Combining Dynamic Agents and Collaborative Filtering without Sparsity Rating Problem for Better Recommendation Quality

Saranya Maneeroj, Hideaki Kanai and Katsuya Hakozaki
Graduate School of Information Systems
University of Electro-Communications
1-5-1 Chofugaoka, Chofu, Tokyo 182-8585 Japan

Abstract Information Filtering and Collaborative Filtering techniques have been used for selecting information based on the user's previous preference tendency and the opinion of other people who have similar tastes with the user. Combining both Information Filtering and Collaborative Filtering, or a hybrid systems, have also been proposed to get better recommendation results. In this paper, we present an improved recommendation method that copes with the sparsity problem of the hybrid systems and increases the accuracy of recommendation results. We also present an experimental recommender system for movie, called e-Yawara (extended Yawara), for implementing our method. Evaluation shows that e-Yawara is more efficient and provides more satisfactory result than conventional filtering system.

1. Introduction

In everyday life, we rely on recommendations from other people either by word of mouth, or by recommendation letters, movies and book reviews printed in newspaper and general surveys such as restaurant guides. However, the explosive growth of the Internet has brought us a vast amount of information that any person can hardly digest. To cope with the flood of information, various recommender systems [1] have been created to assist and augment this natural social process, such as GroupLens [2], a recommender system on Netnews community, PHOAKS [3], recommendation of Web resources mined from Usenet news messages, Yawara [4], a documents strolling space based recommender system, MovieLens [5], a movie recommender system created by GroupLens research team, and Ringo [6], a music recommender system. The recommender systems advise users to select information that users may be interested in and filter out what users may not be interested in.

The early recommender systems, such as Yawara system [1], used Information Filtering (IF) techniques as the central role. Information filtering systems use individual previous behavior to produce recommendation. The certain drawback of IF techniques is that they do not provide much in the way of serendipitous discovery.

To cope with drawback of IF techniques, Collaborative Filtering (CF) techniques have been proposed. CF based system recommends items based on the opinion (rating) of other users who have similar tastes. One well-known example of CF techniques is GroupLens system, which has been implemented for filtering Usenet news postings [2]. The limitations of CF techniques are "early-rater" and "sparsity rating" problems. The early-rater problem occurs when a user is the first on the system, hence he/she rates documents without receiving any recommendation. For sparsity rating problem, it occurs when each user has rated a tiny percentage of total number of items, then overlap between user's ratings (or number of co-rated items) is small, or sometimes no overlap occurs. Since CF techniques use co-rated items in finding correlated neighbors for an active user, this sparsity ratings causes recommendation results to be not so accurate and sometimes cannot be produced .

The next level of recommender system is hybrid system. Hybrid systems combine IF and CF techniques in an effort to overcome the limitations of each, such as Movie Lens system [5], a movie recommender system created by the Group Lens research team. It adds filterbots (IF agent) into CF system. Filterbots are rating robots that participate as members of a CF system, they help users who agree with them by providing more ratings upon which recommendations could be made. Although current hybrid systems can solve lacking of serendipitous discovery problem in IF techniques and early-rater problem in CF techniques, sparsity rating problem still remains on the system. This is because most of the current hybrid systems still use co-rated items among users in finding correlated neighbors for an active user, and co-rated items between user and filterbot to find agreed filterbots.

Example: If each user has rated a tiny percentage of total items as shown in Table 1.

Table 1 Example of user's rating and filterbots (IF agent) data

(Where rating scale ranges from –1 to 1)

Item | User	A	B	Filterbot1	Filterbot2	Filterbot3	Filterbot4
1	0.5		1	1	0	0
2			1	0	0	0
3			0	1	1	1

Overlap Items

From Table 1, the overlap (or number of co-rated item) between UserA and Filterbot1 is only 1 item. Then correlated value between UserA and Filterbot1 will have low quality. GroupLens system has specified that the overlap between user's ratings must not be less than 50 items in order to achieve qualified correlated value.

In addition, IF agents in current hybrid systems tend to produce low quality rating data, because their user profile is fixed beforehand and cannot be changed to reflect the user's preference accurately even user's interest has successively changed.

The purpose of this paper is to propose a better recommendation method to cope with problems mentioned above. An experimental recommender system for movie, called e-Yawara (extended Yawara) for performing our method, is also described.

2.Method

We have realized that the sparsity rating problem would be eliminated when co-rated items are not used in finding correlated neighbors for an active user. Accordingly, we proposed to use similarity between user feature vectors (*UFVs*) of each couple of users in finding correlated neighbors instead. When sparsity rating problem is eliminated, the system would be able to produce more accurate results.

