Hybrid Collaborative Filtering and Content-Based Filtering for Improved Recommender System

Kyung-Yong Jung[1], Dong-Hyun Park[2], and Jung-Hyun Lee[3]

[1] HCI Lab., Department of Computer Science & Engineering,
[2] Department of Industrial Engineering,
[3] Department of Computer Science & Engineering,
Inha University, Korea
kyjung@gcgc.ac.kr, {dhpark,jhlee}@inha.ac.kr

Abstract. The growth of the Internet has resulted in an increasing need for personalized information systems. The paper describes an autonomous agent, WebBot: Web Robot Agent, which integrates with the web and acts as a personal recommender system that cooperates with the user on identifying interesting pages. Hybrid components from collaborative filtering and content-based filtering, a hybrid recommender system can overcome traditional shortcomings. In this paper, we present an effective hybrid collaborative filtering and content-based filtering for improved recommender system. Experimental results indicate the hybrid collaborative filtering and content-based filtering better than collaborative, content-based, and combined filtering approach.

1 Introduction

The world wild web hypertext system is a very large distributed digital information space. Some estimates suggested that the web included about 160 million pages and this number double every four months. As more information becomes available, it becomes increasingly difficult to search for information without specialized aides.

Recommender systems are designed to predict user preferences using features of the items and ratings given by other users. To be effective, a recommender system must deal with well with two fundamental problems. First, the sparse rating problem; the number of ratings already obtained is very small compared to the number of ratings that need to be predicted. Effective generation from a small number of examples is thus important. This problem is particularly severe during the startup phase of the system when the number of users is small. Second, the first-rater problem; an item cannot be recommended unless a user has rated it before. This problem applies to new items and also obscure items and is particularly detrimental to users with eclectic tastes. [1,2,4,5,7,9,11] are presented as solutions to the aforementioned problems by using both collaborative filtering and content-based filtering method. LSI [11] and SVD [2] classification are used to decrease the number of dimensions in the matrix to solve the sparse rating problem in collaborative filtering, yet they fail to fix the first-

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3036, pp. 295–302, 2004.

rater problem. [1,4,5] solve the first-rater problem, yet fail to fix the sparse rating problem. In an attempt to find a solution to both the sparse rating problem and the first-rater problem, method [9] was implemented. We overcome these drawbacks of collaborative filtering systems, by exploiting content information of the items already rated. Our basic approach uses content-based filtering to convert a sparse user ratings matrix into a full ratings matrix; and then uses collaborative filtering to provide recommendations. We present the framework for hybrid filtering. We apply this framework in the domain of movie recommendation and show that our approach performs significantly better than both collaborative and content-based filtering.

2 Collaborative Filtering and Content Based Filtering

2.1 Collaborative Filtering

Collaborative filtering systems recommend objects for a target user based on the opinions of other users by considering how much the target user and another users have agreed on other objects in the past [4,5]. Collaborative filtering technique predicts the rating of a particular user u for an item i. And it compares the predicted rating with the rating of all other users who have rated the item i. Then a weighted average of the other users rating is used as a prediction. If I_u is set of items that a user u has rated then we can define the mean rating of user u by Equation (1).

$$w(u,a) = \frac{\sum_{i \in I_u \cap I_a} (r_{u,i} - \overline{r_u})(r_{a,i} - \overline{r_a})}{\sqrt{\sum_{i \in I_u \cap I_a} (r_{u,i} - \overline{r_u})^2 \bullet \sum_{i \in I_u \cap I_a} (r_{a,i} - \overline{r_a})^2}} \qquad \overline{r_u} = \frac{1}{|I_u|} \sum_{i \in I_u} r_{u,i} \tag{1}$$

Collaborative filtering algorithms predict the rating based on the rating of similar users. When Pearson correlation coefficient is used, similarity is determined from the correlation of the rating vectors of user u and another users a. It can be noted that $w \in [-1, +1]$. The value of w measures the similarity between the two users' rating vectors. A high value close to +1 signifies high similarity and a low value close to 0 signifies low correlation (not much can be deduced) and a value close to –1 signifies that users are often of opposite opinion. The general prediction formula is based on the assumption that the prediction is a weighted average of the other users' rating. The weights refer to the amount of similarity between the user u and the other users by Equation (2). U_i represents the users who rated item i.

$$p^{collab}(u,i) = \overline{r_u} + \frac{1}{\sum_{a \in U_i} w(u,a)} \sum_{a \in U_i} w(u,a)(r_{a,i} - \overline{r_a}) \tag{2}$$

It is common for the active user to have highly correlated neighbors that are based on very few co-rated item (overlapping; $I_u \cap I_a$). These neighbors based on a small number of overlapping item tend to be bad predictor. To devalue the correlation based on few co-rated items, we multiply the correlation by significance weighting factor. If two users have less than 45 co-rated items, we multiply their correlation by a factor

$sg_{a,u}=n/45$, where n is the number of co-rated items. If the number of overlapping items in greater than 45, then we leave the correlation unchanged i.e. $sg_{a,u}=1$.

