Refining Preference-Based Search Results Through Bayesian Filtering

Jiyong Zhang
Human Computer Interaction Group
Swiss Federal Institute of Technology (EPFL)
CH-1015, Lausanne, Switzerland
+41 21 6931246
jiyong.zhang@epfl.ch

Pearl Pu
Human Computer Interaction Group
Swiss Federal Institute of Technology (EPFL)
CH-1015, Lausanne, Switzerland
+41 21 6936081
pearl.pu@epfl.ch

ABSTRACT

Preference-based search (PBS) is a popular approach for helping consumers find their desired items from online catalogs. Currently most PBS tools generate search results by a certain set of criteria based on preferences elicited from the current user during the interaction session. Due to the incompleteness and uncertainty of the user's preferences, the search results are often inaccurate and may contain items that the user has no desire to select. In this paper we develop an efficient Bayesian filter based on a group of users' past choice behavior and use it to refine the search results by filtering out items which are unlikely to be selected by the user. Our preliminary experiment shows that our approach is highly promising in generating more accurate search results and saving user's interaction effort.

ACM Classification: H.1.2[Models and Principals]: User/ Machine Systems–human factors; H.3.3[Information Search and Retrieval]: Information filtering–Selection process.

General terms: Algorithms, Human Factors, Performance.

Keywords: preference-based search, Bayesian filtering, interaction effort.

INTRODUCTION

Preference-based search (PBS) is a popular approach that has been applied in many e-commerce websites for helping consumers find their desired items. In a typical interaction cycle, a PBS tool generates several candidate items as search results and invites users to refine the search by critiquing these items as feedback to influence the system in generating the next set of search results. This interaction process continues for multiple times until users find their desired targets. The main advantage of PBS is that users can be stimulated to reveal preferences gradually by some concrete examples. PBS is in line with behavioral decision theory that users construct their preferences adaptively in light of options shown to them [5] and has been implemented in many applications [2, 6, 9, 12].

IUI'07, January 28–31, 2007, Honolulu, Hawaii, USA..

One key problem for PBS is to generate desirable search results for users to review. Currently most PBS tools generate the search results by some ranking criteria such as utility [12], similarity or diversity [10] based on the user's current preferences. The search results determined by such criteria are not always consistent with user's target choice because of the following facts. Firstly, users' preferences are often not revealed completely during the interaction process. Secondly, these criteria may not fully represent user's actual preference order on the displayed items. For instance, when ranking the items by utility, the utility function is often given in a simple form to avoid complicated preference elicitation process. Consequently, the items having the highest ranking scores are not always those that the users want to select in real situations. In another word, the search results determined by such criteria may contain some items that users have less (or no) desire to select.

In this paper we propose an approach of refining PBS results by filtering out items that are less likely to be selected by the user in each interaction cycle. The intuition behind this approach is that if many users who construct the same preferences didn't select a particular product in the search results, then a new user holding the same preferences will also unlikely select that product. Thus that product should be filtered out from the search results to save the user's effort in processing that information. Specifically, we train a Bayesian filter from a group of users' past choice behavior and apply it to the search results that generated by a certain ranking criteria, and remove those items having very low probabilities of being selected by the user under the current specified preferences.

PBS tools may allow users to input various types of preferences as feedback [11]. In this paper we concentrate on the preference-based feedback on items(or item-base feedback), which has a simple form like "I prefer item A to B". Such preference can be easily obtained by asking the user to select the most desired product among the search results at each interaction cycle. The user does not necessarily have to fully understand the features in the product domain. In this paper we adopt utility values as the ranking criteria and use the approach which generates search results by utility values as the baseline. In our approach, we first generate a list of candidate products according to their utility values, and then we apply the Bayesian filtering technique to remove those products that are not likely to be selected by the user. Our hypothesis is that our approach can save the user's interaction effort substantially.

RELATED WORK

Bayesian Filtering is a well-known technique in classifying data into different classes and has been successfully used in email filtering and anti-spam software [8]. The main point of this approach is to learn the probability of each word occurring in spam email class and in non-spam email class respectively from a set of email samples (labeled with *spam* or *non-spam*). When a new email comes, its probabilities in spam and non-spam classes are computed according to the words it contains, and if the ratio of the spam probability to the non-spam probability exceeds a certain threshold, then the unknown email is marked as spam. The Bayesian filtering approach can discover over 99.99% of spam emails. Inspired by this idea, we apply this technique to refine PBS results by filtering out items that are not likely to be selected by the user. We believe that by applying such filter the search results are more attractive and are more likely to be selected by users as their desired products.

