Trust-Aware Recommender System Incorporating Review Contents

Hideyuki Mase, Katsutoshi Kanamori, and Hayato Ohwada

***Abstract*—Personalized recommendation systems can help people find things that interest them and are widely used in developing the Internet or e-commerce. Collaborative filtering (CF) seems to be the most popular technique in recommender systems. However, CF is weak in the process of finding similar users. To resolve these problems, trust-aware recommender systems (TaRSs) have been developed in recent years. In this study, we propose a new approach that incorporates the content of reviews in a TaRS. In addition, we use a new dataset that is collected from the Yahoo!Movie website, whereas traditional research has used Epinions or Movielens. Finally, we evaluate the experiment results using precision and coverage.**

***Index Terms*—Collaborative filtering, content of reviews, trust network, Yahoo!Movie dataset.**

I. Introduction

The development of Internet and e-commerce systems has yielded a plethora of available information. Thus, recommendation systems that employ information filtering technology have been developed to provide useful data. CF is the most successful information filtering technique in research and in the real world [1], (e.g., Amazon.com or ebay.com). However, CF is weak in the recommending process of finding similar users, which involves computing similarities in the items that users rate. However, the number of items (e.g., books or movies) is very large, and computing user similarity is very difficult because users seldom rate many items in real world. Thus, the recommending process of computing user similarity has failed. That failure is especially clear when the user rates only a few items, which is known as the "cold start user" problem [2]. To solve this problem, a trust-aware recommender system (TaRS) has been developed in recent years [3], [4].

CF is implicitly related with only a user community of composing users in on-line shop or recommender system through the rated common items by users. However, on consumer review and price comparison web sites (e.g., Amazon.com or Epinions.com), users have the opportunity that to rate to the reviews of other users. Thus, a user is explicitly connected with other users. As illustrated in Fig. 1, this network is made up of trust statements. TaRS is based on the implicit trust-network developed by the trust propagation of users.

Manuscript received August 8, 2013; revised December 10, 2013.

Hideyuki Mase, Katsutoshi Kanamori, and Hayato Ohwada are with the Department of Industrial Administration, Graduate School of Science and Technology, Tokyo University of Science, Yamazaki 2641, Noda-City, Chiba, Japan (e-mail: h-mase@ohwada-lab.net, katsu@rs.tus.ac.jp, ohwada@ia.noda.tus.ac.jp).

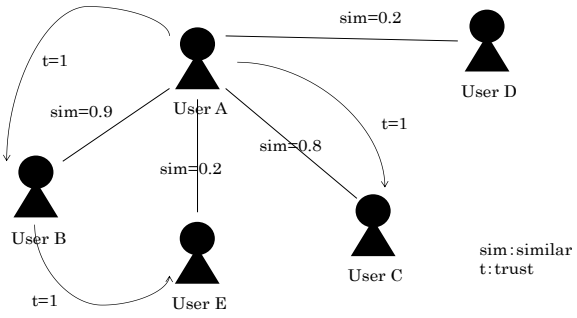

Fig. 1. Similarity and trust network.

This trust -network is utilized for finding similar users and thus resolves the weakness of CF. Traditional TaRS approach has ever researched with using some methods. However, the content of reviews is not taken into account in the recommendation process. Therefore, we propose a new TaRS approach that combines the trust- network and the content of reviews and have collected a dataset in the real world and used it in our experiment.

This paper is structured as follows. Section II details the motivation for our proposal, describes related studies on TaRS, and compares them. Section III describes the proposed method, and Section IV describes the evaluation experiment conducted to determine the validity of the proposed technique. Section V presents and discusses the experiment results. Finally, Section VI describes the conclusions.

II. Related Works and Motivation

This section describes related works on TaRS and the motivation of our research.

