A Reinforcement Learning Agent for Personalized Information Filtering

Young-Woo Seo
Artificial Intelligence Lab (SCAI)
Dept. of Computer Engineering
Seoul National University
Seoul, 151-742, Korea
Phone: +82 2 880 1835
ywseo@scai.snu.ac.kr

Byoung-Tak Zhang
Artificial Intelligence Lab (SCAI)
Dept. of Computer Engineering
Seoul National University
Seoul, 151-742, Korea
Phone: +82 2 880 1833
btzhang@scai.snu.ac.kr

ABSTRACT
This paper describes a method for learning user's interests in the Web-based personalized information filtering system called WAIR. The proposed method analyzes user's reactions to the presented documents and learns from them the profiles for the individual users. Reinforcement learning is used to adapt the term weights in the user profile so that user's preferences are best represented. In contrast to conventional relevance feedback methods which require explicit user feedbacks, our approach learns user preferences implicitly from direct observations of user behaviors during interaction. Field tests have been made which involved 7 users reading a total of 7,700 HTML documents during 4 weeks. The proposed method showed superior performance in personalized information filtering compared to the existing relevance feedback methods.

Keywords
Web-based information filtering, user interface agents, learning user's preferences, reinforcement learning.

INTRODUCTION

With the rapid progress of computer technology in recent years, electronic information has been explosively increased. This trend is especially remarkable on the Web. As the availability of the information increases, the need for finding more relevant information on the Web is growing [4].

Currently, there are two major ways of accessing information on the Web. One is to use Web index services such as AltaVista, Yahoo!, and Excite. The other is to manually follow or browse the hyperlinks of the documents by a user himself. However, these methods have some drawbacks. Since Web-index services are based on general purpose indexing methods, much of the retrieved results may be irrelevant to user's interests. In addition, the manual browsing involves much time and efforts. High quality service requires to catch the personal interests of individual users during the interaction with the information retrieval system.

In this paper, we describe a method for learning user's interests by observing user behaviors during his interaction with the system. Based on the observations, the system estimates user's relevance feedback implicitly. This information is used to modify the profiles using reinforcement learning. Reinforcement learning is a goal-directed learning method based on interactions with the environment. We take the reinforcement learning method because it can learn on-line in incremental fashion. The method is implemented as a Web-based personalized information filtering system called WAIR (Web-Agents for Information Retrieval).

The paper is organized as follows. In the following section, we review existing methods for learning user preferences. Then, our method for learning user interests is described along with the architecture of the WAIR system and its filtering procedure. Finally, experimental results are reported and conclusions are drawn.

IUI 2000 New Orleans LA USA

GETTING RELEVANCE FEEDBACKS

In information retrieval and filtering, users are usually not able to express their interests and information needs with exact terms. But, they easily can evaluate on whether a document is relevant or not to their information needs. The evaluation by a user is called user relevance feedback and is used for improving retrieval and filtering accuracy. These opinions are reflected in his (or her) profiles. Most profiles are composed of multiple terms and their weights. Therefore, updating the profile involves selecting terms and modifying their weights.

A general model of information retrieval is the vector space model that represents queries and documents as vectors of terms [4]. In this model, Rocchio has suggested a relevance feedback method as follows [8]:

$$Q' = Q + \frac{1}{n_1}\alpha\sum_{i=1}^{n_1} R_i - \frac{1}{n_2}\beta\sum_{i=1}^{n_2} S_i,$$

where Q is the vector for the initial query, R_i is the vector for relevant document i, S_i is the vector for irrelevant document i, and α, β are Rocchio's weights. Q' is the modified vector of the original query plus the vectors of the relevant and the irrelevant documents. Ide has devised three particular strategies extending Rocchio's work [9]. To compare its performance in the experiment, we describe one of the methods here:

$$Q' = \alpha Q + \beta\sum_{i=1}^{n_1} R_i - S_1,$$

where S_1 is the top-ranked irrelevant document. But these methods have several drawbacks. One is that they cannot discriminate well which term is more relevant to the user's initial need because they make use of all the terms in the

retrieved documents. Another drawback is that the cost for computing vector sum is very high in relatively dynamic document sets like the Web. Therefore, a straightforward Rocchio algorithm is not appropriate for on-line information filtering on the Web.