PBS tools had been developed in many applications with various ranking criteria. Stolze [12] proposed the scoring tree method for building interactive e-commerce system based on utility values. The SmartClient approach [6] uses both utility and constraint problem solving techniques to allow users build and refine their preference models in the domain of travel planning. In [4] different variant of similarity criteria were studied in developing the preference-based recommendation system. Diversity as a ranking criteria is also investigated to improve the efficiency of case-based recommender systems [10]. Lately Price [7] proposed an approach of optimizing recommendation set to cover the maximal uncertainty over the user's all possible preferences. In our approach, We not only rank items according to the preferences specified by the current user, but also consider the possibility of items being selected from other users' past choice behavior under the same preference context. Moreover, the Bayesian filtering is a data-driven approach which has the potential to systematically increase the quality of search results if enough train data are available.

In this paper we use the well-known multi-attribute utility theory (MAUT) as the ranking criterion for PBS [3]. Such theory has been used in representing user's preferences in many earlier work [1, 12]. In this paper we only consider the simplified weighted additive form of the utility function. Very briefly, a weighted additive form of utility function is commonly adopted to calculate the utility of a product $X = \langle x_1, x_2, ..., x_n \rangle$ as follows:

$$U(\langle x_1, \cdots, x_n \rangle) = \sum_{i=1}^{n} w_i V_i(x_i) \quad (1)$$

where n is the number of attributes that the products may have, the weight $w_i (1 \leq i \leq n)$ is the importance of the attribute x_i, and V_i is a value function of the attribute x_i which can be given according to the domain knowledge and the user's current preferences.

BAYESIAN FILTER FOR PREFERENCE-BASED SEARCH

In a typical interaction cycle of PBS, the system proposes a list of candidate products as search results for the user to choose, and the user selects the most desired one among them. In other words, each time the candidate products will be classified into two classes according to the user's action: *selected* (s) and *not selected* ($\neg s$). The selected product will be put into class s, and the others will be put into class $\neg s$ automatically. Also, we could gather users' past interaction records as train data sets for these two classes.

For a given candidate product $X = \langle x_1, x_2, \cdots, x_n \rangle$, one important fact is that the probability of being selected by the user not only depends on the product itself, but also depends on the user's current preferences. While the user's current preferences are difficult to capture, here we use the product that the user has selected in the previous interaction cycle to approximate them. We call the previously selected product the reference product and denote it as $X^o = \langle x_1^o, x_2^o, \cdots, x_n^o \rangle$. Thus whether a product X would be selected depends on the reference product X^o and the product itself.

Formally, for a given product X with the reference X^o (we denote it as an ordered pair $\langle X, X^o \rangle$), the probability that the user would select it can be represented as $P(s|\langle X, X^o \rangle)$. By the Bayesian theorem, we have

$$P(s|\langle X, X^o \rangle) = \frac{P(\langle X, X^o \rangle|s)P(s)}{P(\langle X, X^o \rangle)}, \quad (2)$$

where $P(s)$ is the prior probability of the *selected* class, and $P(\langle X, X^o \rangle|s)$ is the likelihood of the pair $\langle X, X^o \rangle$ belonging to the *selected* class. Similarly, we can also compute the probability of a product X with the reference product X^o being classified in the $\neg s$ class: $P(\neg s|\langle X, X^o \rangle)$.

When a new product X is shown with the reference product X^o, the ratio ρ of probability in *selected* class to probability in *not selected* class can be computed as the following:

$$\rho(\langle X, X^o \rangle) = \frac{P(s|\langle X, X^o \rangle)}{P(\neg s|\langle X, X^o \rangle)} = \frac{P(s)P(\langle X, X^o \rangle|s)}{P(\neg s)P(\langle X, X^o \rangle|\neg s)}. \quad (3)$$

The ratio ρ can be used as the indicator of the Bayesian classifier. We define the Bayesian filter with the following rule:

if $\rho(\langle X, X^o \rangle) < \rho_0$ **then**
$X \in$ class $\neg s$ and filter X out of the results (4)
else $X \in$ class s

where ρ_0 is a constant threshold for the Bayesian filter. In theory ρ_0 should be 1, but in practice we set ρ_0 as a smaller number (such as 0.25) to ensure that the products being filtered out are really those unlikely to be selected by the user.