2.2 Content-Based Filtering

We use a multinomial text model, in which a document is modeled as an ordered sequence of ordered sequence of word events drawn from the same vocabulary, V. The naïve Bayes assumption states that the probability of each word event is dependent on the document class but independent of the word's context and position. For each class c_j, and word, $w_k \in V$, the probability, $P(c_j)$ and $P(w_k|c_j)$ must be estimated from training data. Then the posterior probability of each class given a document D, is computed using naïve Bayesian classifier [8] by Equation (3).

$$p(c_j \mid D) = \frac{P(c_j)}{P(D)} \prod_{i=1}^{|D|} P(a_i \mid c_j) \tag{3}$$

Where a_i is the ith word in the document, and $|D|$ is the length of the document in words. Since for any given document, the prior $P(D)$ is a constant, this factor can be ignored if all that is desired is a rating rather than a probability estimate. A ranking is produced by sorting documents by their odds ratio, $P(c_1|D)/P(c_0|D)$, where c_1 represents the positive class and c_0 represents the negative class. An example is classified as positive if the odds are greater than 1, and negative otherwise. In our case, since movies are represented as a vector of "documents", d_m, one for each feature (where f_m denotes the mth feature), the probability of each word given the category and the feature $P(w_k|c_j, f_m)$, must be estimated and the posterior category probability for a film, F, computed using Equation (4).

$$p(c_j \mid F) = \frac{P(c_j)}{P(F)} \prod_{m=1}^{|f|} \prod_{i=1}^{|d_m|} P(a_{m,i} \mid c_j,\ f_m) \tag{4}$$

Where $|f|$ is the number of features and $a_{m,i}$ is the ith word in the mth feature. The class with the highest posterior probability determines the predicted rating. The *Laplace* smoothing is used to avoid zero probability estimates [8].

3 An Approach Hybrid Collaborative Filtering and Content-Based Filtering

The proposed hybrid filtering transparently creates and maintains user preferences. It assists users by providing both collaborative filtering and content-based filtering, which are updated in real time whenever the user changes his/her current page using any navigation technique. The WebBot uses the URLs provided in the EachMovie dataset to download movie content from IMDb [6]. WebBot keeps track of each individual user and provides that a user online assistance. The assistance includes two lists of recommendations based on two different filtering paradigms: collaborative filtering and content-based filtering. WebBot updates the list each time the user changes his/her current page. Content-based filtering is based on the correlation between the content of the pages and the user preferences. The collaborative filtering is based on a comparison between the user path of navigation and the access patterns of past users. Hy-

brid filtering may eliminate the shortcomings in each approach. By making collaborative filtering, we can deal with any kind of content and explore new domains to find something interesting to the user. By making content-based filtering, we can deal with pages un-seen by others. Fig. 1 is the system overview for hybrid filtering.

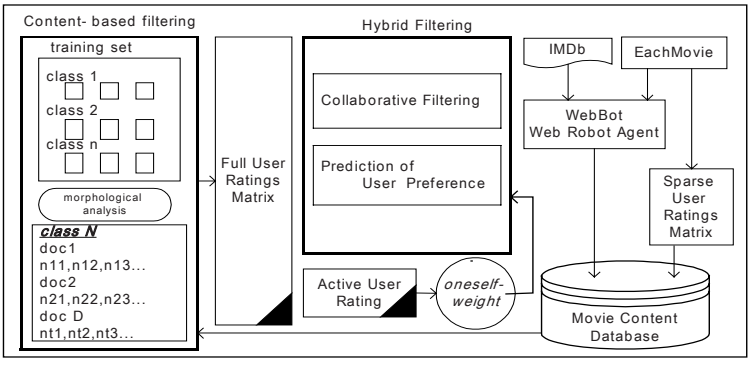

Fig. 1. System overview for hybrid collaborative filtering and content-based filtering

