Vol. 43 No. SIG 5(TOD 14) IPSJ Transactions on Databases June 2002

*Regular Paper*

# An Improved Recommendation Method for Better Filtering Information out of Database

Saranya Maneeroj,† Hideaki Kanai†† and Katsuya Hakozaki†

Content-based filtering and collaborative filtering techniques have been used for selecting information based on user's previous preference tendency and opinions of other people who have similar tastes with the user. Combining both filtering techniques or 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 rating problem and increases the quality of Information Filtering agent of the hybrid systems. This propose is to recommend information that reflect the user interest more accurately. As implementing our method, we also present an experimental recommender system for movie, called e-Yawara (extended Yawara). The evaluation shows that e-Yawara is more efficient and provides more accurate results than conventional filtering systems, both collaborative filtering and hybrid systems.

## 1. Introduction

In everyday life, we rely on recommendations from other people, movies and book reviews printed in newspaper, and general surveys such as restaurant guides. 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. Examples are 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. Various recommendation methods have been proposed. The recent recommendation methods try to combine content-based filtering and collaborative filtering techniques in order to get better recommendation results.

However, they still suffer from several problems. The first one, they use co-rated items in finding correlated neighbors for an active user, so they always suffer from sparsity rating problem. The other one, a user profile in their Information Filtering (IF) agent (or rating robots) is fixed, cannot be changed to reflect the user's preference accurately, so their recommendation results may be inaccurate.

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

The next section reviews the exist filtering techniques used in current recommender systems and discuss their problems. We then present our approach on how to cope with these problems in Section 3. In Section 4, we explain about our system structure. In Section 5, we present evaluation of our approach and its results. We then discuss about the derived evaluation results in Section 6. Finally, we give some concluding remarks in the last section.

## 2. Related Works

### 2.1 Content-based Filtering

The early recommender systems use content-based filtering techniques. The systems build a profile of user preference by observing the behavior of an individual user to predict which information would be selected or rejected. The Yawara system [4)] is an example of content-based filtering system, which is developed by our laboratory members. It is a Web-based virtual library. Yawara recommends documents for a user by changing configuration of objects on document space, according to successive change of the individual user preference (or user's profile), in order to make that user eas-

† Graduate School of Information Systems, University of Electro-Communications
†† Division of Mathematics & Computer Science, Vrije Universiteit Amsterdam, the Netherlands

**Table 1** Example of user's rating and filterbots (IF agent) data (where rating scale ranged from −1 to 1).

| Item | User | | filterbot1 | filterbot2 |
|---|---|---|---|---|
| | A | B | | |
| 1 | 0.5 | | −1 | 1 |
| 2 | | ↓ | 0.5 | 0 |
| 3 | | Overlap Data | 0 | 1 |

ily understand which information on the space he/she seems to be interested in.

Although content-based filtering techniques have been success, certain drawback still remains on the techniques. The drawback is that they do not provide much in the way of serendipitous discovery. The serendipitous discovery means that a system will give satisfactory recommendation results, which a user has never thought that he/she will be interested in.

### 2.2 Collaborative filtering

To cope with the drawback of content-based filtering techniques, collaborative filtering techniques have been proposed. Collaborative filtering systems recommend items based on the opinion (rating) of other users who have similar tastes. One well-known example of these system is GroupLens system [2], which has been implemented for filtering Usenet news postings. It provides an open architecture wherein people can rate articles and their ratings are distributed through the net. GroupLens employs *Pearson r correlation* coefficients to determine correlated value between users. It uses co-rated items to find correlation neighbors for an active user (a user for whom the system is finding recommendation results).

The limitations of collaborative filtering 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 collaborative filtering 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.

### 2.3 Hybrid System (Combination of Content-based and Collaborative Filtering Techniques)

The next level of recommender systems is hybrid system. Hybrid systems combine content-based and collaborative filtering techniques in an effort to overcome the limitations of each, such as MovieLens system [5], a movie recommender system created by the GroupLens research team. It adds filterbots (Information Filtering agent) into collaborative filtering system. Filterbots are rating robots that participate as members of a collaborative filtering system. Filterbots help users who agree with them by providing more ratings upon which recommendations could be made. Although current hybrid systems can solve the lacking of serendipitous discovery problem in content-based filtering techniques and early-rater problem in collaborative filtering 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 filterbot and user to find agreed filterbots. For example, if each user has rated a tiny percentage of total items as shown in **Table 1**, the overlap (or number of co-rated items) between UserA and Filterbot1 is only one item. Then the quality of correlated value between UserA and Filterbot1 will be low. 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 if the user's interest or user's behavior has successively changed.

