Modeling and Learning User Profiles for Personalized Content Service

Heung-Nam Kim, Inay Ha, Seung-Hoon Lee, and Geun-Sik Jo

Intelligent E-Commerce Systems Laboratory,
Department of Computer Science & Information Engineering, Inha University
{nami,inay,shlee}@eslab.inha.ac.kr, gsjo@inha.ac.kr

Abstract. With the spread of the digital library and the web, users can obtain a wide variety of information, and also can access novel content. In this environment, finding useful information from a huge amount of available content becomes a time consuming process. In this paper, we focus on user modeling for personalization to recommend content relevant to user interests. We exploit the data mining techniques for identifying useful and meaningful patterns of users. Each user model, collectively called PTP (Personalized Term Pattern), is represented as both interest patterns and disinterest patterns. We present empirical experiments using *NSF research award* datasets to demonstrate our approach and evaluate performance compared with existing methods.

1 Introduction

Numerous technological developments related to the Internet and the World Wide Web provide anyone living in today's information society with accessing a variety of content and information on the web. Due to the nonstop growth of the internet information, users often face the challenges with huge amount of content, and need to waste plenty of time to find content relevant to their interest. Beside that, the advent of bolgs, DL (digital library), and RSS (Really Simple Syndication) generate millions of information overnight. As a result, such an information overload increases user's frustration to find out the content of their interest. Therefore, a user modeling for efficient personalization plays a significant role in modern information filtering system [7, 14].

One notable challenge in a user modeling is the ability to identify meaningful or useful patterns for users. In content-based personalization it is important to recognize meaningful patterns for representing items or contents [11]. For example, when content contains 'apple Macintosh computer', the semantic of 'apple' is discriminated from those of 'apple' in 'apple pie'. Likewise, mouse in 'optical mouse' implies not an animal but an input device of computers. Therefore, our aim is to build a user model that supports the identification of useful patterns of users, and thus can be used for personalized recommendation services. In our research, we exploit a data mining technique for identifying important pattern of user's preferences. Considering the contents of user interest (positive) and disinterest (negative), we mine the frequent term patterns residing in the user's positive contents and negative contents. And each

D.H.-L. Goh et al. (Eds.): ICADL 2007, LNCS 4822, pp. 85–94, 2007.
© Springer-Verlag Berlin Heidelberg 2007

user model, collectively called PTP (Personalized Term Pattern), is represented as both interest patterns and disinterest patterns, which will boost recommendations of contents related to user interests. We also take advantage of content-based filtering approach to recommend content that is very close to not negative term patterns but positive term patterns.

The subsequent sections of this paper are organized as follows: The next section contains our approach for modeling user preference. In section 3, we describe a content-based filtering for a personalized recommendation. A performance evaluation is presented in section 4 and related work is discussed in section 5. Finally, we conclude with a discussion and future directions.

2 Modeling User Profiles from Positive and Negative Examples

In this section, we describe our approach to modeling user preference, which is mined from the user's interest contents (positive contents) and disinterest contents (negative contents). The proposed method is divided into three main types of tasks: (a) Observing relevance feedback of a given user, (b) Modeling user preference from observed contents, and (c) Generating content recommendations for a given user. Fig. 1 provides a brief overview of the proposed approach.

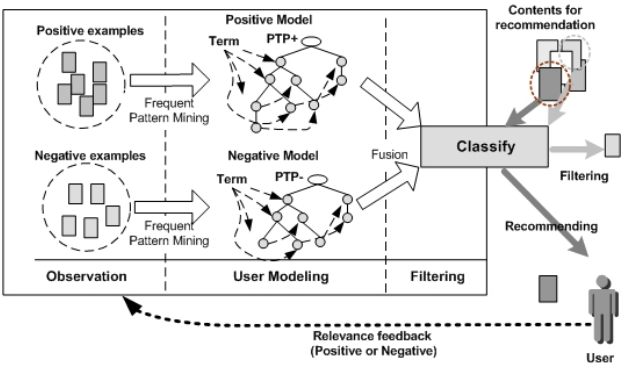

Fig. 1. Overview of the proposed method for personalized content recommendations

Since every user has different interests, feature selection for representing users' interests should be personalized and be performed individually for each user [7]. The first step in user modeling is the extraction of the terms from positive or negative contents that have been preprocessed by removing stop words and stemming words [12]. After extracting terms, each content C_j is represented as a vector of attribute-value pairs; $\vec{C}_j = (w_{1,j}, w_{2,j}, \ldots, w_{m,j})$, where $w_{i,j}$ is the weight of term T_i in C_j, which is computed by static TF-IDF term-weighting scheme [1] and defined as follows:

