous section, our approach is novel because it allows each user to perform a fine-grained search by capturing the changes in each user’s preferences without any user effort.

Figure 1 shows an overview of our system. When a user submits a query to a search engine through a Web browser, the search engine returns search results corresponding to the query. Based on the search results, the user may select a Web page in an attempt to satisfy his/her information need. In addition, the user may access more Web pages by following the hyperlinks on his/her selected Web page and continue to browse. Our system monitors the user’s browsing history and updates his/her profile whenever his/her browsing page changes. When the user submits a query the next time, the search results adapt based on his/her user profile.

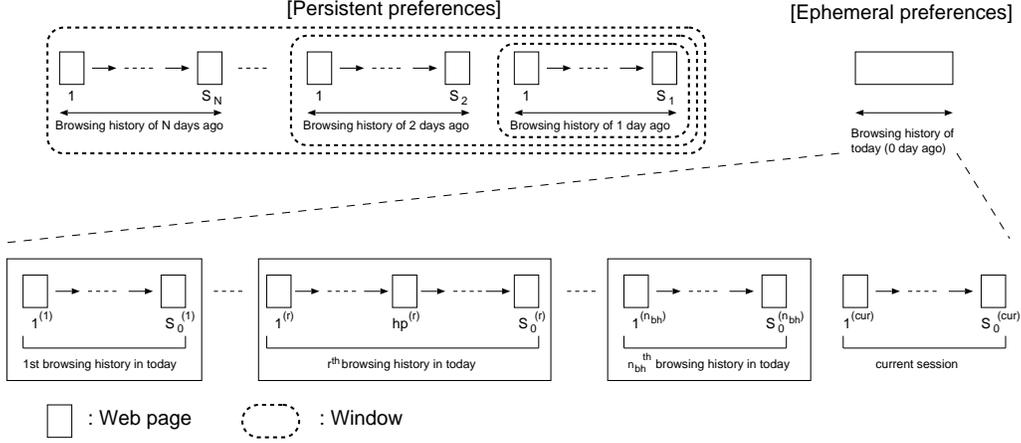


Figure 2: User's browsing history in today and N days before today.

In the following sections, we explain how to construct a user profile in the “update profile” component illustrated in Figure 1. In our approach, the user profile is constructed implicitly. In other words, a user does not need to perform explicit efforts such as feedback, ratings and so on in order to construct his/her profile. We construct each user profile based on the following two methods: (1) Pure browsing history, and (2) Modified collaborative filtering.

3.1 User Profile Construction Based on Pure Browsing History

In this method, we assume that the preferences of each user consist of the following two aspects: (1) persistent (or long term) preferences, and (2) ephemeral (or short term) preferences. In persistent preferences, the user profile is incrementally developed over time and it is stored for use in later sessions. The information exploited for constructing the profile usually comes from various sources, so it relies on different aspects of the user. On the other hand, in ephemeral preferences, the information used to construct each user profile is only gathered during the current session, and it is immediately exploited for executing some adaptive process aimed at personalizing the current interaction. In our prior works [41, 44], user's searches and browsing activities fall into one logical session. However, users usually do different tasks in one day and they may well do several searches and browsing activities in that time period. Therefore, it is necessary to analyze user's browsing behavior in one day in more detail. In our methods, we assume that user's preferences are constructed by accumulating his/her past preferences. Therefore, we construct each user profile \mathbf{P} considering both persistent preferences, \mathbf{P}^{per} , and ephemeral preferences, \mathbf{P}^{today} . \mathbf{P}^{per} shows a user profile constructed exploiting the user's browsing history of Web page from N days ago. Figure 2 illustrates the user's browsing history in today and N days before today. Here, we introduce the concept of window size in order to construct \mathbf{P}^{per} , and define S_j ($j = 0, 1, 2, \dots, N$) as the number of Web pages the user browsed on the j^{th} day. “ $j = 0$ ” means “today” as shown in Figure 2. In this figure, we consider that users perform n_{bh} different searches before the current session cur in today. In other words, the cur^{th} session, that is the newest session in today, is subsequent to the n_{bh}^{th} session. Therefore, the relation between n_{bh} and cur is defined by the following equation:

$$cur = n_{bh} + 1.$$

In each day, \mathbf{P}^{today} is constructed through the following process. At first, we denote the feature vector $\mathbf{w}^{hp^{(r)}}$ of browsed Web

page $hp^{(r)}$ ($hp = 1, 2, \dots, S_0$) in the r^{th} ($r = 1, 2, \dots, n_{bh}$) session as follows:

$$\mathbf{w}^{hp^{(r)}} = (w_{t_1}^{hp^{(r)}}, w_{t_2}^{hp^{(r)}}, \dots, w_{t_m}^{hp^{(r)}}),$$

where m is the number of distinct terms in the Web page $hp^{(r)}$, and t_k ($k = 1, 2, \dots, m$) denotes each term. Using the TF (term frequency) scheme, each element $w_{t_k}^{hp^{(r)}}$ of $\mathbf{w}^{hp^{(r)}}$ is defined as follows:

$$w_{t_k}^{hp^{(r)}} = c^{hp^{(r)}} \cdot \frac{tf(t_k, hp^{(r)})}{\sum_{s=1}^m tf(t_s, hp^{(r)})}, \quad (1)$$

where $tf(t_k, hp^{(r)})$ is the frequency of term t_k in each browsed Web page $hp^{(r)}$, and $c^{hp^{(r)}}$ is a constant that shows to what extent our system reflects the contents of the Web page on each user profile. We define constant $c^{hp^{(r)}}$ as follows:

$$c^{hp^{(r)}} = \begin{cases} 1; & dr \geq Th, \\ 0; & dr < Th, \end{cases} \quad (2)$$