## 3. Research Approach

### 3.1 Mechanism for Calculating Correlated Neighbors for an Active User

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 spar-

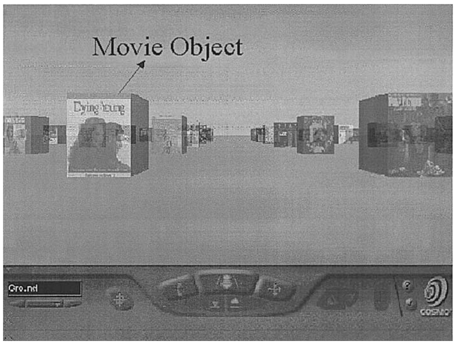



**Fig. 1** Movie space.

sity rating problem is eliminated, the system would be able to produce more accurate results.

### 3.2 Mechanism for Calculating IF Agent Rating

In an effort to obtain higher level of accuracy on the recommendation results, we increase quality of prediction score (or rating) of 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 (or user's interest) 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 (**Fig. 1**).

### 3.3 The characteristic of *UFV* and *mfv*

The idea about characteristic of $UFV$ and $mfv$ is taken from Yawara system[4]. These vectors are represented by Vector Space model. Their characteristics are $UFV = (w_1, w_2, .., w_n)$; where $w_i$ is the weight that user gives for keyword (i), and $mfv(i) = (w_{i1}, w_{i2}, ...., w_{in})$; where $w_{ij}$ 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 the vector $mfv$ 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 vector $UFV$ has the same keyword list as that the vector $mfv$ has. 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.

### 3.4 Similarity between Two Vectors

In order to calculate the similarity between two vectors, we define it using a distance between two vectors, so called a non-similarity. We define non-similarity as *L1* distance (Manhattan distance[7]) because it can take size of each component of a vector into account.☆

The distance between vector $A$ $(w_{a1}, w_{a2}, ....,$ $w_{an})$ and vector $B$ $(w_{b1}, w_{b2}, ...., w_{bn})$ is defined as follows:

$$d = \sum_{i=1}^{n} |w_{ai} - w_{bi}| \qquad (1)$$

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

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

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

where, $0 \leq similarity \leq 1$.

### 3.5 Mechanism for Updating User Profile in IF Agent

Although systems with filterbots such as MovieLens system are effective, certain drawback remains on them. The drawback is user profile in their IF agent is fixed. In order to change user profile in IF agents dynamically, we update user's preference (i.e., user feature vector : $UFV$), by a feedback technique. Our Filterbot can produce prediction score (or ratings) updated every time user's rating increase.

For the update process of $UFV$ in our method, we considered that when a user clicks on some movie objects in our movie space presented in Fig. 1 frequently and he/she is very interested in those movies, his/her feature can be considered to become closer to the feature of

☆ A widely used measure of similarity is the vector cosine similarity measure, namely the cosine of the angle between two vectors. This measure is useful in case of no need to compare size of each component of both vectors. In case of computing similarity between a vector: $A$ and a vector: $B$ produced by scalar multiplication of $A$, the similarity is 1. On the other hand, there is need to differentiate size between each component of $A$ and the corresponding of $B$ in our research. That is why *L1* distance is used here.