$$w_{i,j} = \frac{f_{i,j}}{\max_l f_{l,j}} \times \log \frac{n}{n_i}$$

where $f_{i,j}$ is the frequency of occurrence of term T_i in content C_j, n is the total number of contents in the collections, and n_i is the number of contents in which term T_i occurs.

Secondly, the frequent patterns that occur at least as frequently as a predetermined minimum support (*min_sup*), i.e., $PS > min\_sup$, are mined from the positive examples and the negative examples, respectively [9]. If the pattern support of pattern P_k, that is composed of at least l different terms ($l \geq 2$), satisfies a pre-specified minimum support threshold, then pattern P_k is a frequent term pattern. Therefore, two types of a set of terms (pattern) for a user can be discovered; a set of positive frequent patterns, written as F_u^+, and a set of negative frequent patterns, written as F_u^-, $F_u^+ \cup F_u^- = F_u$. In our research, a content of a user corresponds to a transaction and terms extracted from the content are items in transaction.

Definition 1 (Pattern Support, PS). Let $T = \{T_1, T_2, \ldots, T_m\}$ be a set of terms, I_u a set of positive contents of user u where each content C^+ is a set of terms such that $C^+ \subseteq T$, and N_u a set of negative contents of user u where each content C^- is a set of terms such that $C^- \subseteq T$. Let pattern P_k be a set of terms. A content is said to contain a pattern if and only if $P_k \subseteq C^+$ or $P_k \subseteq C^-$. *Pattern support* for pattern P_k, $PS(P_k)$, in I_u or N_u is the ratio of contents in I_u or N_u that contain pattern P_k.

Definition 2 (Personalized Term Pattern, PTP). *Personalized term patterns* for user u, PTP_u, is defined as a set of frequent term patterns whose *pattern weights* are greater than a threshold value θ, i.e., $PTP_u \subseteq F_u$ and $PW(P_k) > \theta$. PTP_u can be divided into two groups, a set of positive patterns, written as PTP_u^+, and a set of negative patterns, written as PTP_u^-, such that $PTP_u^+ \subseteq F_u^+$, $PTP_u^- \subseteq F_u^-$, and $PTP_u^+ \cup PTP_u^- = PTP_u$.

Definition 3 (Pattern Weight, PW). Let $T(P_k) = \{T_1, T_2, \ldots, T_n\}$ be a set of terms contained in pattern P_k such that $P_k \in F_u$. *Pattern weight* of P_k , denoted as $PW(P_k)$, indicates the importance of each term in representing the pattern and is computed as follows:

$$PW(P_k) = \frac{1}{|T(P_k)|} \cdot \sum_{i \in T(P_k)} \left(\frac{1}{|E_u(i)|} \times \sum_{j \in E_u(i)} w_{i,j} \right)$$

where $E_u(i)$ is a set of positive (or negative) contents containing term T_i for user u and $w_{i,j}$ is the term weight of term T_i in content C_j.

Finally, we remove the patterns, which contain unnecessary terms, from F_u^+ and F_u^- of user u and model the user preference based on those patterns. A formal description of a model for user u, $\mathbf{M}_u$, follows: $\mathbf{M}_u = \langle (\mathbf{PTP}_u^+, \mathbf{PTP}_u^-) \rangle$, where PTP_u^+ models the interest patterns and PTP_u^- models the disinterest patterns (Definition 2). In other words, the PTP_u^+ is a set of personalized term patterns mined from positive contents of user u whereas the PTP^- is a set of personalized term patterns mined from negative contents of user u.

To save memory space and explore relationships of terms, the model is stored in a prefix tree structure, called Personalized Term Pattern tree [12]. For example, if four positive patterns are found after mining content of interest for user u as shown in Table 1(left), a tree structure of a model for user u is then constructed as follows.