The prior probabilities of $P(s)$ and $P(\neg s)$ can be estimated directly from the training data set(i.e. $P(s)$ is the percentage of *selected* products to all available products). To estimate

X^o – item selected in the previous interaction cycle (the reference product);
IS – the available item set;
K – the size of items in the final search results;
b – the window size;
ρ_0 – the threshold of the Bayesian filter;

Procedure BayesianFiltering (X^o, IS)
  IS' $\longleftarrow$ sort IS by the utility criteria;
  RS $\longleftarrow$ top $K * b$ items in IS';
  for i:= $K * b$ **to** 1 **do**
    X = the *i-th* item in RS;
    if $\rho(\langle X, X^o \rangle) < \rho_0$ **then**
      remove X from RS;
    if sizeof(*RS*) $\leqslant$ K **then break**;
  RS $\longleftarrow$ top K remaining items in RS;
  return RS;

Figure 1: The procedure of preference-based search with Bayesian filtering.

the likelihood of $P(\langle X, X^o \rangle|s)$ and $P(\langle X, X^o \rangle|\neg s)$, one intuitive way is to count the number of pair$\langle X, X^o \rangle$ in *selected* class and *not selected* classes respectively during the past observations. However, in reality an online catalog may contain a huge number of items for users to select, and most likely the training data set would be too sparse to estimate the likelihood for all product pairs directly.

To solve this data sparsity problem, here we assume that all the attributes in the product domain are mutually independent. Applying the naive Bayesian rule, we have the following equations:

$$P(\langle X, X^o \rangle|s) = \prod_{i=1}^{n} P(\langle x_i, x_i^o \rangle|s), \quad (5)$$

and

$$P(\langle X, X^o \rangle|\neg s) = \prod_{i=1}^{n} P(\langle x_i, x_i^o \rangle|\neg s). \quad (6)$$

where $P(\langle x_i, x_i^o \rangle|s)$ and $P(\langle x_i, x_i^o \rangle|\neg s)$ $(1 \leq i \leq n)$ are the likelihoods of the attribute pair $\langle x_i, x_i^o \rangle$ being selected or not, which could have enough data for estimation from the dataset. Thus when data sparsity occurs, we compute the likelihoods in equation 3 by equation 5 and 6 instead of direct estimations.

REFINING THE SEARCH RESULTS

Once the Bayesian filter is built, we apply it to the search results generated by some underlying ranking criteria that we mentioned earlier. In this paper we use the equation 1 to compute the utility values and rank products in descending order. Instead of recommending the top K products as search results directly, we first select a larger size for the recommendation set (say $K * b$ items, where b is the window size), and then apply the Bayesian filter (equation 4) to these products one by one in ascending order. If there are no more than K items left in the recommendation set, this filter process will stop and return the current recommendation set. If there are still more than K products left in the recommendation set after the filtering process, we will only show the top K products with highest utility values as the search results. Thus there are still K products left in the search results in any case. This filtering procedure is shown in Figure 1.

EXPERIMENT AND RESULTS

We carried out a simulation experiment to compare the performance of the standard PBS based on the utility ranking criteria (denoted as $MAUT$) and our proposed approach of combining utility with Bayesian filter(denoted as $MAUT + BF$). We simulate the scenario that various online users with different preferences are trying to find their ideal products from a website. To simulate the variety of users' preferences, in each simulated interaction process we appoint a product from the dataset as the target product of an artificial user and the procedure is to help her find that target product out from the dataset through interactive PBS. In our previous study involving real users, we observed that an average user states about 3 initial preferences and gradually adds/revises preferences during interaction process [13]. Therefore, we randomly determine the number of the initial preferences from 1 to 5. During the interaction process we assume that the artificial user will always choose the product that is closest to the target product from the search result. In the training period, each product in the dataset is appointed as the target choice for 10 times and we use the MAUT approach to generate the required training data. We then learn the occurrence frequency of both item pairs and attribute pairs to construct the Bayesian filter. In this process we train one Bayesian filter for all users without considering their differences or their final target choices. In the test period, each product is appointed as the target choice another 2 times and the number of interaction cycles for finding the target choice are recorded. In this simulation experiment we use the PC data set [4] which contains 120 PCs with 8 different attributes. This data set is available at *http://www.cs.ucd.ie/staff/lmcginty/PCdataset.zip*.

In our simulation experiment, in each interaction cycle the search results contain 5 products(i.e. $K = 5$), and we set the filter threshold ρ_0 to 0.25, and the window size b to 3. Also, since the product gaining the maximal utility value is the best one matching the user's preference, we will always keep the best product in the search results without filtering it out. Moreover, as the first interaction cycle doesn't have the reference product X^o, the Bayesian filter is not applied in the first interaction cycle.

