Adaptive Web Search Considering User's Ephemeral and Persistent Preferences

Kazunari SUGIYAMA♥ Kenji HATANO◆
Masatoshi YOSHIKAWA♠
Shunsuke UEMURA◆

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 results 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 propose several approaches to adapting search results according to each user's information need considering their ephemeral and persistent preferences.

1. Introduction

It has become increasingly difficult for users to find information on the WWW that satisfies their individual needs. Web search engines help users find useful information on the WWW. In order to achieve much better retrieval accuracy, hyperlink structures of the Web are focused on [3], [6], [12]. However, when the same query is submitted by different users, these systems return the same results regardless of who submits the query. In general, each user has different information needs for his/her query. Therefore, Web search results should be adapted to users with different information needs. Novel information systems that personalize information or provide more relevant information for users have been proposed [2], [8], [9], [11]. In these systems, however, users have to register personal information beforehand, or provide feedback on relevant or irrelevant judgments, ratings, and so on. These types of manipulation can become time consuming and users prefer easier methods. Therefore, we propose several approaches that can be used to adapt search results according to each user's information need by capturing changes of each user's preferences without any user effort.

2. Our Proposed Method

Figure 1 shows an overview of our system. In the following sections, we explain how to construct a user profile in the update profile component illustrated in Figure 1. We construct each user profile based on the following two methods: (1) Pure browsing history, and (2) Modified collaborative filtering.

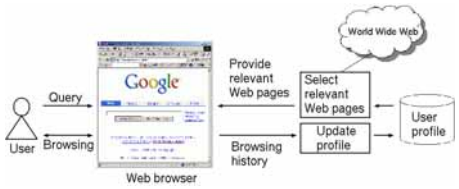

Figure 1. Overview of our system.

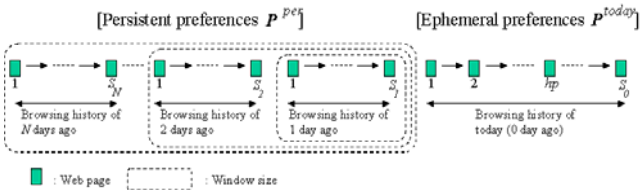

Figure 2. Window size for constructing persistent user profile.

2.1 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 preferences, and (2) ephemeral preferences. Using these two factors, 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 (see Figure 2). In each day, $\mathbf{P}^{today}$ is constructed through the following process. First, we denote the feature vector $\mathbf{w}^{hp}$ of browsed Web page hp as follows:

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

where m is the number of unique terms in the Web page hp, and $t_k (k = 1,2,\cdots,m)$ denotes each term. Using the TF (Term Frequency) scheme, we also define each element $w_{t_k}^{hp}$ of $\mathbf{w}^{hp}$ as follows:

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

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

$$c^{hp} = \begin{cases} 1; & dr \geq Th \\ 0; & dr < Th \end{cases}$$

where dr denotes the time spent reading normalized by the number of terms in Web page hp. We define threshold Th as 0.317 based on preliminary experiments. We then denote user profile $\mathbf{P}^{today}$ as follows:

$$\mathbf{P}^{today} = (p_{t_1}^{today}, p_{t_2}^{today}, \cdots, p_{t_m}^{today}),$$

and define each element $p_{t_k}^{today}$ as follows:

$$p_{t_k}^{today} = \frac{1}{S_0} \sum_{hp=1}^{S_0} w_{t_k}^{hp},$$

as described above, $\mathbf{P}^{today}$ shows a user profile

♥ Student Member Graduate School of Information Science, Nara Institute of Science and Technology (at being recommended for DBSJ Letters Vol.3, No.1)
kazuna-s@is.naist.jp
◆ Member Graduate School of Information Science, Nara Institute of Science and Technology
{hatano, uemura}@is.naist.jp
♠ Director Information Technology Center, Nagoya University yosikawa@itc.nagoya-u.ac.jp

constructed using the user's browsing history of today's Web page. Moreover, we set window size N $(N = 1,2,\cdots,30)$ to construct $\mathbf{P}^{per}$. We also denote $\mathbf{P}^{per}$ as follows:

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

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

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

where $e^{-\frac{\log 2}{hl}(d - d_{t_{k\_init}})}$ is a forgetting factor under the assumption that user's preferences gradually decay as days pass. In this factor, $d_{t_{k\_init}}$ is the day when term t_k initially occurs, d is the number of days following to $d_{t_{k\_init}}$, and hl is a half life span parameter. We set the half-life span hl to 7. In other words, the intuition behind this assumption is that user's preferences reduce by 1/2 in one week. Let us assume that each user browsed S_N pages on each day. Of course, this value of S_N, the number of browsed Web pages, differs user by user. Therefore, we normalize $p_{t_k}^{per}$ using S_N as shown in Equation (2). Using these parameters, we finally construct user profile $\mathbf{P}$ as follows:

$$\mathbf{P} = a\mathbf{P}^{per} + b\mathbf{P}^{today}, \quad (3)$$

where a and b are constants that satisfy $a + b = 1$.