A. Paolo Massa and Paolo Avesani, Introducing the TaRS Architecture

Massa and Avesani [3], [4] used the rating matrix and trust matrix as input data for their system, and used Epinions dataset derived from Epinions.com. They use a trust propagation algorithm (Mole-Trust) to infer indirect trust values and the Pearson Correlation [5] to compute user preference similarity. Mole-Trust [6] is to predict the trust score of a source user on a target user by walking the social networking starting from the source user and by propagating trust along trust edges. Intuitively, the trust score of a user depends on the trust statements of other users weighted by the trust scores of users who issued the trust statements. The weight by which the opinion of a user is considered depends on the perceived trustworthiness of that user.

Massa and Avesani proposed the basic TaRS architecture, in which user similarity replaces the trust metric. The typical CF algorithm involves two steps. The first step is to compute

DOI: 10.7763/IJMLC.2014.V4.399

user similarity as input for a matrix of ratings. The most used and most effective similarity metric is the Pearson correlation coefficient. The second step is to predict the rating the active user would give to a certain item. The predicted rating is the weighted sum of the ratings given by using the value that is computed by the user similarity metric in the first step. The formula for the second step is

$$p_{a,i} = \overline{r_a} + \frac{\sum_{u=1}^{k} w_{a,u}(r_{u,i} - \overline{r_u})}{\sum_{u=1}^{k} | w_{a,u} |} \quad (1)$$

where $p_{a,i}$ is the predicted rating that active user a would provide for item i, r_u is the average of the rating provided by user u, $w_{a,u}$ is the user similarity weight of a and u as computed in the first step, and k is the number of users (neighbors) whose ratings of item i are considered in the weighted sum.

The evaluation experiment has two important results. First, TaRS alleviates the cold-start problem. Although improvement of accuracy compare to that of the CF algorithms is small, the coverage is improved by 20%. The reason for little improvement in accuracy could be the inclusion of dissimilar users' preferences. Therefore, we seek to improve accuracy by considering both the trust statements and the review content. Second, most researchers use the MovieLens dataset, so the recommender system evaluation still has some problems. We therefore collected a new dataset from the Yahoo!Movie web site and used it in our evaluation experiment.

B. Touhid Bhuiyan, Yue Xu, Audun Josang Huizhi Liang and Clive Cox

Bhuiyan *et al.* [7] proposed developing trust networks based on personalized user tagging information. Their tagging information is any type of online information resources or products in an online community (e.g., web pages and videos) that the user tagged. They extract keywords from product descriptions using such text- mining techniques as tf-idf.

Their experiment confirmed that their proposed approach slightly improves precision and recall, compared with traditional CF, based on Jaccard's coefficient.

However, they do not consider the review content and compare to traditional TaRS. Because the review content indicates user preferences or item features, we propose a TaRS that takes them into account and compare it with traditional CF and TaRS.

C. Other Related Works

This subsection describes some related works.

Gollbeck and Hendler [8]-[10] demonstrate their proposed approach using the FilmTrust website in which users can rate movies and write reviews. They also state how much they trust other users' movies ratings on ten levels. Their prediction is based on the trust metric from TidalTrust [11] and ratings. They used reviews to sort movies. The most relevant reviews come from the most trusted users, thus they will be shown these reviews first. In other words, they regard the review as a useful trust statement. However, they do not analyze the review content but use it in recommendation process.

Agarwal and Bharadwaj [12] proposed a Friend Recommender System. This system computes similarity based on user profile and behavior, and then makes a recommendation that uses CF, generating enhanced neighborhood sets based on trust propagation. Kim and Park [13] proposed a movie recommender system using the group-aware social network model. This social network is composed of user profile and user intention based trust model from rating. Their experiment used the Movielens dataset, which consists of user ratings of movies and user profiles (e.g., gender, age, and occupation). They [12], [13] proposed a TaRS based on CF incorporating user profile data but did not use features of the user or items from the review content. Also, their dataset [12] has only the ratings that 20 users rate. So, we collected enough the ratings that user rate items.