where dr denotes the time spent reading normalized by the number of terms in Web page $hp^{(r)}$, and threshold Th is set to 0.317 based on our preliminary experiments. We then define partial user profile $\mathbf{P}^{(r)}$ at the r^{th} browsing history in today as follows:

$$\mathbf{P}^{(r)} = (p_{t_1}^{(r)}, p_{t_2}^{(r)}, \dots, p_{t_m}^{(r)}),$$

and define each element $p_{t_k}^{(r)}$ using Equation (1) as follows:

$$\begin{aligned} p_{t_k}^{(r)} &= \frac{1}{S_0^{(r)}} \sum_{hp=1}^{S_0} w_{t_k}^{hp^{(r)}} \\ &= \frac{1}{S_0^{(r)}} \sum_{hp=1}^{S_0} c^{hp^{(r)}} \cdot \frac{tf(t_k, hp^{(r)})}{\sum_{s=1}^m tf(t_s, hp^{(r)})}. \end{aligned} \quad (3)$$

Moreover, we also define user profile $\mathbf{P}^{(br)}$ obtained by browsing history up to the current session as follows:

$$\mathbf{P}^{(br)} = (p_{t_1}^{(br)}, p_{t_2}^{(br)}, \dots, p_{t_m}^{(br)}).$$

Using Equation (3), each element $p_{t_k}^{(br)}$ is also defined as follows:

$$\begin{aligned} p_{t_k}^{(br)} &= \sum_{r=1}^{n_{bh}} p_{t_k}^{(r)} \\ &= \sum_{r=1}^{n_{bh}} \frac{1}{S_0^{(r)}} \sum_{hp=1}^{S_0} c^{hp^{(r)}} \cdot \frac{tf(t_k, hp^{(r)})}{\sum_{s=1}^m tf(t_s, hp^{(r)})}. \end{aligned}$$

Similarly, we denote the feature vector $\mathbf{w}^{hp^{(cur)}}$ of browsed Web page $hp^{(cur)}$ ($hp = 1, 2, \dots, S_0$) in the current session as follows:

$$\mathbf{w}^{hp^{(cur)}} = (w_{t_1}^{hp^{(cur)}}, w_{t_2}^{hp^{(cur)}}, \dots, w_{t_m}^{hp^{(cur)}}),$$

where m is the number of distinct terms in the Web page $hp^{(cur)}$, and t_k ($k = 1, 2, \dots, m$) denotes each term. Using the TF (term frequency) scheme, each element $w_{t_k}^{hp^{(cur)}}$ of $\mathbf{w}^{hp^{(cur)}}$ is defined as follows:

$$w_{t_k}^{hp^{(cur)}} = c^{hp^{(cur)}} \cdot \frac{tf(t_k, hp^{(cur)})}{\sum_{s=1}^m tf(t_s, hp^{(cur)})}. \quad (4)$$

where $tf(t_k, hp^{(cur)})$ is the frequency of term t_k in each browsed Web page $hp^{(cur)}$, and $c^{hp^{(cur)}}$ is a constant that shows to what extent our system reflects the contents of the Web page on each user profile defined as well as Equation (2). Then, we define partial user profile $\mathbf{P}^{(cur)}$ obtained at the current session in today as follows:

$$\mathbf{P}^{(cur)} = (p_{t_1}^{(cur)}, p_{t_2}^{(cur)}, \dots, p_{t_m}^{(cur)}),$$

and define each element $p_{t_k}^{(cur)}$ using Equation (4) as follows:

$$\begin{aligned} p_{t_k}^{(cur)} &= \frac{1}{S_0^{(cur)}} \sum_{hp=1}^{S_0} w_{t_k}^{hp^{(cur)}} \\ &= \frac{1}{S_0^{(cur)}} \sum_{hp=1}^{S_0} c^{hp^{(cur)}} \cdot \frac{tf(t_k, hp^{(cur)})}{\sum_{s=1}^m tf(t_s, hp^{(cur)})}. \end{aligned}$$

Using $\mathbf{P}^{(br)}$ and $\mathbf{P}^{(cur)}$, \mathbf{P}^{today} is constructed as follows:

$$\mathbf{P}^{today} = x\mathbf{P}^{(br)} + y\mathbf{P}^{(cur)}, \quad (5)$$

where x and y are constants that satisfy $x + y = 1$. In order to emphasize the current session, we assign larger weight to y than x . In other words, y is larger than 0.5, and x is smaller than 0.5 under the condition, $x + y = 1$.

Additionally, we also construct user profile \mathbf{P}^{per} considering persistent preferences. In order to do that, we set the window size N ($N = 1, 2, \dots, 30$). The user profile \mathbf{P}^{per} is denoted as follows:

$$\mathbf{P}^{per} = (p_{t_1}^{per}, p_{t_2}^{per}, \dots, p_{t_m}^{per}), \quad (6)$$

and each element $p_{t_k}^{per}$ is defined as follows:

$$p_{t_k}^{per} = \frac{1}{S_N} \sum_{hp=1}^{S_N} w_{t_k}^{hp} \cdot e^{-\frac{\log 2}{hl}(d-d_{t_k, in it})}, \quad (7)$$

where $e^{-\frac{\log 2}{hl}(d-d_{t_k, in it})}$ is a forgetting factor under the assumption that user's preferences gradually decay as days pass. In this factor, $d_{t_k, in it}$ is the day when term t_k initially occurs, d is the number of days following to $d_{t_k, in it}$, and hl is a half-life span parameter. The half-life span hl is set to 7. In other words, we assume that user's preferences reduce by 1/2 in one week. We also assume that each user browsed S_N pages on each day. This value S_N is

		Item that prediction is computed					
		item 1	item 2	...	item i	...	item I
Active user	user 1	2	3		2		
	user 2	5			1		4
	...						
	user a		3				5
	...						
	user U	4	2		5		

Figure 3: User-item ratings matrix for collaborative filtering.

different user by user. Therefore, we normalize $p_{t_k}^{per}$ using S_N as shown in Equation (7). Using \mathbf{P}^{today} defined by Equation (5), and \mathbf{P}^{per} defined by Equation (6), we finally construct the user profile \mathbf{P} as follows:

$$\mathbf{P} = a\mathbf{P}^{per} + b\mathbf{P}^{today} \quad (8)$$

$$= a\mathbf{P}^{per} + bx\mathbf{P}^{(br)} + by\mathbf{P}^{(cur)}, \quad (9)$$

where a and b are constants that satisfy $a + b = 1$, and x and y are constants that satisfy $x + y = 1$ as described at Equation (5).