To overcome the problem of stateless connection in HTTP, WebBot follows users through tracking their IP address. To track user presence, a timeout mechanism is used to delete user's session information after a predetermined amount of idle time. So that, a connection after the specified period having the same IP is identified as a new user. This method is fairly easy to implement. Consequently, the IP of a proxy server may represent two or more people who are accessing the same web site simultaneously in their browsing sessions, causing an obvious conflict. However, the reality is that many large sites use this method and have not any clashes. The EachMovie dataset also provides the user-ratings matrix; which is a matrix of users versus items, where each cell is the rating given by a user to an item. We will refer to each row of this matrix as a user-rating vector. The user-ratings matrix is very sparse, because most users have not rated most items. The content-based predictor is trained on each user-ratings vector and a pseudo user-ratings vector is created. A pseudo user-ratings vector contains the user's actual ratings and content-based filtering for the un-rated items. All pseudo user-ratings vector put together from the pseudo ratings matrix, which is a full matrix. Now given an active user's ratings, filtering predictions are made for a new item using collaborative filtering on the full pseudo ratings matrix.

3.1 Extracting Information from Web Robot Agent and Building a Database

Our current prototype system, WebBot: Web Robot Agent uses a database of movie content information extracted from web page at IMDb (`www.imdb.com`). Therefore, the system's current content information about titles consists of textual metadata rather than the actual text of the items themselves. An IMDb subject search is performed to obtain a list of movie-description URLs of broadly relevant titles. WebBot then downloads each of these pages and uses a simple pattern based information extraction system to extract data about each title. Information extraction is the task of

locating specific pieces of information from a document, thereby obtaining useful structured data from unstructured text. A WebBot follows the IMDb link provided for every movie in the EachMovie dataset [6] and collects information from the various links off the main URL. We represent the content information of every movie as a set of features. Each feature is represented simply as a bag of words. IMDb produces the information about related directors and movie titles using collaborative filtering: however, WebBot treats them as additional content about the movie. The text in each feature is then processed into an un-ordered bag of words and the examples represented as a vector of bags of words. Fig. 2 shows WebBot for extracting information, example web page.

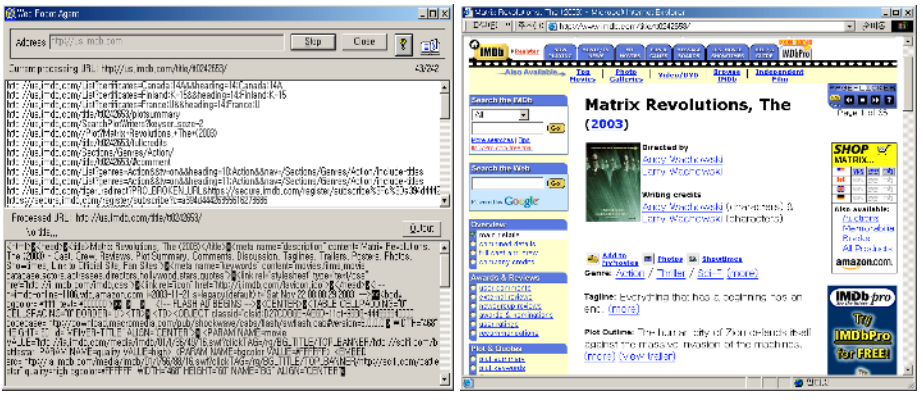

Fig. 2. WebBot: Web Robot Agent, example web page

3.2 Deriving Pseudo User-Ratings Vector from User-Item Matrix

We present now another approach which does not change the collaborative filtering algorithm but instead alters the rating database based on content-based criteria and ratings of real users. This approach was inspired by Sarwar's *rating-bots* approach for the Gouplens news filtering project: for that project software agents which used content-based criteria (spelling and article length) to rate news articles automatically and to increase the amount of ratings in the database [10]. The pseudo user-ratings are content-based filtering for that particular user. This means that some un-rated items are assigned a predicted rating, based on similarity between the rated items and the item for which the rating is missing [7]. We first create a pseudo user-ratings vector for every user u in the database. The pseudo user-ratings vector, v_u, consists of the item ratings provided by the user u, where available, and those predicted by the content-based filtering otherwise. The pseudo user-ratings vectors of all users put together gives the dense pseudo rating matrix V. We now perform collaborative filtering using this dense matrix. The similarity between the active user a and another user u is computed using Pearson correlation described in Equation (1). Instead of the original user votes, we substitute the votes provided by the pseudo user-ratings vectors v_a and v_u.
The accuracy of a pseudo user-ratings vector computed for a user depends on the number of movies he/she has rated. If the user rated many items, the content-based

filtering is good and hence his pseudo user-ratings vector is fairly accurate. On the other hand, if the user rated only a few items, the pseudo user-ratings vector will not be as accurate. We found that inaccuracies in pseudo user-ratings vector often yielded misleadingly high correlations between the active user and other users.