The proposed method exploits a method that reflects user's opinion directly to the terms in the profile. That is, if a term in the document estimated as "relevant" is included in the user's profile, the weight of the term is updated as follows:

$$w_{pk} \leftarrow w_{pk} + \beta r_i, \qquad \text{if } k \in D_i \qquad (1)$$

where r_i is the relevance feedback to the filtered document i, and w_{pk} is weight of the kth term for profile p. Here β is the learning rate that controls the step size. The importance of a term is increased as it filters more relevant documents.

So far, we described methods that modify the initial user information needs into more descriptive terms. These methods have a drawback that the user has to participate in relevance feedback himself. The more a filtering system gets user's opinions, the less convenient the system is to use. In the next section, we describe methods that get user's potential opinions by observing his behaviors during the interaction with the information filtering system.

LEARNING USER PREFERENCES BY ANALYZING USER BEHABIORS

As discussed above, it is important in personalized information systems to get user's preferences without requiring extra efforts from him.

Letizia [6], which is an assistant for browsing the Web, traced the user behavior in the conventional Web browser. It analyzed his (or her) behaviors, such as following-up of the hyperlinks in an HTML document. And then it estimated his interests by parsing the document and recommending HTML documents.

WebWatcher [7] learns the user interests using reinforcement learning as in WAIR. In WebWatcher, it is assumed that the information space is linked with hyperlinks. While the retrieval agent seeks the relevant documents, it is directed by the value of reinforcement learning:

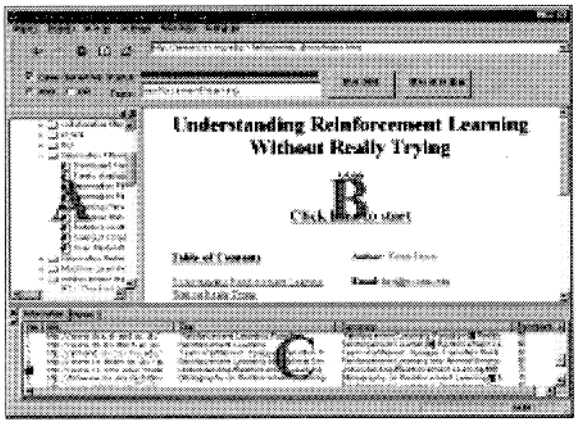
$$Q_{n+1}(s,a) = R(s') + \gamma \max_{a' \in actions\_in\_s'} [Q_n(s',a')]$$

Here, *Q-value* is the discounted sum of future rewards that will be obtained when the agent follows a hyperlink in an HTML document and subsequently chooses the optimal hyperlink.

As described above, we analyze user's behaviors in the Web browser and then estimate explicit user's feedback by using analyzed information. We refer to this type of feedback as *implicit* feedback. Implicit feedback (I) is obtained by observing user's behavior on the filtered ith document (D_i). It consists of several factors: the time for reading (rt), bookmarking (bm), and following up (fl) the hyperlinks in the filtered documents. The total score of implicit feedback is computed as:

$$R_I(i) = \sum_{v \in F} c_v f_v(i), \qquad (2)$$

where, $F = \{rt, bm, fl\}$ and c_v is a weight for each factor of the implicit feedback. It is determined by an of explicit feedback sessions.



[Figure 1: The user interface of WAIR.]

LEARNING FROM USER WEB-BROWSING BEHAVIORS FOR PERSONALIZED INFORMATION FILTERING

WAIR consists of three agents: a user-interface agent, a Web-document retrieval agent, and a learning agent. The user-interface of WAIR directly interacts with the user (Figure 1). Part A is a repository of bookmarks. Part B a browser where the agent observes the user's behavior. Part C is for presenting the filtering results and getting the user's explicit feedback.