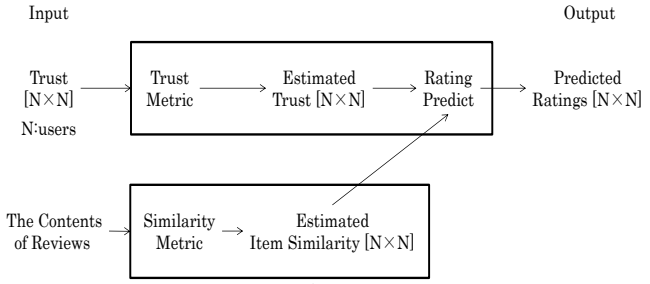

Fig. 2. Architecture of the proposed approach.

III. Proposed Approach

This section describes the proposed approach, the architecture of which is presented in Fig. 2. Our approach has an input trust statement and the review content, and the output is predicted ratings. In the recommendation process, the proposed approach has two main steps : computing the trust metric and computing the similarity metric from the review content.

A. Trust Metric

We use Mole-Trust [6] as the trust metric. Mole-Trust [6] is to predict the trust score of a source user on a target user by walking the social networking starting from the source user and by propagating trust along trust edges. The Mole-Trust metric can be modeled in two steps. Step1 involves removing cycles in the trust network and hence transforming it into a directed acyclic graph. Step 2 consists of a graph walk starting from the source node with the goal of computing the trust score of visited nodes. The formula of the predicted score of a user is as follows.

$$trust(u) = \frac{\sum_{i \in predecessors} (trust(i) * trust\_edge(i,u))}{\sum_{i \in predecessors} (trust(i))} \quad (2)$$

For example, in Fig. 3, when predicting the trust score of user Mark regarding Lisa, Mole-Trust accepts only the opinions of Bob and Carol about Lisa; it does not accept the trust statement issued by Brown because the predicted trust score of Brown is 0.1, less than the threshold (0.5 in this example). Therefore, the predicted trust score of Lisa is

$(0.8 \times 0.6 + 0.9 \times 1.0) / (0.7+0.9) = 0.825$.

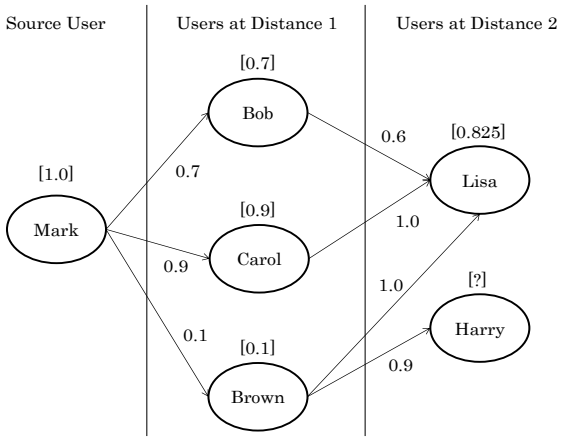

Fig. 3. Mole-trust example.

B. Similarity Metric

To compute the similarity metric, we first investigate 510,449 reviews. We analyze every word in the review content, and investigate the frequency of the word's appearance. Based on those results, we then read the review contents watching for the word and extract words which we assume that become the feature of items. Finally, we extracted 5000 words from the reviews. Table I lists some of the keywords. For example, the keyword is "fashion" if some of the reviews consist of "fashion". It seems that users are interested in the "fashion" of the movie or the movie has "fashion" potential element.

TABLE I: THE EXAMPLES OF SOME KEYWORDS LIST

story	wonderful	masterpiece	missed	positive
performance	actor	academy	fiction	Shakespeare
action	feeling	composition	New York	Doraemon
time	award	cruelty	Judea	location
content	speech	man and woman	yakuza	originality
interesting	expression	impact	otaku	biotechnology
music	character	spy	situation	science
love	episode	sexy	BGM	sex
impression	difficult	humor	beatles	train
family	end	entertainment	CIA	authority
disappointment	fantasy	voice actor	the kabuki	infection

We set keyword vectors for an item and compute item similarity using cosine-based similarity [5]. Two keywords are regarded as two vectors in an m-dimensional user space.