Table 1. After mining positive and negative content of user u, four positive personalized term patterns (left table) and four negative personalized term patterns are found (right table)

Pattern	PTP^+	PS
P_1^+	$\{T_1, T_2, T_3\}$	0.56
P_2^+	$\{T_1, T_2, T_3, T_4\}$	0.51
P_3^+	$\{T_1, T_2, T_5\}$	0.47
P_4^+	$\{T_2, T_3, T_4\}$	0.32

Pattern	PTP^-	PS
P_1^-	$\{T_5, T_6, T_7\}$	0.52
P_2^-	$\{T_4, T_5, T_6\}$	0.41
P_3^-	$\{T_5, T_7\}$	0.37
P_4^-	$\{T_6, T_8\}$	0.31

All terms are stored in header table and sorted according to descending order of their frequency. First, create the root of the tree, labeled with "null". For the first term pattern, $\{T_1, T_2, T_3\}$ is insert into the tree as a path from root node where T_2 is linked as child of the root, T_1 is linked to T_2, and T_3 is linked to T_1. And *PS* and *length* of the pattern ($PS(P_1^+)$=0.56, $|T(P_1^+)| = 3$) are then attached to the last node T_3. For the second pattern, since its term pattern, $\{T_1, T_2, T_3, T_4\}$, shares common prefix $\{T_2, T_1, T_3\}$ with the existing path for the first term pattern, a new node T_4 is created and linked as a child of node T_3. Thereafter, $PS(P_2^+)$ and $|T(P_2^+)|$ are attached to the last node T_4. The third, and fourth patterns are inserted in a manner similar to the first and second patterns. To facilitate tree traversal, header table is built in which each term points to its occurrence in the tree via a *Node-link*. Nodes with the same *term-name* are linked in sequence via such *node-links*. Finally, PTP_u^+ for user *u* is constructed as shown in Fig. 2(left). Likewise, PTP_u^- in Table 1(right) is constructed as shown in Fig. 2(right).

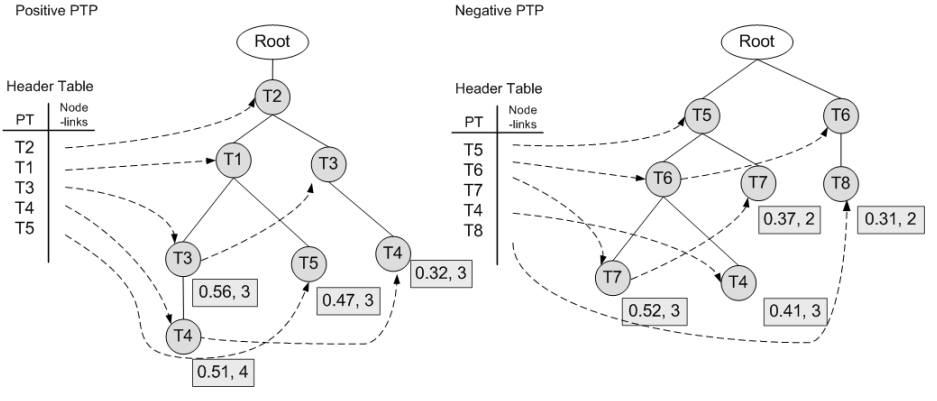

Fig. 2. A tree structure of M_u for personalized term patterns in Table 1

3 Content-Based Filtering for Personalized Service

In this paper, the filtering approach considers matched patterns between the new contents and **PTP** for a user. In addition, we consider both positive feedback and negative feedback for judging whether content are relevant or irrelevant to the user.

Definition 4 (Matched Pattern). Let $T(P_k)$ be a set of terms contained in pattern P_k such that $P_k \in PTP_u$, If all terms in contained P_k appear new content C_n, $T(P_k) \subseteq C_n$, then pattern P_k is deemed a *matched pattern* between P_k and content C_n.

The positive similarity, a measure of how positively the content is relevant to the user, between new content C_n and positive PTP of user u is defined in equation (1).

$$pos(u, C_n) = \frac{|MP^+|}{|PTP_u^+|} \times \sum_{P_k \in MP^+} |T(P_k)| \cdot PS^+(P_k) \tag{1}$$

where PTP_u^+ is a set of positive personalized patterns for user u, $T(P_k)$ is a set of terms contained in pattern P_k, and MP^+ is a set of matched patterns between PTP_u^+ and content C_n. $PS^+(P_k)$ refers to the positive support value of matched pattern P_k. The higher the similarity value, the more relevant the content is to the user.