In an effort to obtain higher level of accuracy on the results, we increase quality of prediction score in IF agent by using similarity between user feature vector (*UFV*) and movie feature vector (*mfv*) as rating data predicted by our IF agents. The *UFV* in our method will be updated to get closer to *mfv* of movie that user needs according to successive change of user's ratings and user's history data. Therefore, our IF agents can produce better rating data every time, after each user has rated interest value toward any movies or taken action on our movie space.

The idea about characteristic of those *UFV* and *mfv* is taken from Yawara system [4]. Their characteristics are *UFV* = (*W1*,*W2*,…,*Wn*); where *Wi* is the weight that user gives for keyword (i), and *mfv*(*i*) = (*Wi1, Wi2*, ….., *Win*); where *Wij* is the weight that movie(i) has toward keyword(j) and *n* in both *UFV* and *mfv* is the number of keywords. The keyword list in *mfv* is 20 movie categories extracted from category table in cinema magazine. The weight of *mfv* ranges from –1 to 1. It is positive when that keyword (or category) matches with the movie, 0 when unknown and negative when it does not match with the movie. The keyword list in the vector *UFV* is same movie category list as that in the vector *mfv*. The weight of *UFV* also ranges from –1 to 1. It is positive when the user likes a movie in that category (or keyword), 0 when the user feels neutral, and negative when the user dislikes a movie in that category.

Here, in order to calculate the similarity between vectors, we define it as a distance between two vectors, so called a non-similarity. We define non-similarity as *L1* distance (Manhattan distance [7]). The distance between vector *A* (*Wa1, Wa2*,….,*Wan*) and vector *B* (*Wb1*,*Wb2*,…,*Wbn*) is defined as follows.

$$d = \sum_{i=1}^{n} | Wai - Wbi |$$

where, *n* is the number of weight elements, and $0 \le d \le 2n$ and the size of each weight *W* is $-1 \le W \le 1$.

We define the similarity as the difference between the value of full distance (*2n*) and distance (*d*), (*2n*-*d*). Then normalize the similarity.

$$Similarity = 1 - \frac{d}{2n}$$

where, $0 \le \text{Similarity} \le 1$

For the update process of user feature vector (*UFV*) in our method, we considered that when a user click on some movie objects frequently and he/she is very interested in those movies, his/her feature can be considered to become closer to the feature of those movies. Accordingly, the change of rating data (or interest value) and history data of a user are mapped to the movie features, then these mapped properties will be used to update user feature of such user to get closer to the features of the movies that such user needs.

$$\boldsymbol{UFVupdate = \mathrm{a}\times(\mathrm{b}Hchange + \mathrm{c}Ichange)MFV + UFVprevious}$$

Where ***UFVupdate*** is the updated user feature vector, ***Hchange*** is a vector which represents the history data of an active user, ***Ichange*** is a vector which represents the interest value of an active user, ***MFV*** is a matrix of movie feature vector for all movies, and **a, b** and **c** are all coefficients. The detail of this equation is described in paper of Yawara system [4].

It means that user profiles in our IF agents can be change dynamically to accurately reflect user's preference.

3.Results

In order to implement and evaluate our method, we generated an experimental recommender system for movie called e-Yawara. In our experimental evaluation, 100 movie data were provided in database and 16 users were willing to use our system. Total ratings collected from our experiment sum up to 599 ratings. 20% of the ratings of each user were randomly selected. These ratings comprised the *test set*. The remaining 80% formed the *source set*.

In order to evaluate our method, we compared our method with a conventional method. We simulated the method of GroupLens system[2] on the same data set of e-Yawara system, and then we predicted a value for each rating in the test set based on each method, using only data in the source set. We used Mean Absolute Error (MAE) which has been used previously by Shadanand & Maes in Ringo system [6] and Coverage to be criteria in comparing. As presented in Table 2, the MAE of e-Yawara (0.3402) is less than GroupLens system (0.3586). The difference of MAE is around 0.018. Considering that the rating scale of our experiment ranges from -1 to + 1, the difference of 0.018 is significant. From this difference of MAE, it can be concluded that our method produces more accurate results than recommendation method used in GroupLens system. Table 2 also shows that our method can provide the good performance with no loss in coverage.

We also evaluated our method by determining user's preference (Qualitative Result) between the result of e-Yawara and GroupLens system. We simulated recommendation method of GroupLens system on the same data set of e-Yawara. We then let users give preference scores toward the top three movies that have highest score predicted from both e-Yawara and GroupLens methods. From Table3, the number of users who like the top three movies predicted by e-Yawara (67% from all users) is higher than those done by GroupLens (56% from all users) and the number of users who dislike the top three movies predicted by e-Yawara (0% from all users) is less than those done by GroupLens(22% from all users). It is therefore apparent that our method produces more satisfactory results than a conventional system (GroupLens).