The similarity between these two vectors is measured by computing the cosine of the angle between them. Formally, in the $m \times n$ ratings matrix, the row is user and the column is keyword. Similarity between keywords i and j denoted by $sim(i, j)$ is given by

$$sim(i, j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\| \vec{i} \|_2 \otimes \| \vec{j} \|_2} \quad (3)$$

where "·" is the dot-product of the two vectors. This similarity has both item preference similarity and trust similarity because we assume that the review content has the item's features and helps users judge whether to trust other users.

C. Rating Prediction

This subsection proposes the following equation to predict rating for user to item. This equation is composed of item similarity and trust metric.

$$p_{a,t} = 0.5 \times (\overline{r_a} + \frac{\sum_{u=1}^{k} w_{a,u} (r_{u,t} - \overline{r_u})}{\sum_{u=1}^{k} | w_{a,u} |}) + 0.5 \times \frac{\sum_{i=1}^{h} sim(i,t)(r_{a,i})}{| sim(i,t) |} \quad (4)$$

Here, $p_{a,t}$ is the predicted rating that active user a would provide for item t, r_u is the average of the rating provided by user u, $w_{a,u}$ is the user similarity weight of a and u as computed in trust value. k is the number of users (neighbors) whose ratings of item t are considered in the weighted sum. $sim(i,t)$ is the item similarity, and h represents the items user a rated.

The metric of this equation is that it includes both the trust statement and the item similarity. Improvement in accuracy and coverage is expected because we include item features from real user opinions in the review content.

IV. EXPERIMENT

In this section, we describe experiments that we conducted to evaluate our proposed approach. We present the dataset used and introduce the evaluation protocol and measures.

A. Yahoo!Movie Dataset

The dataset we used in our experiments was collected from the Yahoo!Movie web-site (http://movies.yahoo.co.jp/) Fig. 4. This website is a consumer opinion site on which users review movies and assign them numeric ratings from 1 (min) to 5 (max). A user can also state whether he or she trusts other users' movie ratings or reviews. On this website, the trust statement value is 0 (distrust) or 1 (trust). For example, if user A trust user B, the trust statement value is 1.

Our dataset consists of 15,367 users who rated 23,154 different items at least once. The total number of reviews is 510,449, and the total number of trust statement is 127,814.

Fig. 4. Review and a rating on the Yahoo! Movie web site.

Rating matrix sparsity (the percentage of empty cells in the

matrix users × items) of the collected dataset is 99.85653%. In addition, the total number of cold-start users who rated less than 5 is 5,270, which represents 34.29426% of the population. Another point is the distribution of ratings. In our dataset, 29% of the ratings are 5 (best), 32% are 4, 23% are 3, 10% are 2, and 6% are 1 (worst). The mean rating is 3.66. The characteristics we present differ from those of the MovieLens dataset, which is the most commonly used dataset for recommender system evaluation and from the Epinions dataset, which is the most commonly used dataset for TaRS evaluation. In the Movielens dataset, all users rate items at least 20 times and all ratings balance is good. Thus, it has no cold-start users. For the Epinions dataset, 52.82% of the population are cold-start users; 45% of the ratings are 5, and 29% are 4. This is a good dataset for TaRS; furthermore almost half of the ratings are 5. Therefore, our dataset is good for both ratings balance and cold-start users.

B. Evaluation Protocol and Measures

We apply three approaches in our experiment: traditional user-based CF, a TaRS based on a Mole-Trust [6] metric, and our proposed approach.

Our experiment protocol is 10-fold cross-validation on the Yahoo! Movie dataset. The data is first partitioned into ten equal sized segments or folds. One fold is used for testing, while the remaining nine folds are used for learning. This is process is repeated 10 times, and mean accuracy is taken. We also apply two types of dataset in our experiment: all ratings and only the ratings of cold-start users.