3.2 User Profile Construction Based on Modified Collaborative Filtering Algorithm

In this section, we first briefly review the pure collaborative filtering algorithms, especially neighborhood-based algorithms, and then describe how to construct user profiles using the modified collaborative filtering algorithms.

3.2.1 Overview of the Pure Collaborative Filtering Algorithm

Collaborative filtering can be represented as the problem of predicting missing values in a user-item ratings matrix. Figure 3 shows a simplified example of a user-item ratings matrix.

In the neighborhood-based algorithm [15], a subset of users is first chosen based on their similarity to the active user, and a weighted combination of their rating is then used to produce predictions for the active user. The algorithm we use can be summarized in the following steps:

1. Weight all users with respect to similarity to the active user. This similarity between users is measured as the Pearson correlation coefficient between their rating vectors.
2. Select n users that have the highest similarity with the active user. These users form the *neighborhood*.
3. Compute a prediction from a weighted combination of the neighbor's ratings.

In step 1, $S_{a,u}$, which denotes similarity between users a and u , is computed using the Pearson correlation coefficient defined below:

$$S_{a,u} = \frac{\sum_{i=1}^I (r_{a,i} - \bar{r}_a) \times (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^I (r_{a,i} - \bar{r}_a)^2 \times \sum_{i=1}^I (r_{u,i} - \bar{r}_u)^2}}, \quad (10)$$

where $r_{a,i}$ is the rating given to item i by user a , and \bar{r}_a is the mean rating given by user a , and I is the total number of items.

In step 2, i.e., neighborhood-based methods, a subset of appropriate users is chosen based on their similarity to the active user, and a weighted aggregate of their ratings is used to generate predictions for the active user in the next step 3.

In step 3, predictions are computed as the weighted average of deviations from the neighbor's mean:

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^n (r_{u,i} - \bar{r}_u) \times S_{a,u}}{\sum_{u=1}^n S_{a,u}},$$

		Term weight that prediction is computed					
		term 1	term 2	term i	term T
Active user	user 1	0.745	0.362				0.718
	user 2		0.835		0.534		0.126
	⋮						
	user a		0.639				0.485
	⋮						
	user U	0.247	0.461		0.928		

(a)

		Term weight that prediction is computed									
		term 1	term 2	term i	term T	term T+1	term T+2	term T+v
Active user	user 1	0.745	0.362				0.718		0.451		
	user 2		0.835		0.534		0.126	0.723			
	⋮										
	user a		0.639				0.485		0.328		0.563
	⋮										
	user U	0.247	0.461		0.928			0.686			0.172

(b)

Figure 4: User-term weights matrix for modified collaborative filtering [(a) when each user browsed k Web pages, (b) when each user browsed $k + 1$ Web pages].

where $p_{a,i}$ is the prediction for active user a for item i . $S_{a,u}$ is the similarity between users a and u as described at Equation (10), and n is the number of users in the neighborhood.

3.2.2 User Profile Construction using Modified Collaborative Filtering Algorithm

In the pure collaborative filtering algorithms described in Section 3.2.1, we considered a user-item ratings matrix. Similarly, in the construction of a user profile, we can consider a user-term weights matrix like that shown in Figure 4(a). In addition, based on the pure collaborative filtering algorithms described in Section 3.2.1, we can apply their predictive algorithms to predict a term weight in each user profile. In other words, since each user profile is computed based on term weights in a Web page the user browsed and the browsed pages are different according to each user, the profile is constructed in the form of a user-term weights matrix with missing values, as illustrated in Figure 4. This is very analogous to the user-item ratings matrix used in the pure collaborative filtering algorithms. Therefore, we expect that a more accurate user profile is constructed since these missing values are predicted using the algorithms in collaborative filtering. In this approach, we propose the following two methods: (1) user profile construction based on the static number of users in the neighborhood, and (2) user profile construction based on dynamic number of users in the neighborhood.

(1) User Profile Construction Based on the Static Number of Users in the Neighborhood

In this method, our proposed algorithms are explained in the following steps (note the similarity to the collaborative filtering algorithms described in Section 3.2.1):

1. Weight all users with respect to similarity to the active user. This similarity between users is measured as the Pearson correlation coefficient between their term weight vectors unlike the rating vectors described in Section 3.2.1.
2. Select n users that have the highest similarity to the active user. These users form the neighborhood.
3. Compute a prediction from a weighted combination of the neighbor's term weights

In step 1, $S_{a,u}$, which denotes similarity between users a and u , is computed using the Pearson correlation coefficient, defined below:

$$S_{a,u} = \frac{\sum_{i=1}^T (w_{a,i} - \bar{w}_a) \times (w_{u,i} - \bar{w}_u)}{\sqrt{\sum_{i=1}^T (w_{a,i} - \bar{w}_a)^2 \times \sum_{i=1}^T (w_{u,i} - \bar{w}_u)^2}}, \quad (11)$$

where $w_{a,i}$ is the weight of term i regarding user a computed based on term frequency in a browsed Web page defined by Equation (4),

and \bar{w}_a is the mean term weight regarding user a , and T is the total number of terms.