Likewise, the negative similarity, a measure of how the content is irrelevant to the user, between content C_n and negative PTP of user u is defined in equation (2). However, as opposed to the positive similarity, the content which has the highest value of the negative similarity is the most irrelevant to the user.

$$neg(u, C_n) = \frac{|MP^-|}{|PTP_u^-|} \times \sum_{P_k \in MP^-} |T(P_k)| \cdot PS^-(P_k) \tag{2}$$

The main concept of the similarity schemes dictates that specific patterns (positive or negative) with numerous occurrences in user preference (positive or negative) present a greater contribution with regard to similarity than general patterns with a smaller number of occurrences. Finally, the combined similarity between user u and content C_n is obtained as the following.

$$Sim(u, C_n) = \alpha \cdot pos(u, C_n) + (\alpha - 1) \cdot neg(u, C_n) \tag{3}$$

where α is a parameter in [0,1] which specifies for adjusting the relative weighting between the positive similarity and the negative similarity. If α=0 then $Sim(u,C_n)$ just takes $neg(u,C_n)$ into account whereas if α=1 then $Sim(u,C_n)$ just coincides with $pos(u,C_n)$. Given two contents C_i and C_j, content C_i is of more interest to user u than content C_j if and only if a similarity between user u and content C_i is higher than that of content C_j, $sim(u,C_i) > sim(u, C_j)$.

Once the scores between user u and new contents are computed, the contents are sorted in order of descending score value. Thereafter, a set of N rank contents that have obtained higher similarity values are identified for user u, and then those contents are recommended to user u (Top-N recommendation) [8].

4 Experimental Evaluation

In this section, we present the quality evaluation of the proposed approach with experimental details. The experiment data is taken from NSF (National Science Foundation) research award abstracts [16]. The original data set contains 129,000 abstracts describing NSF awards for basic research from 1900 to 2003. However, the data is too large to be used for experiments, and thus we selected the award abstracts from

2000 to 2003, i.e. the selected data set contained 30,384 abstracts and 22,236 distinct terms as obtained from the abstracts. 10 users were participated in the experiments by providing a positive and negative feedback according to their interests from the total contents (30,384 contents). Whenever they found the content related to their preferences, they added that content into their positive list or negative list. To evaluate the performance, we divided the collected positive contents of the users into *a test set* with exactly 100 contents per user and *a training set* with the remaining contents. A model $\mathbf{M}_u$ of each user was constructed using only the *training set*. Thereafter, we computed the similarity scores of contents except the content list of a given user in the training set and subsequently identified a set of N rank contents that obtained the higher scores.

The performance was measured by looking at the number of *hits*, and their *ranking* within the *top*-N contents that were recommended by a particular scheme. We computed the quality measures that are defined as follows.

Hit Rate (HR). In the context of *top*-N recommendations, *hit-rate*, a measure of how often a list of recommendations contains contents that the user is actually interested in, was used for the evaluation metric [5, 8]. The *hit-rate* for user u is defined as:

$$HR(u) = \frac{\left| Test_u \cap TopN_u \right|}{\left| Test_u \right|}$$

where $Test_u$ is the content list of user u in the test data and $TopN_u$ is a *top*-N recommended content list for user u. Finally, the overall *HR* of the *top*-N recommendation for all users is computed by averaging these personal *HR* in test data.

Reciprocal Hit Rank (RHR). One limitation of the *hit-rate* measure is that it treats all hits equally regardless of the ranking of recommended contents. In other words, content that is recommended with a top ranking is treated equally with content that is recommended with an Nth ranking [8]. To address this limitation, therefore, we adopted *the reciprocal hit-rank* metric described in [8]. The *reciprocal hit-rank* for user u is defined as:

$$RHR(u) = \sum_{C_n \in (Test_u \cap TopN_u)} \frac{1}{rank(C_n)}$$

where $rank(C_n)$ refers to a recommended ranking of content C_n within the *hit set* of user u. That is, hit contents that appear earlier in the *top*-N list are given more weight than hit contents that occur later in the list. Finally, the overall *RHR* for all users is computed by averaging the personal *RHR(u)* in test data. The higher the *RHR*, the more accurately the algorithm recommends contents.