We used four evaluation measures. One is Mean Absolute Error (MAE) as an evaluation accuracy measure. Formally, if n is the number of actual ratings in an item set (test data), then MAE is defined as the average absolute difference between n pairs of predicted ratings p_k and actual ratings r_k, and is given by

$$MAE = \frac{\sum_{k=1}^{n} | p_k - r_k |}{n} \quad (5)$$

A lower MAE produces more accurate predictions. And better recommendations.

Second is the Mean Absolute User Error (MAUE) [4]. We first compute the MAE for each user independently and then average all the Mean Error computations. This is very important when the dataset has many cold-start users.

Third is user coverage (Ucov) that how much recommender system can recommend to users. This is interesting with analyzing the behavior of the recommend algorithm to cold-start users. This is given by

$$U\,\mathrm{cov} = \frac{\sum_{u \in U} \rho_u}{| U |} \quad (6)$$

where ρ_u is predictable ratings to users (if ρ_u is more than 1, ρ_u is 1, otherwise it is 0).

Fourth is ratings coverage (Rcov) as an evaluation measure. It is important for a recommender system to be able to predict the number of ratings because many of the ratings become hardly on a very sparse dataset that contains a large portion of cold-start users and of items rated just by one user. The formula is

$$R\,\mathrm{cov} = \frac{\sum_{u \in U} | recset_u |}{| I |} \quad (7)$$

where $recset_u$ is predictable ratings and I is an item set. Higher the coverage results in a better recommendation system.

V. Experiment Results and Discussion

This section presents actual experiment results and then discusses them.

A. Propagation Numbers

This subsection describes the number of propagations. Here, we refer to the algorithm that propagates trust up to distance 1 as MT1, the one that propagates trust up to distance 2 as MT2, and the one that propagated trust up to distance 3 as MT3. The average number of directly trusted users (MT1) is 38.1, while the average number of comparable users is 138.6, for which the Pearson Correlation coefficient is computable. Propagating for MT2 is 919.6, and that for MT3 is 4583.7. This increase is significant. This pattern was observed for cold-start users (16.8 for MT1, 541.6 for MT2, 3433.0 for MT3).

These results confirm that using trust propagation is more effective than using CF in finding neighbor users.

B. Results of Using Ratings for Cold-Start Users

As indicated in Table II, our proposed approach outperforms the traditional CF and TaRS. In Fig. 5, the proposed MAE and MAUE are lower than the others with the exception of TaRS Mole MT1. However, in Fig. 6, our approach has significantly better performance in rating coverage and user coverage. Our proposed approach thus outperforms traditional CF and TaRS on the ratings of cold-start users because our prediction includes both trust and item similarity from the reviews.

TABLE II: Results of Cold-Start Users

Cold User Ratings	MAE	MAUE	Rating Cov	User Cov
CF	0.859	0.857	7.74%	7.49%
TaRS_Mole_MT1	0.323	0.262	0.64%	0.40%
TaRS_Mole_MT2	0.689	0.640	2.10%	1.35%
TaRS_Mole_MT3	0.772	0.756	9.30%	6.20%
Proposed_MT1	0.730	0.720	84.89%	83.81%
Proposed_MT2	0.710	0.701	85.77%	84.97%
Proposed_MT3	0.702	0.692	93.88%	93.67%

TABLE III: Results of All Users

All User Ratings	MAE	MAUE	Rating Cov	User Cov
CF	0.747	0.795	87.10%	90.43%
TaRS_Mole_MT1	0.795	0.841	40.26%	30.82%
TaRS_Mole_MT2	0.731	0.773	61.92%	42.55%
TaRS_Mole_MT3	0.723	0.755	64.07%	44.17%
Proposed_MT1	0.752	0.791	56.20%	40.71%
Proposed_MT2	0.745	0.767	81.46%	58.65%
Proposed_MT3	0.722	0.753	87.01%	60.76%

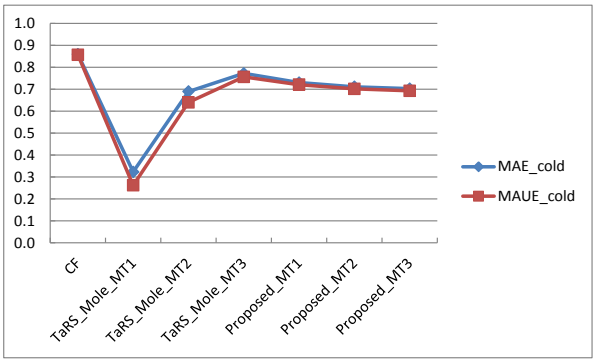

Fig. 5. Accuracy results of cold-start users.