In step 2, i.e., neighborhood-based methods, a subset of appropriate users is chosen based on their similarity to the active user, and a weighted aggregate of their term weights is used to generate predictions for the active user in the coming step 3. In this step, the number of selected users is fixed to n for any user. That is why we call this method "static."

In step 3, predictions are computed as the weighted average of deviations from the neighbor's mean:

$$p_{a,i} = \bar{w}_a + \frac{\sum_{u=1}^n (w_{u,i} - \bar{w}_u) \times S_{a,u}}{\sum_{u=1}^n S_{a,u}},$$

where $p_{a,i}$ is the prediction for the active user a for weight of term i , $S_{a,u}$ is the similarity between users a and u as described at Equation (11), and n is the number of users in the neighborhood.

(2) User Profile Construction Based on Dynamic Number of Users in the Neighborhood

In this method, our proposed algorithms are explained in the following steps (note the similarity to the collaborative filtering algorithms described in Section 3.2.1, and aforementioned static approach):

1. Generate clusters of users by means of the k -Nearest Neighbor algorithms [19]. The similarity between user a and these clusters is measured as the Pearson correlation coefficient between their term weight vectors.
2. Select n clusters that have higher similarity to the active user than the threshold. We consider the centroid vectors of these selected clusters as the neighborhood of the active user.
3. Compute a prediction from a weighted combination of the term weights using centroid vectors of clusters.

In step 1, $S_{a,g}$, which denotes similarity between users a and centroid vectors of clusters g , is computed using the Pearson correlation coefficient, defined below:

$$S_{a,g} = \frac{\sum_{i=1}^T (w_{a,i} - \bar{w}_a) \times (w_{g,i} - \bar{w}_g)}{\sqrt{\sum_{i=1}^T (w_{a,i} - \bar{w}_a)^2 \times \sum_{i=1}^T (w_{g,i} - \bar{w}_g)^2}}, \quad (12)$$

where $w_{a,i}$ is the weight of term i regarding user a computed based on term frequency in a browsed Web page defined by Equation (4), and \bar{w}_a is the mean term weight regarding user a , and T is the total number of terms.

In step 2, several clusters are chosen based on their similarity to the active user, and a weighted aggregate of their term weights is used to generate predictions for the active user in the next step 3. In this step, the number of selected clusters is different user by user. That is why we call this method "dynamic." Therefore, it is

expected that this method allows each user to perform more fine-grained search.

In step 3, predictions are computed as the weighted average of deviations from the neighbor's mean:

$$p_{a,i} = \bar{w}_a + \frac{\sum_{g=1}^n (w_{g,i} - \bar{w}_g) \times S_{a,g}}{\sum_{g=1}^n S_{a,g}},$$

where $p_{a,i}$ is the prediction for the active user a for term weights i , $S_{a,g}$ is the similarity between users a and centroid vectors of clusters g as described at Equation (12), and n is the number of centroid vectors of clusters in the neighborhood.

4. EXPERIMENTS

4.1 Experimental Setup

We conducted experiments in order to verify the effectiveness of the three approaches: (1) relevance feedback and implicit approaches, (2) user profiles based on pure browsing history as described in Section 3.1, and (3) user profiles based on the modified collaborative filtering algorithm described in Section 3.2. While users have to provide feedback explicitly in relevance feedback, users do not have to provide any effort in our proposed methods (2) and (3) since our system implicitly captures changes in user's preference. The experiments were implemented using Perl on a workstation (CPU: UltraSparc-II 480MHz \times 4, Memory: 2GBytes, OS: Solaris8). We used 50 query topics that were employed as test topics in the TREC WT10g test collection [14]. Note that we only used query topics of the test collection, and did not use the contents of the test collection. In summary, we change the each of 50 topic descriptions into query keywords that the subjects can easily submit to the search engine Google. After the subjects submit these queries to Google, our system reorders the search results according to each user's profile constructed by the methods described in Section 3. In our experiments, we observed the browsing history of 20 subjects for 30 days. The subjects browsed 12 Web pages in one day on average. In addition, the number of terms in user profiles accumulated during the 30 days is about 810,000. In the following, let the h^{th} Web page in the search results and the user profile as defined by Equation (9) be rp_h and \mathbf{P} , respectively. Then, the feature vector of the h^{th} Web page rp_h in the search results, \mathbf{w}^{rp_h} , is defined as follows:

$$\mathbf{w}^{rp_h} = (w_{t_1}^{rp_h}, w_{t_2}^{rp_h}, \dots, w_{t_m}^{rp_h}),$$

where m is the number of distinct terms in the Web page rp_h , and t_k ($k = 1, 2, \dots, m$) denotes each term. We also define each element $w_{t_k}^{rp_h}$ of \mathbf{w}^{rp_h} based on the TF (term frequency) scheme as follows:

$$w_{t_k}^{rp_h} = \frac{tf(t_k, rp_h)}{\sum_{s=1}^m tf(t_s, rp_h)},$$

where $tf(t_k, rp_h)$ is the frequency of term t_k in the rp_h . The similarity $sim(\mathbf{P}, \mathbf{w}^{rp_h})$ between the user profile \mathbf{P} and the feature vector of the h^{th} Web page in search results \mathbf{w}^{rp_h} is computed by the following Equation.

$$sim(\mathbf{P}, \mathbf{w}^{rp_h}) = \frac{\mathbf{P} \cdot \mathbf{w}^{rp_h}}{|\mathbf{P}| \cdot |\mathbf{w}^{rp_h}|}. \quad (13)$$

Based on the value obtained by Equation (13), the search results are adapted to each user according to his/her profile. These results are compared with the search results of Google [6]. We then evaluate the retrieval accuracy using R -precision [1]. We employed 30 as the value of R because users tend to take a look at the first 30 documents retrieved.