Table 2 Algorithm Evaluation Result

Method	MAE	Coverage
GroupLens	0.3586	83.08%
e-Yawara	0.3402	100%

where rating scale in our experiment range from -1 to 1

Table 3 Qualitative Result

Method	Percentage of all users		
	Like	Dislike	Feel Neutral
GroupLens	56%	22%	22%
e-Yawara	67%	0%	33%

The reason that makes the result of e-Yawara more satisfactory and more accurate than GroupLens System is that GroupLens System employed Pearson r correlation coefficient to find correlated neighbors for an active user. However, there is a large number of items in the general systems, so it is difficult for users to have co-rated items enough to find highly correlated neighbors who are the good predictors. For example, some parts of the rating data that we got from our experiment presented in Table 4 shows that among of 5 movies, User 1 and User 4 have only 1 co-rated item, and other users have no any mutual co-rated item. From the data in Table 4, GroupLens System claimed that User 4 is the good predictor for User 1 because ratings of their co-rated items (item2 or Mars Attacks), the value are almost equal (0.5 and 0.6). As a matter of fact, only one number of co-rated items cannot conclude whether User 4 is a good predictor. Therefore, the results of GroupLens System tend to be incorrect in assuming User 4 is a good predictor of User 1, so is the data of User 4 used to predict User 1. e-Yawara uses similarity between user features in place of co-related items in finding correlated neighbor for an active user. So the result of e-Yawara is still accurate even there is very few or none of co-rated items.

Table 4 Matrix of user's ratings (where rating scale range from –1 to 1)

Movie	User	1	2	3	4
Reality Bites					
Mars Attacks		0.5			0.6
Trainspotting					
Empire Records					
Swingers					

4.Conclusion and Future Work

In this paper, a new hybrid method that combines CF technique ***without "sparsity rating" problem*** and IF technique ***with dynamic user profile*** has been proposed in order to provide better recommendation**.** A movie recommender system based on our hybrid method called e-Yawara system has also been created to evaluate our method. The remaining sparsity rating problem that occurs when co-rated items is used can be eliminated by adopting similarity between user feature vectors. The quality of prediction value from IF agent is also improved by adopting user profile which can be updated according to successive change of user's preference. The evaluation of recommendation Coverage and Mean Absolute Error (MAE) shows that e-Yawara is superior to a conventional system. The evaluation about qualitative result also shows that the recommendation results of e-Yawara system are more satisfactory than the results from the conventional system such as GroupLens system.

We have realized that further study and development is required in order to make e-Yawara more efficient. One problem of e-Yawara system is that it uses only rating number of user's interest to express user's opinion. However, only rating number cannot be used to express all type of preferences that people have toward each movie. We have 40 completed questionnaires on what reasons (or factors) individual uses to make decision in watching a movie. Their reasons could be categorized into Movie categories, Popularity of actor or actress, His/her interest degree toward actor or actress, Director, Freshness of film, Popularity degree of each film, Movie preview, Soundtrack, Intensity of visual effect, Intensity of animation effect, Location where the film is shot, Story Line, Movie's origin, Award and Title obtained, Top ranking film and Critic's complements. In our future work, we consider these reasons (or factors) as essential to be included in our movie recommender system in order to give better recommendation results.

Acknowledgements

The authors wish to express our special thank to Professor Minoru Ohyama and Research Associate Yasuto Nakanishi for their valuable suggestions. We would also like to thank our laboratory members for useful comments.

References

1. Paul Resnick and Hal R. Varian. Recommender Systems. Communications of the ACM, vol. 40, No. 3, pp.56—58 (1997)
2. Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom and John Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In Proceeding of 1994 Conference on Computer Supported Collaborative Work, pp.175--186.: ACM (1994)
3. Loren Terveen, Will Hill, Brian Amento, David McDonald and Josh Creter. PHOAKS: A System for Sharing Recommendations. Communications of the ACM, vol. 40, No. 3, pp 59—62 (1997)
4. Hideaki Kanai, Katsuya Hakozaki:A Browsing system for a database using Visualization of User Preferences, Proceedings on Information Visualisation 2000, pp.277--282, IEEE CS (2000)
5. Nathaniel Good, J. Ben Schafer, Joseph A. Konstan, Al Borchers, Badrul Sarwar, Jon Herlocker, and John Riedl. Combining Collaborative Filtering with Personal Agents for Better Recommendations. Proceedings of the 1999 Conference of the American Association of Artifical Intelligence (AAAI-99). pp 439—446 (1999)
6. Upendra S. and Patti M. Social Information Filtering: Algorithms for Automating "Word of Mouth". In proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems, pp. 210—217 (1995)
7. Ingwer Borg and Patrick Groenen. Modern Multidimensional Scaling Theory and Applications. Springer-Verlag, pp 99—101 (1997)