The user's interaction effort of a PBS tool can be measured by the length of the interaction cycles that the user has to take for reaching the final target. In our experiment we counted the length of each interaction cycle for both approaches and the results are shown in Figure 2. Users can reach the final target with nearly 5 cycles in average by using the baseline MAUT approach which decides the search results only by their utility values. When applying the Bayesian filtering to refine the search results, the interaction cycles can be reduced to 4.15 cycles, which is a 16.5% saving of the interaction effort and the difference is significant(t-test p-value: < 0.001). This results support our earlier hypothesis that the Bayesian

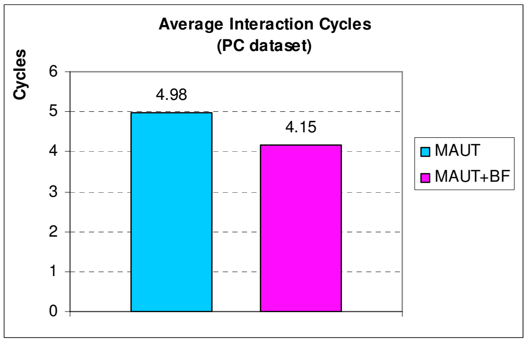

Figure 2: The average interaction cycles

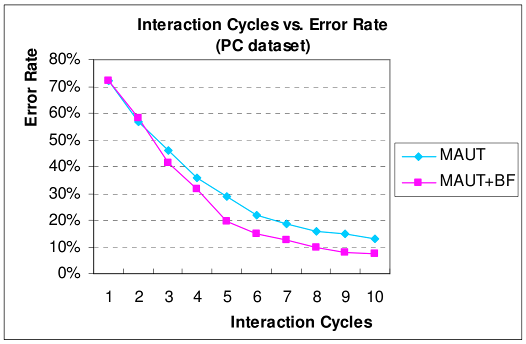

Figure 3: The search error rate of finding the target choice within given number of interaction cycles.

filtering method can make a PBS tool more efficient by saving the user's interaction effort.

We define the search *error rate* of a PBS tool as the percentage of times that a user couldn't find the target product within a certain number of interaction cycles. The results are shown in Figure 3. When users carry on more than 2 interaction cycles, we can see that the MAUT+BF approach has a lower search error rate than the baseline MAUT approach. For example, when a user is willing to carry on 5 interaction cycles with the PBS tool, the search error rate can be reduced from 29% to 20% when the Bayesian filter is applied.

CONCLUSIONS AND FUTURE WORK

In this paper we proposed an approach of enhancing the quality of the preference-base search results through Bayesian filtering. The main contribution of this work is that we provided a data-driven mechanism to improve the recommendation quality systematically through other user's past interaction behavior. Our preliminary simulation experiment results show that the Bayesian filtering can substantially make a PBS tool more efficient in helping users find their target products. In the future, we will carry out a large-scale real user studies to verify the efficiency of this approach. We also plan to improve this approach by developing algorithms to train the Bayesian filter incrementally so that the filter can be updated online during each interaction session. Moreover, we will classify all users in different user groups according to their interaction behavior, and train Bayesian filters adaptively to the specific user.

ACKNOWLEDGMENTS

Funding for this research was provided by Swiss National Science Foundation under grant 200020-111888.

REFERENCES

1. V. Ha and P. Haddawy. Problem-focused incremental elicitation of multi-attribute utility models. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 215–222, 1997.

2. U. Junker. Preference-based search and multi-criteria optimization. In *Proceedings of the Eighteenth national conference on Artificial intelligence*, pages 34–40, 2002.

3. R.L. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs.* John Wiley and Sons, New York, 1976.

4. L. McGinty and B. Smyth. Comparison-based recommendation. In *Proceedings of ECCBR'2002*, volume 2416 of *Lecture Notes in Computer Science*, pages 575–589. Springer, 2002.

5. J.W. Payne, J.R. Bettman, and E.J. Johnson. *The Adaptive Decision Maker*. Cambridge University Press, 1993.

6. P. Pu and B. Faltings. Enriching buyers' experiences: the smartclient approach. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 289–296. ACM Press, New York, 2000.

7. R. Price and P.R. Messinger. Optimal recommendation sets: Covering uncertainty over user preferences. In *Proceedings of AAAI'2005*, pages 541–548, 2005.

8. M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to filtering junk E-mail. In *Learning for Text Categorization, AAAI Technical Report WS-98-05*, 1998.

9. S. Shearin and H. Lieberman. Intelligent profiling by example. In *Proceedings of the Conference on Intelligent User Interfaces*, pages 145–151. ACM Press, New York, 2001.

10. B. Smyth and P. McClave. Similarity vs. diversity. In *ICCBR'2001*, volume 2080 of *Lecture Notes in Computer Science*, pages 347–361. Springer, 2001.

11. B. Smyth and L. McGinty. An analysis of feedback strategies in conversational recommenders. In *Proceedings of the 14th Irish International Conference on Artificial Intelligence and Cognitive Science*, 2003.

12. M. Stolze. Soft navigation in electronic product catalogs. *International Journal on Digital Libraries*, 3(1):60–66, July 2000.

13. J. Zhang, P. Pu, and P. Viappiani. A study of user's online decision making behavior. Technical report, Swiss Federal Institute of Technology (EPFL), Lausanne (Switzerland), 2006.