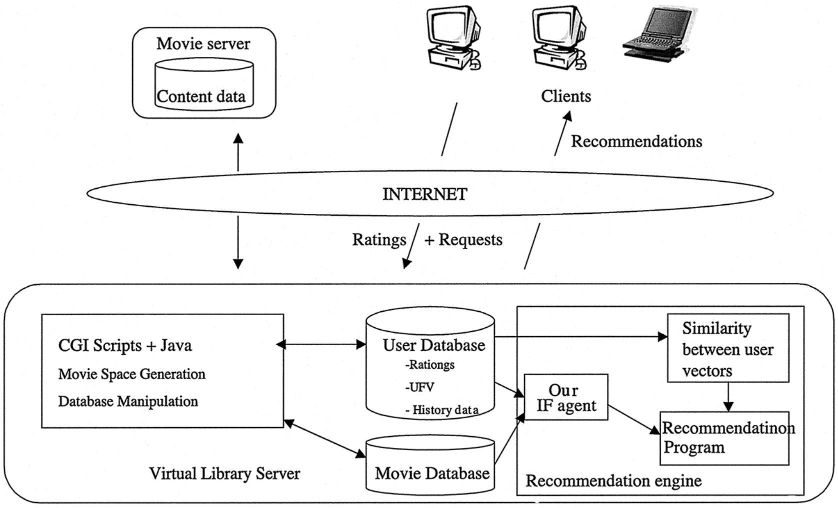



**Fig. 2** e-Yawara architecture.

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 the user feature.

$$UFV_{update} = a \times (bH_{change} + cI_{change})MFV + UFV_{previous} \quad (3)$$

where $UFV_{update}$ is the updated user feature vector, $H_{change}$ is a vector which represents the history data of an active user, $I_{change}$ is a vector which represents the interest value of an active user, $MFV$ is matrix of all movie feature vectors, and $a$, $b$ and $c$ are all coefficients. The detail of this equation is described in paper of Yawara system [4].

## 4. Our System Structure

In order to implement our approach, we generated an experimental recommender system for movie called e-Yawara (extended Yawara). Our system is designed based on an assumption that the target information are distributed on the Internet. **Figure 2** shows our system architecture.

As shown in Fig. 2, our Virtual Library Server consists of 4 main parts: Movie space, Movie database, User database and Recommendation engine. Movie space (Fig. 1) contains 100 3D movie objects. Movie database contains $mfv$ (movie feature vector) of all movies in our system. User database can be divided into 4 types of database: "Original $UFV$ database", "Updated $UFV$ database", "User's interest value database" and "History database". *Original $UFV$ database* contains the original $UFV$ (user feature vector) of all users. *Updated $UFV$ database* contains the most updated version of $UFV$ of all users. The vector $UFV$ will be updated all times after users click on some movie objects on Movie space (Fig. 1) or rate interest value for such movies, in order to reconstruct $UFV$ (user feature vector) to get closer to the real interest of that user. *User's interest value database* contains all ratings of interest value toward each movie of all users, $-1 \leq interestvalue \leq 1$. *History database* contains data about activities of all users. Activities refer how often each user clicks on each movie object on Movie space (Fig. 1). The last part of Virtual library server is Recommendation engine which generates recommendation results.

In our system, Apache 1.3.6 on Linux PC acts as WWW server software, is used for a server. The system is implemented by Java (JDK 1.2) and VRML 2.0. In order to access database in the server from user clients, Perl is used for CGI Script. In addition, EAI (External Authoring Interface) is used in order to operate the Movie space described with VRML 2.0 (Virtual Reality Modeling Language) [8] from Java Applet dynamically. User clients enable users to access and operate virtual library server. The clients are implemented as Netscape Communicator 4.7 (or newer) on Windows 98 (or newer) PC with Cosmo Player 2.1; plug-in software for VRML 2.0. Movie servers contains movie information which will be referred by users via URL when they want to see the movie in detail.

## 5. Experimental Evaluation

### 5.1 Data

We generated an experimental recommender system for movie called e-Yawara. The structure is presented in section 4. 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*.

### 5.2 Evaluation Criteria

In our experiment, three different measures were used for determining the quality of recommendation produced.

**Coverage** is a measure of the percentage of items for which system can provide recommendations. A high coverage value indicates that recommendation system provides assistance in selecting among most of the items.

**MAE (Mean Absolute Error)** is a measure of deviation of recommendations from the true user- specified value. The lower the MAE, the more accurate the recommendations.

**ROC** sensitivity is an indication of how effectively the system can steer people toward highly-rated items and away from low-rated one. Operationally, it is the area under the receiver operating characteristic curve (ROC)-a curve that plots the *sensitivity* vs. *1-specificity* [9] of the *test set*. Sensitivity refers to the probability of randomly selected good items being accepted by the system. Specificity refers to the probability of randomly selected bad items being rejected by the system. The ROC sensitivity ranges from 0 to 1, where 1 is perfect filter, 0.5 is random filter. To operationalize ROC, we must determine which items are "good" and which are "bad". Since the rating scale in our experiment ranges from −1 to +1, we use the user's own rating, with a mapping that the one greater than or equal to 0.3 are good and the one less than 0.3 are bad.