4.2 Experimental Results

4.2.1 User Profile Based on Relevance Feedback

Relevance feedback [34] is the most popular query reformulation strategy. In a relevance feedback process, the user is presented with a list of the retrieved documents and marks those that are relevant after examining them. The basic idea is to reformulate the original query vector \mathbf{Q}^{org} into new query vector \mathbf{Q}^{new} such that it gets closer to the term-weight vector space of the relevant documents. In our experiments, we use the Rocchio formulation defined as follows:

$$\mathbf{Q}^{new} = \alpha \mathbf{Q}^{org} + \frac{\beta}{|D_r|} \sum_{\mathbf{d}_j \in D_r} \mathbf{d}_j - \frac{\gamma}{|D_n|} \sum_{\mathbf{d}_j \in D_n} \mathbf{d}_j,$$

where D_r and D_n are the set of relevant and non-relevant documents as identified by the user among the retrieved documents, respectively, and $|D_r|$ and $|D_n|$ are the number of documents in the sets D_r and D_n , respectively. We set α , β and γ that are tuning constants to 1, 1 and 1, respectively. In other words, the subjects provide both positive and negative feedbacks.

We believe that the new query vector \mathbf{Q}^{new} obtained by the user's judgement, whether the retrieved documents are relevant or not, reflects the user's preferences. Therefore, we treat \mathbf{Q}^{new} as $\mathbf{P}^{(cur)}$ defined by Equation (9), and employ \mathbf{Q}^{new} as an initial preference of a user to construct a user profile. In this case, using Equation (9), the user profile \mathbf{P} is defined as follows:

$$\mathbf{P} = a\mathbf{P}^{per} + bx\mathbf{P}^{(br)} + by\mathbf{Q}^{new}. \quad (14)$$

We asked each subject to judge if the top 30 search results returned by Google according to the query keywords are relevant or not, and constructed user profile \mathbf{P} based on Equation (14). In this experiment, we varied the number of feedbacks FB that each subject provided from 1 to 3. Figures 5 to 7 show the R -precision when the values of a and b are varied such that these values satisfy $a + b = 1$ under the condition that the numbers of feedbacks for the top 30 search results are 1, 2, and 3.

4.2.2 User Profile Based on Pure Browsing History

In this approach, each user profile is constructed as mentioned in Section 3.1. The user profile \mathbf{P} is defined as follows:

$$\mathbf{P} = a\mathbf{P}^{per} + bx\mathbf{P}^{(br)} + by\mathbf{P}^{(cur)}.$$

Figure 8 shows the R -precision when the values of a and b are varied such that these values satisfy $a + b = 1$.

4.2.3 User Profile Based on Modified Collaborative Filtering

In this approach, when the user browses a new Web page, new terms are added to his/her user profile. However, other users do not always browse the same pages, so missing values occur in the user-term weights matrix as illustrated in Figure 4. These missing values are predicted using the algorithms described in Section 3.2.2, and then the matrix is filled. We consider that this user-term vector reflects the user's preferences. Let this user-term vector with predicted value be \mathbf{V}^{pre} . We treat \mathbf{V}^{pre} as $\mathbf{P}^{(cur)}$ defined by Equation (9), and employ \mathbf{V}^{pre} as an initial preference of a user to construct a user profile. In this case, using Equation (9), the user profile \mathbf{P} is defined as follows:

$$\mathbf{P} = a\mathbf{P}^{per} + bx\mathbf{P}^{(br)} + by\mathbf{V}^{pre}. \quad (15)$$

Figures 9 to 12 show the R -precision of static approaches when the values of a and b are varied such that these values satisfy $a + b = 1$

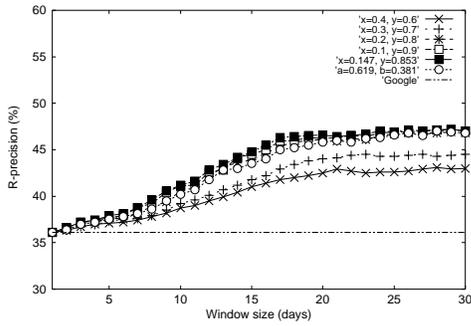


Figure 5: R -precision obtained using relevance feedback-based user profile ($FB = 1$).

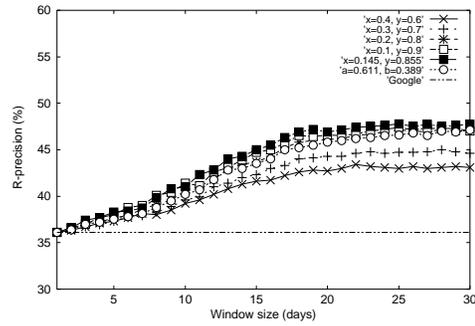


Figure 7: R -precision obtained using relevance feedback-based user profile ($FB = 3$).

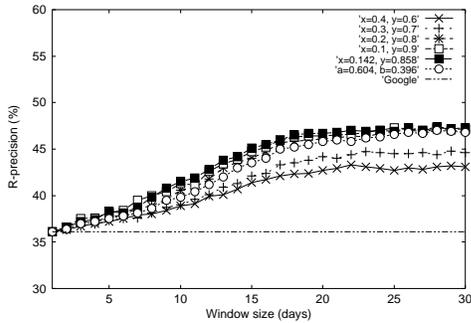


Figure 6: R -precision obtained using relevance feedback-based user profile ($FB = 2$).