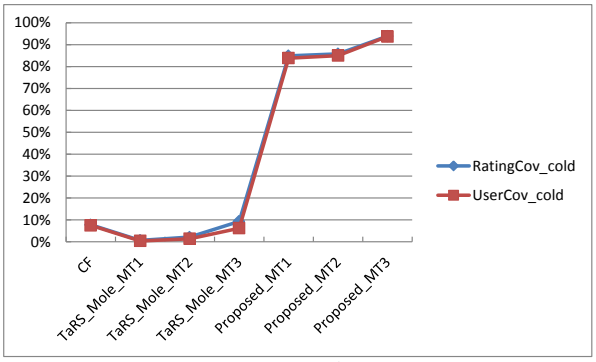

Fig. 6. Coverage results of cold-start users.

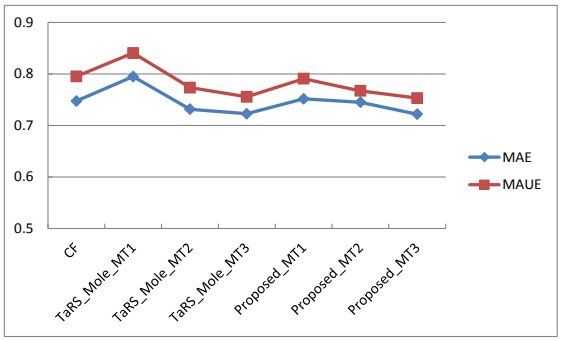

Fig. 7. Accuracy results of all users.

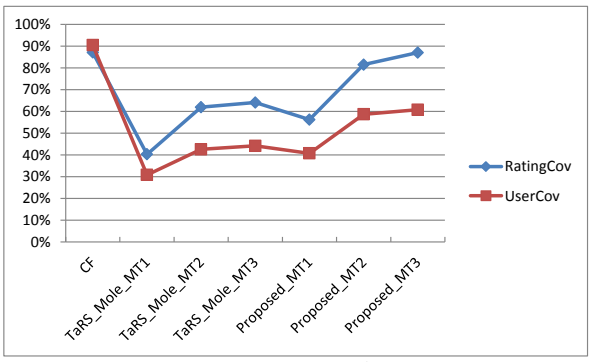

Fig. 8. Coverage results of all users.

In general, it is difficult for them to find neighborhoods and compute weighting because cold-start users rate few items. Our proposed method takes item similarity from the content of the reviews. Therefore, in computing the prediction rating for cold-start users, we can find both trust network and item similarity.

C. Results of Using All Ratings

As indicated in Table III, our proposed MT1, MT2, MT3 outperforms a TaRS based on a Mole-Trust. In the precision (MAE, MAUE), our proposed is not much different from TaRS Mole-Trust. However, in the coverage (Rcov, Ucov), our proposed significant outperforms. And, the MAE of our proposed MT2 is higher than a TaRS Mole-Trust MT2, but the MAUE is lower. Thus, this result shows that our proposed is effective for cold-start users.

Next, our proposed MT2, MT3 outperforms CF in the precision. We assume that the reason is the number of heavy users: that is, finding neighbors in CF is easy because our experiment database includes more heavy users than the Epinions dataset does [4]. Therefore, we believe that the number of predictable in CF is more than our proposed.

Finally, the more number of propagations is, the better the precision and coverage is. This is because the more number of propagations is easy to find neighbors.

VI. Future Works

This paper presents TaRS taking into account the content of reviews. We show that our proposed approach outperforms traditional approaches in the accuracy and coverage.