4.1 Experimental Results

The performance evaluation is divided into two dimensions. In the first experiment, we determine the *minimum support* that controls the size of $\mathbf{M}_u$ and the parameter α that blends two similarity (positive and negative) measures. The second experiment presents successful performance of our method for a content relevant personalized recommendation in comparison with other approaches. In order to compare the performance of the proposed scheme, a probabilistic learning algorithm, which applies a

naïve Bayesian classifier (denoted as *NB*) [3, 4], and a TF-IDF vector-based algorithm, which is employed in the *Webmate* system (denoted as *Webmate*) [2], were implemented. To make the comparison fair, both of the algorithms were designed to learn users' preferences from positive examples and negative examples. Because *Webmate* was not originally designed to learn from negative examples, the learning of negative examples is performed by subtracting the feature vector of a learned content from the profile [5]. For the content filtering process, in the case of *NB*, the probability of new content belonging to the "interest (positive)" class of a user divided by the probability of the content belonging to the "no interest" (negative) class is used. In the case of *Webmate*, contents are ranked using the calculated cosine similarity between contents and a user profile. The *top*-N recommendation of our strategy was then evaluated in comparison with the benchmark algorithms.

4.1.1 Experiments with α Value

First of all, we considered about two significant factors affecting the quality of our algorithm, which are minimum support (*min_sup*) and α value. A high *min_sup* discards more patterns, and thus remaining term patterns may not be sufficient to represent user preference. In contrast, a low *min_sup* may include many noise patterns. The other factor, parameter α, determines which will be given more weight, a positive similarity or a negative one. For our main comparisons with existing works, we empirically determined these two values which showed the most reasonable performance in both evaluation metrics, HR and RHR. *min_sup* values used for mining personalized term patterns were 5%, 8%, 10%, and 20%. In addition, we varied α value from 0 to 1 in an increment of 0.2. In this experiment we set the value of N=100 as the number of recommended contents and $\theta = 0.5$ as the pattern weight threshold.

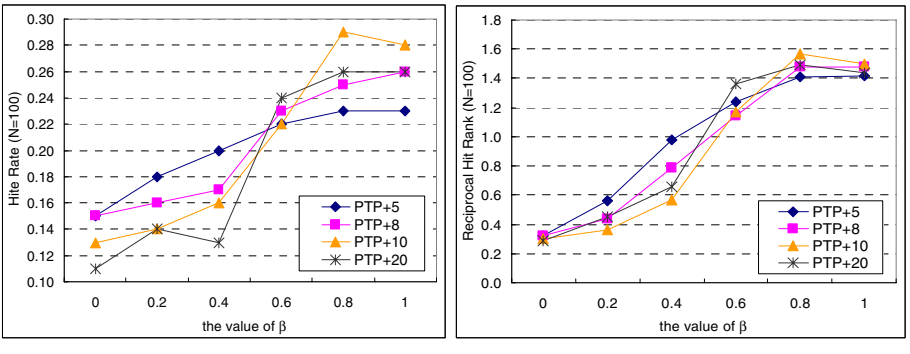

Fig. 3. Hit rate (HR) and reciprocal hit rank (RHR) according to variation of α value

Fig. 3 presents a variation of HR (left) and RHR (right), by changing the α value. We describe the lines as PTP + k where k means *min_sup* of 5%, 8%, 10%, and 20%. It can be observed from the graph that the parameter α affected the performance and overall performance was improved with the growth of α except for a few cases. Generally, with respect to HR, a low *min_sup* levels (i.e., 5%, 8%) showed better quality than a high *min_sup* levels (i.e., 10%, 20%) when α was close to 0 (negative similarity weighted).

On the contrary, high *min_sup* levels performed better (positive similarity weighted) when α was close to 1. When we compare the results of RHR, the four cases demonstrate similar types of charts and elevate RHR as the α value increases from 0.0 to 0.8; beyond this point, RHR deteriorates slightly. For example, when α is set to 0.8, *PTP+10* yields a RTR of 1.57, which is the best value, whereas it gives a RTR of 1.50 in the case of α=1. Roughly speaking, considering the positive similarity rather than the negative one between a user and contents might reflect user's preference better. We conclude from this experiment that the fusion of the positive and negative similarity for a content filtering is effective in terms of improving the performance, compared to the positive similarity or the negative similarity only.