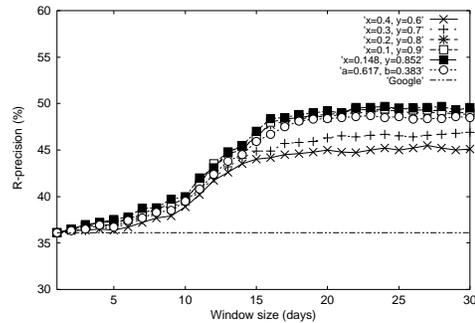


Figure 8: R -precision obtained using pure browsing history-based user profile.

under the condition that the numbers of neighbors n are 5, 10, 15, and 20, respectively. In addition, Figure 13 shows the R -precision of dynamic approaches.

4.3 Discussion

In this section, we discuss the results obtained using each approach discussed in Section 4.2. Note that, in Figures 5 to 13, the R -precision of Google is constant because it does not depend on the window size. In these figures, we compare the following two cases: (1) where user's browsing activities fall into one logical session [44]; and (2) where user's browsing activities in one day are analyzed in more detail. The precisions in the former case are obtained when each user profile is constructed by Equation (8) based on the values of a and b . On the other hand, the precisions in the latter case are obtained when each user profile is constructed by Equation (9) based on the values of a and b that bring the best result in the former case, and the values of x and y .

In the relevance feedback-based user profile shown in Figures 5 to 7, we found that a user profile that provides search results adaptive to a user can be constructed when a window size with about 18 days is used regardless of the number of feedbacks. As mentioned in Section 4.2.1, we used query vector reformulated by relevance feedback as an initial preference of a user. However, we could not observed significant improvement in precision even if the number of feedbacks increases. We consider that this effect is caused because the initial preference of a user is absorbed by persistent preferences constructed using the window size. In addition, it is valid that we conducted experiments by examining the number of feedbacks from 1 to 3 since the precision is not improved largely in this range.

In the user profile based on pure browsing history shown in Figure 8, we found that a user profile that provides search results adap-

tive to a user can be constructed when a window size with about 15 days is used. This approach can achieve about 3% higher precision than the relevance feedback-based user profile, and the result shows that the user's browsing history strongly reflects the user's preference.

In addition, in the user profile based on modified collaborative filtering shown in Figures 9 to 13, we found that a user profile that provides search results adaptive to a user can be constructed when a window size with about 10 days is utilized. In user profile construction based on the static number of users in the neighborhood described in Section 3.2.2(1), the best precision is obtained in the case of $n = 5$ as illustrated in Figure 9; in other words, the 5 nearest neighbors of each user are taken. Therefore, as shown in Figures 9 to 12, we found that it is not so effective to adapt search results to each user even if more nearest neighbors are used. In addition, the user preferences of not only a certain user but also other users are exploited in this approach. We consider that this method obtained higher precision than the aforementioned approaches. In user profile construction based on the dynamic number of users in the neighborhood described in Section 3.2.2(2), we could obtain the best precision in all of our experimental results in the case of $x = 0.129$ and $y = 0.871$ in Equation (15) as shown in Figure 13. In this method, the neighborhood of each user is determined by the centroid vectors of clusters of users, and the number of the clusters is different user by user. Therefore, we believe that this method allows each user to perform more fine-grained search compared with static method.

The best precision of any of the methods is obtained when x is smaller than 0.2 and y is larger than 0.8 under the condition that $x + y = 1$ as described by Equation (5). This shows that search results that adapt to each user can be provided by focusing on the current session than the browsing history in today. Moreover, when

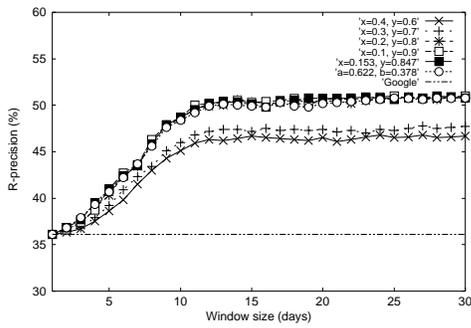


Figure 9: R -precision obtained using modified collaborative filtering-based user profile (static, $n = 5$).

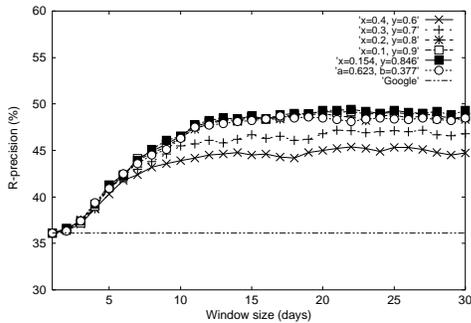


Figure 10: R -precision obtained using modified collaborative filtering-based user profile (static, $n = 10$).

the window size is small in this case, the large fluctuation in precision is observed. Therefore, we found that it is necessary to use a little larger window size in order to construct the user profile that appropriately captured user's persistent and ephemeral preferences. Furthermore, the precision obtained by each of our proposed methods can outperform the precision obtained by Google as shown in Figures 5 to 13. We believe that our proposed methods can perform a fine-grained search for each user that typical search engines can not perform.