3.3 Prediction of User Preference through Hybrid Filtering

The prediction for the active user is computed as a weighted sum of the mean-centered votes of the *best-n-neighbors* of the user. In our approach, we also add the pseudo active user to the neighborhood. However, we may want to give the pseudo active user more importance than the other neighbors. In other words, we would like to increase the confidence we place in the content-based filtering for the active user. Combining the above two weighting schemes, the final hybrid filtering prediction for the active user a and item i is produced by Equation (5).

$$p_{a,i} = \overline{v_a} + \frac{(c_{a,i} - \overline{v_a}) + \sum_{i=1,u\neq a}^{n} Hybridw_{a,u} w(a,i)(v_{u,i} - \overline{v_u})}{\sum_{i=1,u\neq a}^{n} Hybridw_{a,u} w(a,i)} \tag{5}$$

In above equation $c_{a,i}$ corresponds to the content-based filtering for the active user a and item i. $v_{u,i}$ is the pseudo user-rating for a user u and item i. And $\overline{v_u}$ is the mean over all items for that user. $w(a,i)$ are as shown in Equation (1) respectively; n is the size of neighborhood. The denominator is a normalization factor that ensures all weights sum to one.

4 Performance Evaluation

We used a subset of the EachMovie dataset [6]. This dataset contains 7,291 randomly selected users and 1,628 movies for which content was available from IMDb. To evaluate various approaches of filtering, we divided the rating dataset in test-set and training-set. The rating database is used a subset of the ratings data from the Eachmovie dataset. The training-set is used to predict ratings in the test-set using a commonly used error measure. The metrics for evaluating the accuracy of a prediction algorithm are used mean absolute error(MAE) and rank scoring measure(RSM) [3].

For evaluation, this paper uses the following methods: The proposed hybrid collaborative filtering and content-based filtering (HMW_HF), a collaborative filtering (P_Corr), the recommendation method using only the content-based filtering (Content), and a naïve combined approach (N_Com). The naïve combined approach takes the average of the ratings generated by the collaborative filtering and the content-based filtering. The various methods were used to compare performance by changing the number of clustering users. Also, the proposed method was compared with the previous methods in section 1 that use both collaborative filtering and content-based filtering method by changing the number of user evaluations on items. The aforementioned previous method includes the Soboroff method [11] that solved the sparse rat-

ing problem, the Fab method [1] that solved the first-rater problem, and the Pazzani method [9] that solved both the sparse rating problem and the first-rater problem.