2.2 User Profile Construction Based on Modified Collaborative Filtering Algorithm

In the pure collaborative filtering algorithms, a user-item ratings matrix is usually considered [5]. Similarly, in the construction of a user profile, we can consider a user-term weights matrix like that shown in Figure 3(a). In addition, we can apply predictive algorithms in the pure collaborative filtering to predict missing term weights in each user profile. In this approach, we propose the following two methods: user profile construction (1) based on the static number of users in the neighborhood, and (2) 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:

(i) 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.

(ii) Select n users that have the highest similarity to the active user. These users form the neighborhood.

(iii) Compute a prediction from a weighted combination of the neighbor's term weights.

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

$$S_{a,u} = \frac{\sum_{i=1}^{T}(w_{a,i} - \overline{w}_a) \times (w_{u,i} - \overline{w}_u)}{\sqrt{\sum_{i=1}^{T}(w_{a,i} - \overline{w}_a)^2 \times \sum_{i=1}^{T}(w_{u,i} - \overline{w}_u)^2}}, \quad (4)$$

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		0.461		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		0.461		0.928			0.686			0.172

(b)

Figure 3. User-term weights matrix for modified collaborative filtering [(a) when each user browsed k Web pages, (b) when each user browsed $k+1$ pages].

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

In step (ii), 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 next step (iii).

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

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

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 in Equation (4), 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:

(i) Generate clusters of users by means of the k-Nearest Neighbor algorithms [7]. The similarity between user a and these clusters is measured as the Pearson correlation coefficient between their term weight vectors.

(ii) 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.

(iii) Compute a prediction from a weighted combination of the term weights using centroid vectors of clusters.

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

$$S_{a,g} = \frac{\sum_{i=1}^{T}(w_{a,i} - \overline{w}_a) \times (w_{g,i} - \overline{w}_g)}{\sqrt{\sum_{i=1}^{T}(w_{a,i} - \overline{w}_a)^2 \times \sum_{i=1}^{T}(w_{g,i} - \overline{w}_g)^2}}, \quad (5)$$

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

In step (ii), 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 (iii). 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 that is better adapted to each user's preferences.

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

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

where $p_{a,i}$ is the prediction for the active user a for weight of term i, $S_{a,g}$ is the similarity between users a and centroid vectors of clusters g, as described in Equation (5), and n is the number of centroid vectors of clusters in the neighborhood.

3. Experiments

3.1 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, and (3) user profiles based on the modified collaborative filtering algorithm. 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. We used 50 query topics that were employed as test topics in the TREC WT10g test collection [4]. In our experiments, we observed the browsing history of 20 subjects for 30 days. In the following, let the h^{th} Web page in the search results, $\mathbf{w}^{rp_h}$, is defined as follows:

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

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

$$sim(\mathbf{P}, \mathbf{w}^{rp_h}) = \frac{\mathbf{P} \cdot \mathbf{w}^{rp_h}}{|\mathbf{P}| \cdot |\mathbf{w}^{rp_h}|}. \quad (6)$$

Based on the value obtained in Equation (6), the search results are adapted to each user according to his/her profile. These results are compared with the search results of Google [3]. 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.

3.2 Experimental Results

3.2.1 User Profile Based on Relevance Feedback

In our experiment, we use the Rocchio formulation [10] 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 believe that the new query vector $\mathbf{Q}^{new}$ obtained by the user's judgment, whether the retrieved documents are relevant or not, reflects the user's preferences. Therefore, we treat $\mathbf{Q}^{new}$ as $\mathbf{P}^{today}$ defined by Equation (3), and employ $\mathbf{Q}^{new}$ as an initial preference of a user to construct a user profile. In this case, using Equation (3), the user profile $\mathbf{P}$ is defined as follows:

$$\mathbf{P} = a\mathbf{P}^{per} + b\mathbf{Q}^{new}, \quad (7)$$

We asked each subject to judge if the search results returned by the search engine are relevant, and constructed the user profile $\mathbf{P}$ based on Equation (7). In this experiment, we varied the number of feedbacks FB that each subject provided from 1 to 3.