5. CONCLUSION

In this paper, in order to provide each user with more relevant information, we proposed several approaches to adapting search results according to each user's information need. Our approach is novel in that it allows each user to perform a fine-grained search, which is not performed in typical search engines, by capturing changes in each user's preferences. We conducted experiments in order to verify the effectiveness of the approaches: (1) relevance feedback and implicit approaches, (2) user profiles based on pure browsing history, and (3) user profiles based on the modified collaborative filtering. We evaluated the retrieval accuracy of these approaches. The user profile constructed based on modified collaborative filtering achieved the best accuracy. This approach allows us to construct a more appropriate user profile and perform a fine-grained search that is better adapted to each user's preferences. In the future, if broadband networks spread widely, information is expected to be provided in a variety of forms such as music, movies and so on. In addition, more information will be provided for mobile terminals such as cellular phones, PDAs, or terminals in cars for Intelligent Transportation Systems (ITS). We believe that the technique proposed in this paper can be applied to situations where users require more relevant information to satisfy their informa-

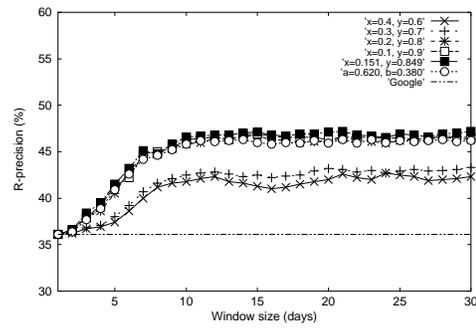


Figure 11: R -precision obtained using modified collaborative filtering-based user profile (static, $n = 15$).

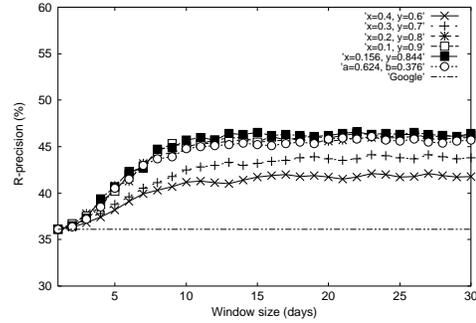


Figure 12: R -precision obtained using modified collaborative filtering-based user profile (static, $n = 20$).

tion needs. In future work, we plan to conduct experiments with a greater number of subjects and attempt to improve our proposed approaches by using a longer term of the user's browsing history in order to achieve much more adaptive search for each user.

6. REFERENCES

- [1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, 1999.
- [2] M. Balabanovic and Y. Shoham. Fab: Content-Based, Collaborative Recommendation. *Communications of the ACM*, 40(3):66-72, 1997.
- [3] C. Basu, H. Hirsh, and W. Cohen. Recommendation as Classification: Using Social and Content-Based Information in Recommendation. In *Proc. of the 15th National Conference on Artificial Intelligence (AAAI '98)*, pages 714-720, 1998.
- [4] K. Bharat, T. Kamba, and M. Albers. Personalized Interactive News on the Web. *Multimedia Systems*, 6(5):349-358, 1998.
- [5] J. S. Breese, D. Heckerman, and C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proc. of the 14th Conference on Uncertainty in Artificial Intelligence (UAI '98)*, pages 43-52, 1998.
- [6] S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Proc. of the 7th International World Wide Web Conference (WWW7)*, pages 107-117, 1998.
- [7] H. Chang, D. Cohn, and A. K. McCallum. Learning to Create Customized Authority Lists. In *Proc. of the 17th International Conference on Machine Learning (ICML 2000)*, pages 49-54, 2002.
- [8] R. Cooley, B. Mobasher, and J. Srivastava. Data Preparation for Mining World Wide Web Browsing Patterns. *Knowledge and Information Systems*, 1(1):5-32, 1999.
- [9] A. Field, P. Hartel, and W. Moosj. Personal DJ, an Architecture for Personalised Content Delivery. In *Proc. of the 10th International World Wide Web Conference (WWW10)*, pages 1-8, 2001.
- [10] L. Fitzpatrick and M. Dent. Automatic Feedback Using Past Queries: Social Searching? In *Proc. of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '97)*, pages 306-313, 1997.

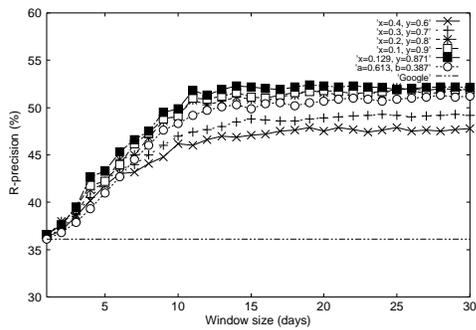


Figure 13: R-precision obtained using modified collaborative filtering-based user profile (dynamic).