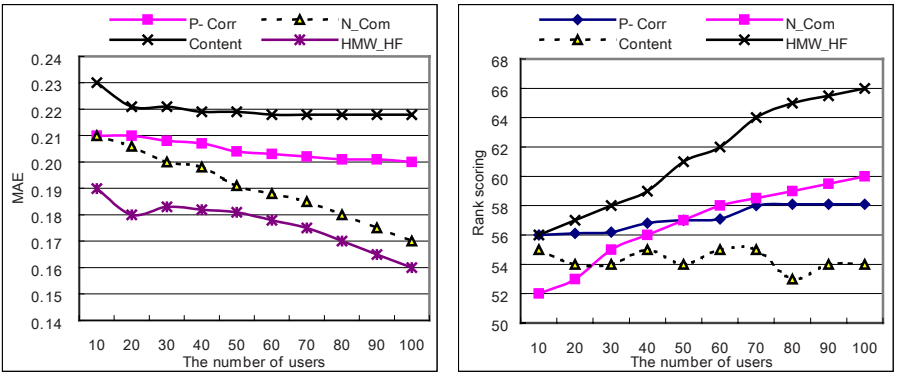

Fig. 3. MAE, Ranking scoring measure at varying the number of users

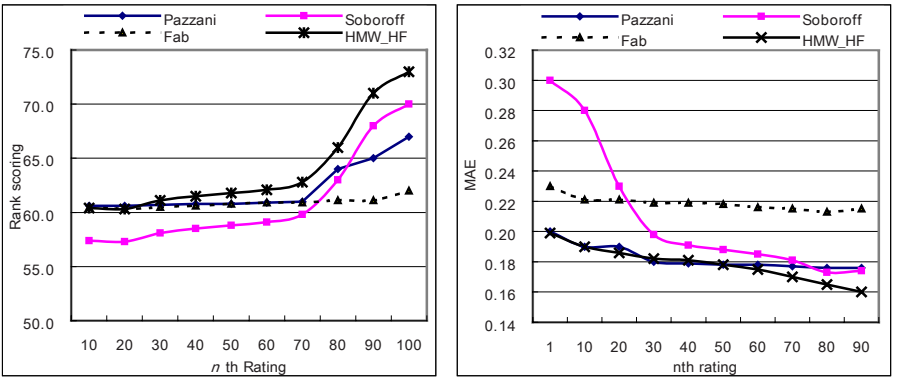

Fig. 4. MAE, Ranking scoring measure at *n*th rating

Fig. 3 shows the MAE and RSM of varying the number of users. Fig. 3, as the number of users increases, the performance of the HMW_HF, and the P_Corr also increases, whereas the method using content shows no notable change in performance. In terms of accuracy of prediction, it is evident that method HMW_HF, which uses both collaborative filtering and content-based filtering, is more superior to method N_Com. Fig. 4 is used to show the MAE and RSM when the number of user's evaluations is increased. In Fig. 4, the Soboroff method, which has the first-rater problem, shows low performance when there are few evaluations; the other methods outperform the Soboroff method. Although the Pazzani method, which solved both the sparse rating problem and the first-rater problem, along with the HMW_HF show high rates of accuracy, the HMW_HF shows the highest accuracy of all methods.

Since we use a pseudo ratings matrix, which is a full matrix, we eliminate the root of the sparse rating problem and the first-rater problem. Pseudo user-ratings vectors

contain ratings for all items; and hence all users will be considered as potential neighbors. This increases the chances of finding similar users. The original user-ratings matrix may contain items that have not been rated by any user. In a collaborative filtering approach these items would be ignored. However in HMW_HF, these items would receive a content-based prediction from all users. Hence these items can now be recommended to the active user, thus overcoming the first-rater problem.

5 Conclusion

Hybrid collaborative filtering and content-based filtering can significantly improve predictions of a recommender system. In this paper, we have shown how hybrid collaborative filtering and content-based filtering performs significantly better than collaborative, content-based, and combined filtering approach. The proposed hybrid filtering exploits content-based filtering within a collaborative framework. It overcomes the disadvantages of both collaborative filtering and content-based filtering, by bolstering collaborative filtering with content and vice versa. Further, due to the nature of the approach, any improvements in collaborative filtering or content-based filtering can be easily exploited to build a powerful improved recommender system.

References

1. M. Balabanovic, Y. Shoham, “Fab: Content-based, Collaborative Recommendation,” Communication of the Association of Computing Machinery, 40(3), pp. 66-72, 1997.
2. D. Billsus, M. J. Pazzani, “Learning Collaborative Information Filters,” In Proc. of the 15th International Conference on Machine Learning, pp. 46-54, 1998.
3. J. S. Breese, et al., “Empirical Analysis of Predictive Algorithms for Collaborative Filtering,” In Proc. of the Conference on Uncertainty in Artificial Intelligence, pp. 43-52, 1998.
4. N. Good, et al., Combining Collaborative Filtering with Personal Agents for Better Recommendations, In Proc. of National Conference on Artificial Intelligence, pp439-446, 1999.
5. W. S. Lee, “Collaborative Learning for Recommender Systems,” In Proc. of the 18th International Conference on Machine Learning, pp. 314-321, 1997.
6. P. McJones, EachMovie dataset, URL: `http://www.research.digital.com/SRC/eachmovie`
7. P. Melville, et al., "Content-Boosted Collaborative Filtering for Improved Recommendations," In Proc. of the National Conference on Artificial Intelligence, pp. 187-192, 2002.
8. T. Mitchell, Machine Learning, McGraw-hill, New York, pp. 154-200, 1997.
9. M. J. Pazzani, “A Framework for Collaborative, Content-based and Demographic Filtering,” Artificial Intelligence Review, 13(5-6), pp. 393-408, 1999.
10. B. M. Sarwar, et al., ”Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System,” In Proc. of the Conference on Computer Supported Cooperative Work, pp. 345-354, 1998.
11. Soboroff, C. Nicholas, “Combining Content and Collaboration in Text Filtering,” In Proc. of the IJCAI’99 Workshop on Machine Learning in Information Filtering, pp. 86-91, 1999.