3.2.2 User Profile Based on Pure Browsing History

In this approach, each user profile is constructed as mentioned in Section 2.1. The user profile $\mathbf{P}$ is defined as follows:

$$\mathbf{P} = a\mathbf{P}^{per} + b\mathbf{P}^{today}.$$

3.2.3 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 3(b). These missing values are predicted using the algorithms described in Section 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 $\mathbf{P}^{today}$ defined by Equation (3), and employ $\mathbf{V}^{pre}$ as an initial preference of a user to construct a user profile. In this case, using Equation (3), the user profile $\mathbf{P}$ is defined as follows:

$$\mathbf{P} = a\mathbf{P}^{per} + b\mathbf{V}^{pre}.$$

3.2.4 Summary of Experimental Results

Table 1 summarizes the best precisions obtained using our proposed methods. In the relevance feedback-based user profile, we could not observe 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. The user profile based on pure browsing history can achieve higher precision than the relevance feedback-based method, and the results show that the user's browsing history strongly reflects the user's preference. In addition, in the user profile based on modified collaborative filtering, the best precision is obtained in the case of $n = 5$, in other words, 5 nearest neighbors of each user are taken in the static approach described in Section 2.2(1). Therefore, 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 2.2(2), we could obtain the best precision in all of our experimental results. 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 the static method.

4. 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 by capturing changes in each user's preferences. We found that the user profile constructed based on modified collaborative filtering achieved the best accuracy. This approach allows fine-grained search that is better adapted to each user's preferences. We believe that the technique proposed in this paper can be applied to situations where users require more relevant information to satisfy their information needs.

Table 1. Comparison of the best precision obtained using our proposed methods.

	% best precision	%improvement
Google	36.10	--
Relevance feedback-based user profile (*FB=2, a=0.604, b=0.396, N=26*)	46.91	+10.81
Pure browsing history-based user profile (*a=0.617, b=0.383, N=18*)	48.77	+12.67
Modified collaborative filtering-based user profile (static, *n=5, a=0.622, b=0.378, N=17*)	50.82	+14.72
Modified collaborative filtering-based user profile (dynamic, *n=5, a=0.613, b=0.387, N=28*)	51.34	+15.24

[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] 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)*, pp. 107–117, 1998.

[4] D. Hawking. Overview of the TREC-9 Web Track. *NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC-9)*, pp. 87–102, 2001.

[5] J. Herlocker and J. Konstan and 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)*, pp. 230-237, 1999.

[6] IBM Almaden Research Center. Clever Searching. http://www.almaden.ibm.com/cs/k53/clever.html.

[7] R. A. Jarvis and E. A. Patrick. Clustering Using a Similarity Measure Based on Shared Near Neighbors. IEEE Transactions on Computers, Vol. C22, No.11, pp.1025-1034, 1973.

[8] U. Manber and A. Patel and J. Robison. Experience with Personalization on Yahoo! *Communications of the ACM*, 40(3):35-39, 1997.

[9] P. Melville and R. J. Mooney and R. Nagarajan. Content-Boosted Collaborative Filtering for Improved Recommendations. In *Proc. of the 18th National Conference on Artificial Intelligence (AAAI2002)*, pp.187-192, 2002.

[10] J. Rocchio. Relevance Feedback in Information Retrieval. In G. Salton, editor, The Smart Retrieval System: Experiments in Automatic Document Processing, pp. 313-323. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[11] J. B. Schafer and 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)*, pp. 43-51, 2002.

[12] 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)*, pp. 198–207, 2003.

Kazunari SUGIYAMA
has graduated from Graduate School of Information Science, Nara Institute of Science and Technology, and is currently working for HITACHI, Ltd., Software Division. He has been working in information retrieval. He is a member of ACM, IEEE, AAAI, IEICE, IPSJ, and JSAI.

Kenji HATANO
is an assistant professor of Graduate School of Information Science, Nara Institute of Science and Technology. He has been working in XML database and information retrieval. He is a member of ACM, IEEE CS, IEICE, and IPSJ.

Masatoshi YOSHIKAWA
is a professor of Information Technology Center, Nagoya University. He has been working in database system. He is a member of ACM, IEEE CS, IEICE, and IPSJ.

Shunsuke UEMURA
is a professor of Graduate School of Information Science, Nara Institute of Science and Technology. He has been working in database system. He is a fellow of IEEE, IEICE, and IPSJ.