The duties of each agent can be summarized as follows.

- User-interface agent: It observes user's behavior by using a way of "looking over his (or her) shoulder"[6]. The relation between user's behavior and document relevance will be discussed in more detail in the following section.

- Web-document retrieval agent: Getting the user's interests from the user-interface agent, it retrieves a set of candidate HTML documents. It sets a starting-point of retrieval using meta-search.

- Learning agent: It adapts user's profile by using reinforcement learning. Data for learning is supplied by the user-interface agent. It guides the retrieval direction of the retrieval agent. This means that the learning agent supplies the Web-document retrieval agent with relevance criteria which is a modified profile.

The agents are closely related with each other for personalized information filtering, but we do not focus on their interaction in this paper. Figure 2 shows the overall procedure of the Web-document filtering. The profile for a user consists of one or more topics p. Topics represent user's information needs. In this section, we assume that a profile consists of a single topic. Then, the profile p is represented as a vector: $\mathbf{w}_p = (w_{p1}, w_{p2}, \ldots, w_{pk}, \ldots, w_{pn})$, and w_{pk} is the weight of k-th term where n is the number of terms used for describing profiles. The ultimate goal of WAIR is to learn the profile of the user to filter documents that best reflect his preferences. WAIR searches the Web-documents by using existing Web-index services. That is, it requests the Web-index services for documents and receives N URLs.

The similarity (or relevance) of i and profile p is computed as:

1. Get the initial profile p from the user. Set $t \leftarrow 0$.
2. Generate a collection of candidate HTMLs.
   2.1 Generate N URLs using the existing search engines.
   2.2 While $i \leq N$ do:
   - Retrieve the ith HTML document.
   - Preprocess the document.
   - Estimate the relevance value V_{pi} by Eqn. (3).
   - $i \leftarrow i + 1$.
3. Filter and present k highest-ranked documents.
4. Get the feedback r_i (Eqn. (4)) from the user.
5. Update the user profile by Eqn. (1).
6. Select the next retrieval points from the relevant documents.
7. Set $t \leftarrow t+1$, Goto step 2.

[Figure 2: The overall filtering procedure of WAIR.]

$$V_{pi} = \sum_{k=1}^{n} tf_{ik} \times w_{pk}, \quad \text{if } k \in D_i \qquad (3)$$

where tf_{ik} is the frequency of the kth term in D_i, and w_{pk} is the weight of term k for profile p. We do not use the general $tf \cdot idf$ (term frequency $\cdot$ inverse document frequency) [3] based indexing method in preprocessing the HTML documents. For the reason that we focus on filtering the information stream, it is difficult to define the static document sets. Since we do not take a serious view of term's weight in the document, term weights are maintained only for the profile. The candidate documents are presented to the user sorted by descending order in the retrieval value V_{pi}. After the filtering, the user evaluates the results. WAIR acquires the user's explicit $E(t)$ or implicit $I(t)$ feedback by using equation (2).

WAIR exploits reinforcement learning to learn the profiles. Reinforcement learning is a goal-directed learning method based on interactions with the environment [1]. The learner receives a scalar-valued feedback called reward when it chooses and takes an action at given time and a given state. The objective is to maximize the expected value of the cumulative reward it receives in the long run from the environment [1][2].

The goal of learning in WAIR is to look for the best state of the profile. States in our problem are defined as a profile vector: $\mathbf{w}_p = (w_{pk})$, where w_{pk} is the weight of term k. Accordingly, the best state of the profile can be interpreted as the most similar representation of the user's interests. In our reinforcement learning approach, actions are defined as the picking up of terms that participate in estimating the relevance between documents and the profile. To find the state reflecting the user's interests well, we employ the ε-greedy term-selection method. Specifically, WAIR selects the m terms from the specific profile p, where $m - \varepsilon$ terms have been sequentially selected in order of higher weights and ε terms have been randomly selected. Since this selection method uses its current knowledge about user's interests, it can be described exploitative search [1]. And, it also gives a boost for profiles to find the terms to discriminate which documents are more relevant to the initial user's interests by the notion of exploration [1].