In the future, we will try to select different keywords from this experiment and to classify them into categories. And a weakness of our proposed method is the huge computing cost when using a large amount of data. Thus, we will try to propose a more efficient prediction method in the future.

References

[1] J. Breese, D. Hecherman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proc. 14th Conference on Uncertainty in Artificial Intelligence*, UAI, 1998, pp. 43-52.

[2] S. J. Gong, "A collaborative filtering recommendation algorithm based on user clustering and item clustering," *Journal of Software*, vol. 5, no.7, July 2010.

[3] P. Massa and P. Avesani, "Trust-aware collaborative filtering for recommender systems," *Lecture Notes in Computer Science*, Springer, vol. 3290, pp. 492-508, 2004.

[4] P. Massa and P. Avesani, "Trust-aware recommender systems," in *Proc. the 2007 ACM Conference on Recommender Systems*, Minneapolis, 2007, pp. 17-24.

[5] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. the 10th International Conference on World Wide Web*, ACM, Hong Kong, 2001, pp. 285-295.

[6] P. Massa and P. Avesani, "Trust metrics on controversial users: Balancing between Tyranny of the majority," *International Journal on Semantic Web and Information Systems*, pp.39-64, 2007.

[7] T. Bhuiyan, Y. Xu, A. Josang, H. Liang, and C. Cox, "Developing trust networks based on user tagging information for recommendation making," *Web Information Systems Engineering*, Springer-Berlin Heidelberg, pp. 357-364, 2010.

[8] J. Golbeck and J. Hendler, "Filmtrust: Movie recommendations using trust in web-based social networks," in *Proc. IEEE Consumer communications and networking conference*, University of Maryland, 2006, vol. 96.

[9] J. Golbeck and J. Hendler, "Generating predictive movie recommendations from trust in social networks," in *Proc. Fourth International Conference on Trust Management*, Pisa, Italy, May 2006.

[10] J. Golbeck and J. Hendler, "Inferring binary trust relationships in web-based social networks," in *Proc. the 2006 ACM Transactions on Internet Technology*, pp. 497-529, 2006.

[11] J. Golbeck and J. Hendler, "Computing and applying trust in web-based social networks," Ph.D. Dissertation, University of Maryland, College Park, Maryland, 2005.

[12] V. Agarwal and K .K. Bharadwaj, "Trust-enhanced recommendation of friends in web based social networks using genetic algorithms to learn user preferences," *Trends in Computer Science, Engineering and*

Information Technology, Springer Berlin Heidelberg, pp.476-485, 2011.
[13] M. Kim and S. Oh Park, "Group affinity based social trust model for an intelligent movie recommender system," *Multimedia Tools and Applications*, pp.1-12, 2013.

H. Mase graduated from the Department of Industrial Administration, Faculty of Science and Technology, Tokyo University of Science, Noda City, Japan, in 2013.

He is a student at Tokyo University of Science Graduate School, Division of Science and Engineering Industrial Administration Master's course since 2012, Noda City, Japan. His research interests are in the field of recommendation systems

H. Ohwada graduated from the Department of Industrial Administration, Faculty of Science and Technology, Tokyo University of Science, Noda City, Japan, 1983. Then he graduated from Tokyo University of Science Graduate School, Division of Science and Engineering Industrial Administration Doctoral course Completed program with degree, Noda City, Japan, 1988.

He was a research associate (Tokyo University of Science) from 1988 to 1998, lecturer (Tokyo University of Science) from 1999 to 2000, associate professor (Tokyo University of Science) from 2001 to 2004. Then he is a professor at Tokyo University of Science Faculty of Science and Engineering Department of Industrial Administration from 2005. His research interests are in the fields of Inductive Logic Programming and Bioinformatics.

K. Kanamori earned a doctorate in information Science at the Tokyo University of Science, Noda City, Japan, in 2009.

He is a research associate at Tokyo University of Science, Department of Industrial Administration, Faculty of Science and Technology. His research interests are in the fields of artificial intelligence.