4.1.2 Comparison with Other Methods

To experimentally compare the performance of our algorithm, we calculated the hit rate (HR) and the reciprocal hit rank (RHR) achieved by *PTP*, *Webmate*, and *NB* when the number of recommended contents *N* was 100 and 200. According to the results of the prior experiments in section 4.1.1, *min_sup* and α value were set to 10% and 0.8, respectively.

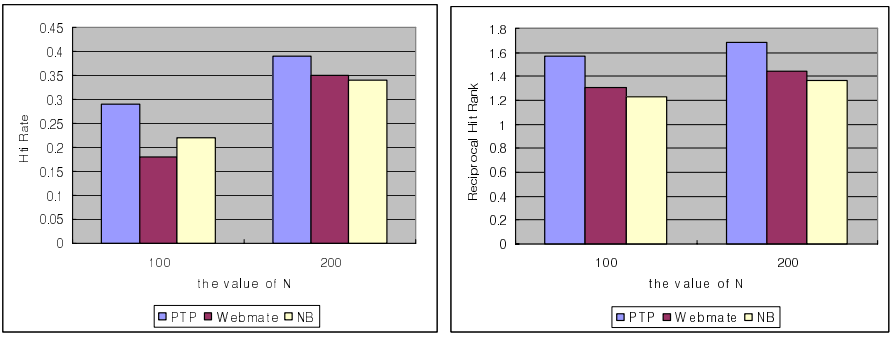

Fig. 4. Comparison of the hit rate (HR) and the reciprocal hit rank (RHR)

Fig. 4 depicts the HR (left) and the RHR (right). In general, with the growth of recommended contents *N*, HR, and RHR tend increase. However, overall HR performance of all three methods did not meet with good results throughout our experiments due to the huge size of total data set. Even though data set used for learning user preferences (a training set) was excluded from total data set, the number of recommended contents was less than 0.01% of total contents. Although HR for all algorithms is unsatisfactorily low, *PTP* provided considerably improved HR on all occasions compared to the benchmark algorithms. Similar conclusions can be made by looking at the RHR results as well. For example, when *N* is 100, *PTP* achieves 19% and 27% improvement in terms of RHR, compared to *Webmate* and *NB*, respectively. These results show that our algorithm can recommend contents at higher ranks for each user as well as it can recommend more accurate contents than the other two methods.

5 Related Work

This section briefly explains previous studies related to user modeling and personalized recommendation. Two approaches for recommender systems have been discussed in the literature, *i.e.*, a content-based filtering approach and a collaborative filtering approach [15]. Our research mainly focuses on the content-based filtering for personalized recommendations. Content-based approaches analyze information object of a user, usually textual contents, and build a model of personal preferences based on the features of the object. *Webmate* tracks user interests from his positive information only (i.e., documents that the user is interested in) and exploits the vector space model using TF-IDF method [2]. A classification approach has been explored to recommend articles relevant user profile, such as *NewsDude* and *ELFI* [3, 4]. In *NewsDude*, two types of the user interests are used: short-term interests and long-term interests. To avoid recommendations of very similar documents, short-term profile is used. For the long-term interests of a user, the probabilities of a document are calculated using Naïve Bayes to classify a document as interesting or not interesting. Instead of learning from users' explicit information, *PVA* learns a user profile implicitly without user intervention, such as relevance feedback, and represents it as keyword vector in the form of a hierarchical category structure [6] as similar to *Alipes* [5]. In *Newsjunkie*, novelty-analysis algorithm is employed to present novel information for users by identifying novelty of articles in the contexts of articles they have already reviewed [10]. Likewise our research, Lops et al. exploits user profiles consisting of two parts, the positive part for modeling user interests and the negative part for user disinterests [15]. Although these systems have their own method to build a user model, they do not deliberate on concurrence of terms and offer the ability to identify meaningful or useful patterns, which are important features for representing articles or contents [11].