- [11] X. Fu, J. Budzik, and K. J. Hammond. Mining Navigation History for Recommendation. In *Proc. of the 5th International Conference on Intelligent User Interfaces (IUI 2000)*, pages 106–112, 2000.
- [12] D. Goldberg, D. Nichols, B. M. Oki, and D. B. Terry. Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM*, 35(12):61–70, 1992.
- [13] T. H. Haveliwala. Topic-Sensitive PageRank. In *Proc. of the 11th International World Wide Web Conference (WWW2002)*, pages 517–526, 2002.
- [14] D. Hawking. Overview of the TREC-9 Web Track. *NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC-9)*, pages 87–102, 2001.
- [15] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An Algorithmic Framework for Performing Collaborative Filtering. In *Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99)*, pages 230–237, 1999.
- [16] W. Hill, L. Stead, M. Rosenstein, and G. W. Furnas. Recommending and Evaluating Choice in a Virtual Community of Use. In *Proc. of the Conference on Human Factors in Computing Systems (CHI '95)*, pages 194–201, 1995.
- [17] R. Hjelsvold, S. Vdaygiri, and Y. Léauté. Web-based Personalization and Management of Interactive Video. In *Proc. of the 10th International World Wide Web Conference (WWW10)*, pages 129–139, 2001.
- [18] IBM Almaden Research Center. Clever Searching. <http://www.almaden.ibm.com/cs/k53/clever.html>.
- [19] R. A. Jarvis and E. A. Patrick. Clustering Using a Similarity Measure Based on Shared Near Neighbors. *IEEE Transactions on Computers*, C22(11):1025–1034, 1973.
- [20] G. Jeh and J. Widom. Scaling Personalized Web Search. In *Proc. of the 12th International World Wide Web Conference (WWW2003)*, pages 271–279, 2003.
- [21] T. Joachims, D. Freitag, and T. M. Mitchell. WebWatcher: A Tour Guide for the World Wide Web. In *Proc. of the 15th International Joint Conference on Artificial Intelligence (IJCAI'97)*, pages 770–777, 1997.
- [22] D. Kelly and J. Teevan. Implicit Feedback for Inferring User Preference: A Bibliography. *SIGIR Forum*, 37(2):18–28, 2003.
- [23] J. M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 46(5):pages 604–632, 1999.
- [24] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. GroupLens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM*, 40(3):77–87, 1997.
- [25] V. Krishnan and S. G. Chang. Customized Internet Radio. In *Proc. of the 9th International World Wide Web Conference (WWW9)*, pages 609–618, 2000.
- [26] H. Lieberman. Letizia: An Agent That Assists Web Browsing. In *Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI '95)*, pages 924–929, 1995.
- [27] H. Lieberman. Autonomous Interface Agents. In *Proc. of the Conference on Human Factors in Computing Systems (CHI '97)*, pages 67–74, 1997.
- [28] U. Manber, A. Patel, and J. Robison. Experience with Personalization on Yahoo! *Communications of the ACM*, 43(8):35–39, 2000.
- [29] P. Melville, R. J. Mooney, and R. Nagarajan. Content-Boosted Collaborative Filtering for Improved Recommendations. In *Proc. of the 18th National Conference on Artificial Intelligence (AAAI2002)*, pages 187–192, 2002.
- [30] B. Meriardo, K. T. Lee, D. Luparello, and J. Roudaire. Automatic Construction of Personalized TV News Programs. In *Proc. of the 7th ACM International Conference on Multimedia (Multimedia '99)*, pages 323–331, 1999.
- [31] M. Morita and Y. Shinoda. Information Filtering Based on User Behavior Analysis and Best Match Text Retrieval. In *Proc. of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '94)*, pages 272–281, 1994.
- [32] L. Page. The PageRank Citation Ranking: Bringing Order to the Web. <http://google.stanford.edu/backrub/pageranksub.ps>, 1998.
- [33] P. Resnick, N. Iacovou, M. Suchak, and J. R. P. Bergstrom. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proc. of the ACM 1994 Conference on Computer Supported Cooperative Work (CSCW '94)*, pages 175–186, 1994.
- [34] J. Rocchio. Relevance Feedback in Information Retrieval. In G. Salton, editor, *The Smart Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice-Hall, Englewood Cliffs, NJ, 1971.
- [35] B. M. Sarwar, G. Karypis, and J. A. Konstan. Analysis of Recommendation Algorithms for E-commerce. In *Proc. of the 2nd ACM Conference on Electronic Commerce (EC '00)*, pages 158–167, 2000.
- [36] J. B. Schafer, J. A. Konstan, and J. Riedl. Meta-recommendation Systems: User-controlled Integration of Diverse Recommendations. In *Proc. of the 11th International Conference on Information and Knowledge Management (CIKM '02)*, pages 43–51, 2002.
- [37] C. Shahabi and Y.-S. Chen. An Adaptive Recommendation System without Explicit Acquisition of User Relevance Feedback. *Distributed and Parallel Databases*, 14(3):173–192, 2003.
- [38] U. Shardanand and P. Maes. Social Information Filtering: Algorithms for Automating “Word of Mouth”. In *Proc. of the Conference on Human Factors in Computing Systems (CHI '95)*, pages 210–217, 1995.
- [39] B. Smyth and P. Cotter. A Personalized Television Listings Service. *Communications of the ACM*, 43(8):107–111, 2000.
- [40] M. Spiliopoulou and L. Faulstich. WUM—A Tool for WWW Utilization Analysis. In *Proc. of the International Workshop on the World Wide Web and Databases (WebDB'98)*, pages 184–203, 1998.
- [41] K. Sugiyama. *Studies on Improving Retrieval Accuracy in Web Information Retrieval*. PhD thesis, Nara Institute of Science and Technology, 2004.
- [42] K. Sugiyama, K. Hatano, M. Yoshikawa, and S. Uemura. A Method of Improving Feature Vector for Web Pages Reflecting the Contents of their Out-Linked Pages. In *Proc. of the 13th International Conference on Database and Expert Systems Applications (DEXA2002)*, pages 891–901, 2002.
- [43] K. Sugiyama, K. Hatano, M. Yoshikawa, and S. Uemura. Refinement of TF-IDF Schemes for Web Pages Using their Hyperlinked Neighboring Pages. In *Proc. of the 14th ACM Conference on Hypertext and Hypermedia (HT '03)*, pages 198–207, 2003.
- [44] K. Sugiyama, K. Hatano, M. Yoshikawa, and S. Uemura. User-Oriented Adaptive Web Information Retrieval Based on Implicit Observations. In *Proc. of the 6th Asia Pacific Web Conference (APWeb'04)*, 2004.
- [45] L. Terveen, W. Hill, B. Amento, D. McDonald, and J. Creter. PHOAKS: A System for Sharing Recommendations. *Communications of the ACM*, 40(3):59–62, 1997.
- [46] T. Tsandilas and M. C. Schraefel. User-Controlled Link Adaptation. In *Proc. of the 14th ACM Conference on Hypertext and Hypermedia (HT '03)*, pages 152–160, 2003.
- [47] J. Wang, Z. Chen, L. Tao, W.-Y. Ma, and L. Wenyin. Ranking User's Relevance to a Topic through Link Analysis on Web Logs. In *Proc. of the 4th ACM CIKM International Workshop on Web Information and Data Management (WIDM'02)*, pages 49–54, 2002.