### 5.3 Comparing e-Yawara with the Conventional Systems

We compared our method with some conventional methods. The first one is the method of GroupLens system — a well-known collaborative filtering system. Another one is the method of MovieLens system — a hybrid system developed by the Grouplens research team. We simulated the method of GroupLens and MovieLens systems 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.

**Table 2** Evaluation results between e-Yawara and GroupLens.

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

Considering MovieLens and e-Yawara, both systems in our experiment are hybrid system that has one filterbot on collaborative filtering framework, but the filterbot of MovieLens is fixed while filterbot of e-Yawara can be updated dynamically according to successive change of user's preference. Therefore, we tried to calculate filterbot rating values of MovieLens when each user has rated only half of the whole rating in the test set. We then took these acquired filterbot rating values to incorporate with the whole rating in test set to calculate recommendation results.

### 5.4 Evaluation Results

**(1) Comparison between e-Yawara and GroupLens**

We used Mean Absolute Error (MAE), which has been used previously by Shardanand and Maes in Ringo system [6] and Coverage to be criteria in comparing e-Yawara and GroupLens system. 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 or it is around 5.3% of MAE derived by e-Yawara. Considering that the rating scale of our experiment ranges from −1 to +1, the difference of 0.018 is significant. Table 2 also shows that our method can provide the good performance with no loss in coverage, but GroupLens system provides only 83.08% of coverage. From these MAE and Coverage values, it can be concluded that our method is more effective and produces more accurate results than recommendation method used in GroupLens system. We did not mention ROC metric, because it can be expressed that e-Yawara is more efficient than GroupLens when just using only basic metric as MAE and coverage.

**(2) Comparison between e-Yawara and MovieLens**

MovieLens research team has proposed many kinds of filterbot (IF agent). In comparing MovieLens and e-Yawara systems, both sys-

**Table 3** Evaluation results between e-Yawara and GroupLens.

| Method | MAE | Coverage |
|---|---|---|
| MovieLens | 0.3635 | 100% |
| e-Yawara | 0.3402 | 100% |

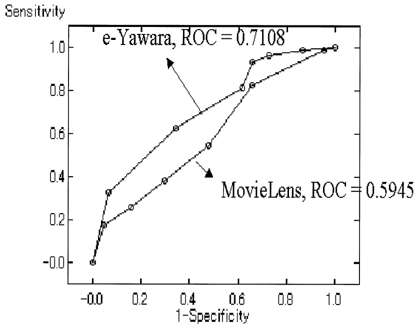



**Fig. 3** ROC value between e-Yawara and MovieLens.

tems are hybrid system, the number of filterbot in these two systems should be equal. Since there is only one kind of filetrbot in e-Yawara, there should be only one kind of filterbot in MovieLens system also. According to the results from MovieLens research paper[5], a certain robot that produced the best recommendation is Mega-GenreBot. We then implemented MovieLens method (a hybrid method; combination of content-based filtering and collaborative filtering techniques) by combining Mega-GenreBot (IF agent) into the GroupLens system, which is a collaborative filtering framework.

We used MAE and ROC to be criteria in comparing recommendation quality between e-Yawara and MovieLens systems. We do not mention coverage metric, because the coverage value of these two systems are equal (no loss of coverage) as presented in **Table 3**. Table 3 also shows that, The MAE value of e-Yawara (0.3402) is less than MovieLens system (0.3635) around 0.023 or it is around 6.8% of MAE derived from e-Yawara. The same reason as comparing with GroupLens above, the rating scale of our experiment ranges from $-1$ to $+1$, so the difference of 0.023 is significant.

As presented in **Fig. 3**, The ROC value of e-Yawara (0.7108) is higher than MovieLens system (0.5945) around 0.116. As the experience of the MovieLens research team, they assert that the ROC difference of "0.01" or more are meaningful. From these MAE and ROC values, it can be concluded that our method is more effective and produces more accurate results than recommendation method used in MovieLens system.