6 Discussion and Conclusions

In the present work, we have presented a new method for modeling and learning user profiles that discriminates interesting information from uninteresting data. The major advantage of our proposed learning method is that it supports the identification of useful patterns of each user. In addition, mining from the contents of user interest (positive) and disinterest (negative), user models could identify disinterest patterns as well as interest patterns. In order to evaluate our work, we compare our experimental results with those of probabilistic learning model and vector space model. The experimental results demonstrate that the proposed method offers significant advantages in terms of improving recommendation quality.

Nevertheless, there remain some research questions. It remains to be evaluated, how different a threshold of pattern weight, θ, affects the learning result. Another research question is how to consider the changes of user interests efficiently. Once user models are built, it is difficult to reflect a new user feedback. Incremental learning is one of the interesting issues that we plan to consider for addressing this problem in the future.

References

1. Salton, G., Buckley, C.: Term Weighting Approaches in Automatic Text Retrieval. Information Processing and Management 24, 513–523 (1988)
2. Chen, L., Sycara, K.: WebMate: Personal Agent for Browsing and Searching. In: Proc. of the 2nd Int. Conf. on Autonomous Agents and Multi Agent Systems, pp. 132–139 (1998)
3. Billsus, D., Pazzani, M.J.: A hybrid user model for News story classification. In: Proc. of the 7th Int. Conf. on User Modeling, pp. 99–108 (1999)
4. Schwab, I., Pohl, W., Koychev, I.: Learning to Recommend from Positive Evidence. In: Proc. of Int. Conf. on Intelligent User Interfaces (2000)
5. Widyantoro, D.H., Ioerger, T., Yen, J.: Learning User Interest Dynamics with a Three-Descriptor Representation. Journal of the American Society for Information Science and Technology 52, 212–225 (2001)
6. Chen, C.C., Chen, M.C., Sun, Y.: PVA: A Self-Adaptive Personal View Agent. Journal of Intelligent Information Systems 18, 173–194 (2002)
7. Eirinaki, M., Vazirgiannis, M.: Web Mining for Web Personalization. ACM Transactions on Internet Technology 3, 1–27 (2003)
8. Deshpande, M., Karypis, G.: Item-based Top-N Recommendation Algorithms. ACM Transactions on Information Systems 22, 143–177 (2004)
9. Han, J., Pei, J., Yin, Y.: Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. Data Mining and Knowledge Discovery 8, 53–87 (2004)
10. Gabrilovich, E., Dumais, S., Horvitz, E.: Newsjunkie: Providing Personalized News-feeds via Analysis of Information Novelty. In: proc. of the 13th Int. Conf. on World Wide Web, pp. 482–490 (2004)
11. Chung, S., McLeod, D.: Dynamic Pattern Mining: An Incremental Data Clustering Approach. In: Spaccapietra, S., Bertino, E., Jajodia, S., King, R., McLeod, D., Orlowska, M.E., Strous, L. (eds.) Journal on Data Semantics II. LNCS, vol. 3360, pp. 85–112. Springer, Heidelberg (2005)
12. Kim, H.N., Kim, H.J., Jo, G.S.: Content-based Document Recommendation in collaborative Peer-to-Peer Network. In: Jin, H., Pan, Y., Xiao, N., Sun, J. (eds.) GCC 2004. LNCS, vol. 3251, pp. 575–582. Springer, Heidelberg (2004)
13. Flesca, S., Greco, S., Tagarelli, A., Zumpano, E.: Mining User Preferences, Page Content and Usage to Personalize Website Navigation. World Wide Web: Internet and Web Information Systems 8, 317–345 (2005)
14. Das, A., Datar, M., Garg, A., Rajaram, S.: Google News Personalization: Scalable Online Collaborative Filtering. In: Proc. of the 16th Int. Conf. on World Wide Web, pp. 271–280 (2007)
15. Lops, P., Degemmis, M., Semeraro, G.: Improving Social Filtering Techniques Through WordNet-Based User Profiles. In: Proc. of the 11th Int. Conf. on User Modeling, pp. 268–277 (2007)
16. Pazzani, M.J., Meyers, A.: NSF Research Awards Abstracts 1990-2003, http://kdd.ics.uci.edu/databases/nsfabs/nsfawards.html