For a document D_i presented by WAIR, we define a scalar-valued feedback from the observation of user behavior as

$$r_i = \alpha R_E(i) + (1-\alpha) R_I(t), \quad 0 \leq R_E(i) \leq 1, 0 \leq R_I(i) \leq 1 \qquad (4)$$

where $R_E(i)$ is an explicit feedback and $R_I(i)$ is an implicit feedback estimated by equation (2). α is a regulating factor that adjusts the ratio of implicit and explicit feedback. Then, the profile is updated as expanded in (1):

$$w_{pk} \leftarrow w_{pk} + \beta r_i, \qquad \text{if } k \in D_i.$$

At time t, WAIR presents a collection of N documents. We also estimate the value of the whole collection with respect to the topic p.

$$V_{p,t} \leftarrow V_{p,t-1} + \alpha[R_t + \gamma \Delta V_{p,t}] \qquad (5)$$

$$\text{where} \quad R_t = \frac{1}{N}\sum_{i=1}^{N} r_i \text{ and } \Delta V_{p,t} = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{n} w_{pk,i} - w_{pk,i-1}.$$

□ is a discount factor which determines the present value of the expected future reward. We approximate the estimated value about the future reward as the change of term weights. A positive value of the change means that the current terms selected are good choices for representing the user's interests and the current contents of the profile will obtain the positive furture feedback from the user.

EXPERIMENTAL RESULTS

Several experiments have been carried out using the proposed method. The objective of the first experiment was to compare the filtering accuracy between the proposed method and the conventional methods. In this experiment, 7 people volunteered to suggest 14 topics. These 14 topics amount to a total of 5,600 HTML documents.

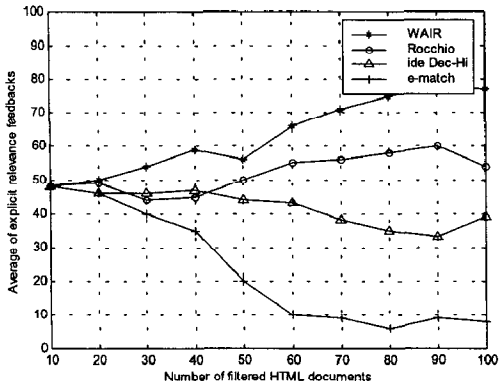
Figure 3 is the average of user feedbacks on 14 topics. Participants evaluate the results explicitly. Also shown are the results for the "e-match" (exact match). E-match is a baseline method in which no relevance feedback is obtained from the user. It follows up the hyperlink which is exactly anchored with user's initial query term in the HTML document. It can be seen how the evaluation results evolve for the different relevance feedback methods. Though the performance decreased at some intermediate steps, the general tendency is that all the feedback methods learn the preference of the user after five or more interactions. The learning effect of WAIR was the greatest compared to that of other feedback methods.

Table 1 summarizes the result of the first experiment in which participants evaluated 400 documents presented for each of the topics.

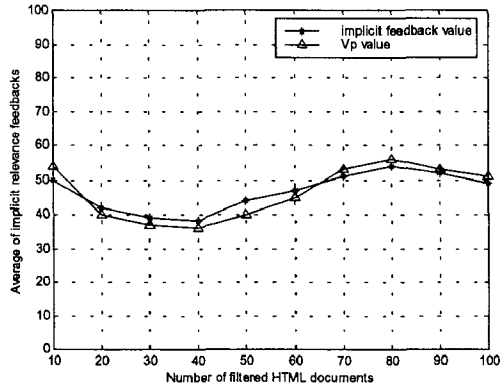
Feedback	Relevant	Neutral	Irrelevant	Total
Bookmark	922 (61%)	575 (38%)	15 (1%)	1,512 (27%)
Follow-up	3,168 (69%)	1194 (26%)	230 (5%)	4,592 (82%)
Reading time	35 ≥ RT ≥ 22	22 > RT ≥ 7	7 ≥ RT > 0	5,376 (96%)
	3,118 (58%)	1,344 (25%)	914 (17%)	