**Table 4** Matrix of user's ratings (where rating scale ranged from $-1$ to 1).

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

## 6. Discussion

### 6.1 Experimental Results

As mentioned in the previous section, e-Yawara is more effective and its results more accurate than both GroupLens and MovieLens systems. One of the reasons is that GroupLens and MovieLens systems employed *Pearson r correlation* coefficient (or using co-rated items) 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, namely the good predictors. For example, some parts of data we got from our experiment in **Table 4** show 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 and MovieLens systems claim that User 4 is the good predictor for User 1 because ratings of their co-rated item (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 item cannot conclude whether User 4 is a good predictor. Therefore, the results of GroupLens and MovieLens systems 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.

In the case of e-Yawara, it uses similarity between user feature vectors in place of co-related items in finding correlated neighbor for an active user. So the results of e-Yawara are still accurate even there is very few or none of co-rated items.

Another reason is filterbot concept. Considering about GroupLens and MovieLens systems, normally, filterbot provides more ratings to users who agree with the filterbot. Therefore, it enables MovieLens system to produce more recommendations than GroupLens system, as shown from coverage value of both systems (GroupLens 83.08%, MovieLens 100%) in Table 2 and Table 3 respectively. The reason is that MovieLens system is a combination of collaborative filtering framework and filterbot, and GroupLens system is a collaborative filtering framework.

As a matter of fact, filterbot should enable the recommendations produced by MovieLens system to be more accurate than those produced by GroupLens system. However, since rating data in our experiment is so few, the accuracy of these two systems are insignificantly different, as shown from MAE value of both systems (GroupLens 0.3586, MovieLens 0.3635) in Table 2 and Table 3 respectively. Therefore, from these coverage and MAE value, it reflects that the system with filterbot (MovieLens) is more effective than the system without filterbot (GroupLens).

Recommendation results produced by MovieLens system is less accurate than e-Yawara because a filterbot in MovieLens system cannot re-rate all times as the number of user's rating increases from only half up to the whole test set. The efficiency in helping users to select high quality items from the item set is also less than e-Yawara, as shown from the result of MAE and ROC values respectively in Table 3 and Fig. 3. The reason is that a filterbot in e-Yawara can produce prediction score (or ratings) updated every time user's rating increase.

### 6.2 Other Systems

Ian Soboroff proposed a model for filtering collaboratively using document content represented by the generalized vector space model [10]. They have addressed a similarity which exits between collaborative filtering algorithms and retrieval using the generalized vector space model. We have concentrated on the recommendation method and updating method of user profile in IF agents.

Fab [11] is a system which combine content-based filtering and collaborative filtering. The system uses only content-based approach to choose which items founded by any IF agents are rated by the user's content based profile and the most highly rated items are recommended to the user. In contrast, our method uses collaboration among users to determine the ratings of predicted movie and uses the content-based user profile to compute similarity among users.

Pazzani [12] has proposed "Collaboration via content", which is the hybrid approach in the context of recommending restaurants. This is similar to our method, but user profiles in IF agents is fixed and user profiles cannot be updated dynamically according to a change of a user's preference. On the other hand, Basu et al.,[13] proposed an inductive learning approach to recommendation. This approach use an inductive learning system *Ripper* instead of IF agents in order to predict user preference. The approach is able to use both ratings information and other forms of information about each artifact in predicting user preferences. In contrast, we have concentrated on dealing with a problem that IF agents in current hybrid systems tend to produce low quality rating data. The inductive learning approach is beyond the scope of our paper.

## 7. Conclusions and Future Work

In this paper, an improved recommendation method has been proposed in order to help users more easily filter the interesting information out from the large database. This proposed method is a new hybrid method that can provide better recommendation results by combining collaborative filtering technique without sparsity rating problem and content-based filtering technique with dynamic user profile. A movie recommender system based on our hybrid method called e-Yawara system has also been created to implement and 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 collaborative filtering system, GroupLens system. The evaluation of recommendation ROC sensitivity and Mean Absolute Error (MAE) also shows that e-Yawara is superior to MovieLens system, which is a current hybrid system developed by GroupLens research team.

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 types 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.

Other problem is this time of our experiment has been built from a small group of users and a small database of movie data. Future work should both incorporate larger movie and user sets in order to make e-Yawara more practical, predictive and accurate.

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

## References

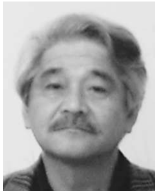
1) Resnick, P. and Varian, H.R.: Recommender Systems, *Comm. ACM*, Vol.40, No.3, pp.56–58 (1997).
2) Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews, *Proc. 1994 Conference on Computer Supported Cooperative Work*, pp.175–186, ACM (1994).
3) Terveen, L., Hill, W., Amento, B., McDonald, D. and Creter, J.: PHOAKS: A System for Sharing Recommendations, *Comm. ACM*, Vol.40, No.3, pp.59–62 (1997).
4) Kanai, H. and Hakozaki, K.: A Browsing system for a database using Visualization of User Preferences, *Proc. Information Visualization 2000 (IEEE CS)*, pp.277–282, IEEE (2000).
5) Good, N., Schafer, J.B., Konstan, J.A., Borchers, A., Sarwar, B., Herlocker, J. and Riedl, J.: Combining Collaborative Filtering with Personal Agents for Better Recommendations, *Proc. 1999 Conference of the American Association of Artificial Intelligence (AAAI-99)*, pp.439–446, AAAI (1999).
6) Shardanand, U. and Maes, P.: Algorithms for Automating Word of Mouth, *Proc. CHI'95 Conference on Human Factors in Computing Systems*, pp.210–217, ACM (1995).
7) Borg, I. and Groenen, P.: *Modern Multidimensional Scaling Theory and Applications*, Chapter 6, pp.99–101, Springer-Verlag New York, Inc. (1997).
8) Hartman, J. and Wernecke, J.: *The VRML 2.0 Handbook — Building Moving Worlds on the Web*, Addison-Wesley (1996).
9) Swets, J.: Measuring the Accuracy of Diagnostic Systems, *Science240*, Vol.240, No.4857, pp.1285–1289 (1988).
10) Soboroff, I. and Nicholas, C.: Collaborative Filtering and the Generalized Vector Space Model, *Proc. SIGIR 00*, pp.351–353, ACM (2000).
11) Balabanovi, P. and Shohama, Y.: Fab: Content-based, Collaborative Recommendation, *Comm. ACM*, Vol.40, No.3, pp.66–72 (1997).
12) Pazzani, M.J.: A Framework for Collaborative, Content-Based and Demographic Filtering, *Artificial Intelligence Review*, Vol.13, No.5–6, pp.393–408 (1999).
13) Basu, C., Hirsh, H. and Cohen, W.: Recommendation as Classification: Using Social and Content-Based Information in Recommendation, *Proc. 1998 Conference of the American Association of Artificial Intelligence (AAAI-98)*, pp.714–720, AAAI (1998).

(Received December 20, 2001)
(Accepted March 28, 2002)

(Editor in Charge: *Keizo Oyama*)

**Saranya Maneeroj** is a Ph.D. student at Graduate School of Information Systems, the University of Electro-Communications. She received a M.S. in computer engineering from Graduate School of Information Systems, the University of Electro-Communications. Her current research focuses on Recommender System based on Collaborative Filtering and Information Filtering techniques.

**Hideaki Kanai** is a postdoctoral fellow in the IMSE Business Informatics group, Division of Mathematics and Computer Science, at Vrije Universiteit Amsterdam the Netherlands, where he works on realizing the Semantic Web in European-Union funded projects. His current research interests include ontologies-based information visualization, Intelligent Web Services and knowledge management using dynamic ontologies. He is a member of IPSJ, ACM and IEEE. He received D. Eng. degree in system engineering from University of Electro-Communications, Tokyo, Japan in 1998 and was a research associate at Graduate School of Information Systems in University of Electro-Communications, Tokyo, Japan until 2001.



**Katsuya Hakozaki** received B.S. degree in electronic engineering in 1963 and D. Eng. degree in 1981 both from Kyushu University. He joined NEC Corporation in 1963 and was engaged in research and development of computer architecture, performance evaluation, and computer networks. He is a professor of University of Electro-Communications, Graduate School of Information Systems since 1994. His recent interests are digital libraries, mobile communication aid, multimedia Internet applications. He was awarded IPSJ Best Paper Award in 1982. He is a member of Information Processing Society in Japan, Institute of Electronics, Information and Communication Engineering, IEEE and ACM.