[Table 1: The relation between user's behavior and relevance]

Based on the statistical analysis of the results, we can conclude that the major factor of implicit feedback is the bookmarking of HTML documents. It means that the bookmarked URLs reflect strong opinions of users on the relevance. In the column of "follow-up" in Table 1, we see that most participants followed up the hyperlinks to evaluate the document. Although the number of hyperlinks



[Figure 3: Comparison of WAIR with other relevance feedback methods.]



[Figure 4: Experiment for verifying the implicit relevance feedback.]

followed up is important, follow-up itself does not affect user's opinion. There is also a tendency that the HTML documents on which the user spent a long time to read were rated as "relevant" and the documents for which only a short time is spent are evaluated as "irrelevant." From Table 1, we determined the weight of each implicit factor; it was 0.6 for boomark (*bm*), 0.3 for reading time (*rt*), and 0.1 for follow-up (*fl*). These weights were used for the next experiment, in which we compared the degree of the user's satisfaction about the presented HTML documents.

Figure 4 shows that the proposed method increases the degree of user's satisfaction by using the implicit relevance feedback. The graph of V_p values in Figure 4 means a process of learning. It is also described "bootstrapping" which update estimates of the values of states based on estimates of the values of successor states [1]. At time t, the V_p value for topic p is the sum of the current value of evaluation from the user and the estimated value of future evaluation. That is, the learning agent bootstraps itself to more relevant points of the user's interests by considering the current reward and the estimated future reward. Thus, it directs the learning agent to the best point of the term space about user's specific interests.

CONCLUDING REMARKS

We proposed a method for information filtering that obtains the relevance information by observing user behaviors during interactions. Through the experiments on a group of users, we verified that the method can provide documents which are more relevant to the user's specific interests when compared with other feedback methods. It also effectively adapts to the user's specific interests with implicit feedbacks only. In terms of adaptation speed, the proposed method converged on the user's specific interest faster than existing relevance feedback methods. Based on the results, we can conclude that "learning from shoulder of the user" can significantly improve the performance of personalized information filtering systems.

In spite of our success in learning the user preferences in the WAIR system, it should be mentioned that the success comes in part from the environments where we made our experiments. One is that the topics used for experiments were usually scientific and thus the filtered documents contained relatively less-ambiguous terms than those that might be contained in other usual Web documents. Another reason might be that the duration of our experiments were not very long during which the user interests did not change very much. Adaptation to user's interests during a longer period of time in a more dynamic environment should still be tested.

ACKNOWLEDGEMENTS

This research was supported in part by the Korean Ministry of Information and Telecommunications under Grant 98-199 through IITA.

REFERENCES

1. Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction*, The MIT Press, 1998.
2. Tom M. Mitchell, *Machine Learning*, McGraw-Hill, 1997.
3. Gerard Salton, *Automatic Text Processing*, Addison Wesley, 1989.
4. Nicholas J. Belkin and W. Bruce Croft, Information filtering and information retrieval: two sides of the same coin?, *Communications of the ACM*, Vol. 35, No. 12, pp. 29-38, Dec. 1992.
5. Pattie Maes, Agents that reduce work and information overload, *Communications of the ACM*, Vol. 37, No. 7, pp. 31-40, 1994.
6. Henry Lieberman, Letizia: An agent that assists Web browsing, In *IJCAI '95*, pp. 475-480, 1995.
7. T. Joachims, Dayne Freitag, and Tom Mitchell, WebWatcher: A tour guide for the World Wide Web, In *IJCAI '97*, pp. 770-777, 1997.
8. J.J. Rocchio, Relevance feedback in information retrieval, In *The SMART Retrieval System*, Prentice Hall, pp. 313-323, 1971.
9. E. IDE, New experiments in relevance feedback, In *The SMART Retrieval System*, Prentice Hall, pp. 337-354, 1971.