J Inf Process Syst, Vol.15, No.1, pp.127~136, February 2019
https://doi.org/10.3745/JIPS.01.0036

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

A Simple and Effective Combination of User-Based and Item-Based Recommendation Methods

Se-Chang Oh* and Min Choi**

Abstract

User-based and item-based approaches have been developed as the solutions of the movie recommendation problem. However, the user-based approach is faced with the problem of sparsity, and the item-based approach is faced with the problem of not reflecting users' preferences. In order to solve these problems, there is a research on the combination of the two methods using the concept of similarity. In reality, it is not free from the problem of sparsity, since it has a lot of parameters to be calculated. In this study, we propose a combining method that simplifies the combination equation of prior study. This method is relatively free from the problem of sparsity, since it has less parameters to be calculated. Thus, it can get more accurate results by reflecting the users rating to calculate the parameters. It is very fast to predict new movie ratings as well. In experiments for the proposed method, the initial error is large, but the performance gets quickly stabilized after. In addition, it showed about 6% lower average error rate than the existing method using similarity.

Keywords

Collaborative Filtering, Electronic Commerce, Recommender System, Sparsity

1. Introduction

The movie recommendation is a typical application of recommendation systems. The solutions for the recommendation systems can be divided into user-based and item-based approaches. Hybrid approaches that combine the two methods also have been studied.

The item-based approach is also referred to as the content-based method. It recommends new items with attributes similar to the items the user highly prefer [1]. In this approach, classification or clustering items using some features is performed to find similar items. It has the advantage that it works relatively well even when data are insufficient. It assumes that the user's rating is based on the features indicating the essential characteristics of items. However, there is a problem that the measure of similarity between items is different from the user's rating.

The user-based approach is also referred to as collaborative filtering. It recommends new items that are highly preferred by other users similar to the target user [2]. In this case, the taste of each user can be reflected sufficiently by using the user's preferences when calculating the similarity between the users [3]. However, this approach has the problem of sparsity caused by incomplete calculation of similarity

Manuscript received July 5, 2016; first revision September 26, 2016; second revision December 6, 2016; accepted January 24, 2017.
Corresponding Author: Min Choi (mchoi@cbnu.ac.kr)
* Dept. of Computer Software, Sejong Cyber University, Seoul, Korea (scoh713@gmail.com)
**Dept. of Information and Communication, Chungbuk National University, Cheongju, Korea (mchoi@cbnu.ac.kr)

between the users when the data are insufficient. As a result, it is difficult to secure the reliability. To alleviate this problem, some demographic information is used for classifying the user [4].

Mixed approaches are new attempts to take advantage of both approaches by combining the two approaches. There is an example of the mixed approach using both the genre of movie in the item-based approach and the address of users in the user-based approach [5]. However, the assumption that the movies belonging to the same genre will get a similar rating from people is weak. Furthermore, the demographic classification is difficult to be considered as a grouping of people with similar preferences. For these reasons, we need a more fundamental approach. We can find a fundamental approach in [6]. The similarity between users and between items is computed using users' rating. Then the user-based filtering and the item-based filtering methods are appropriately combined using this similarity. In this method, however, the time for making prediction is too long due to the complex formulas, and the problem of sparsity still remains due to too many parameters to be calculated.

In this study, we propose a fast and highly accurate recommendation method that makes the complex and incomplete factors simple and clear in combining the user-based and the item-based filtering. In addition, this method is relatively free in the problem of sparsity by minimizing the number of parameters to be calculated.

2. Related Work

This section introduces the movie recommendation method proposed in [6], and analyzes the advantages and disadvantages of this method. This method calculates the similarity between users and between items by using the Pearson correlation coefficient [7]. Then the set of similar users for each user and the set of similar items for each item are obtained based on this similarity. Finally, the estimation of rating is calculated using the rating of users or users' information for items belonging to the set of similarity. The similarity calculated previously is used as the weight for each estimation when each estimation is combined for calculating the final result.

2.1 Calculation of Similarity

In this method, the similarity between users and between items is calculated based on the rating of users for items. For this, the similarity $Sim(a, b)$ between user a and user b in the user-based filtering is obtained by the following equation.

$$Sim(a,b) = \frac{\mathrm{Min}(|I_a \cap I_b|, \gamma)}{\gamma} \cdot \frac{\sum_{i \in I_a \cap I_b} (r_{a,i} - \bar{r}_a) \cdot (r_{b,i} - \bar{r}_b)}{\sqrt{\sum_{i \in I_a \cap I_b} (r_{a,i} - \bar{r}_a)^2} \cdot \sqrt{\sum_{i \in I_a \cap I_b} (r_{b,i} - \bar{r}_b)^2}} \quad (1)$$

In Eq. (1), I_a, $r_{a,i}$ and $\bar{r}_a$ represent the set of items that the user a has evaluated, the rating of the user a for the item i and the average rating of the user a, respectively. In addition, the constant γ is the threshold for $|I_a \cap I_b|$. It is used to prevent the overestimation of similarity when the size of the initial item set is small. The equation calculates the similarity of two users a and b in evaluating on the same item i compared to their average ratings. The similarity is calculated using the Pearson correlation coefficient.

In the same way, the similarity $Sim(i, j)$ between item i and item j in the item-based filtering is

obtained by the following equation.

$$Sim(i,j) = \frac{\mathrm{Min}(|U_i \cap U_j|, \delta)}{\delta} \cdot \frac{\sum_{u \in U_i \cap U_j} (r_{u,i} - \bar{r}_i) \cdot (r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U_i \cap U_j} (r_{u,i} - \bar{r}_i)^2} \cdot \sqrt{\sum_{u \in U_i \cap U_j} (r_{u,j} - \bar{r}_j)^2}} \tag{2}$$

In Eq. (2), U_i and $\bar{r}_i$ represent the set of users who have evaluated the item i and the average rating for the item i, respectively. In addition, the constant δ is the threshold for $|U_i \cap U_j|$. It is used to prevent the overestimation of similarity when the size of the initial user set is small. The equation calculates the similarity of two items i and j evaluated by the same user u compared to their average ratings. The similarity is also calculated using the Pearson correlation coefficient.

2.2 Selection of Similar Neighbors

Based on the similarity obtained earlier, we can define the set of users similar to a particular user and the set of items similar to a particular item. First, we define the set of users similar to a user u as follows.

$$S(u) = \{a \mid Sim(a,u) > \eta, a \neq u\} \tag{3}$$

The constant η in Eq. (3) is the threshold for the similarity between users. It is the baseline for selecting the users similar to a user u.

In the same way, we define the set of items similar to an item m as follows.

$$S(m) = \{i \mid Sim(i,m) > \theta, i \neq m\} \tag{4}$$

The constant θ in this equation is the threshold for the similarity between items. It is the baseline for selecting the items similar to an item m.

2.3 Prediction for Rating

Then we can predict a user's rating for a new item using the sets of similar users and of similar items as follows.

$$\hat{r}_{u,m} = \lambda \cdot \left(\bar{r}_u + \frac{\sum_{a \in S(u)} Sim(a,u) \cdot (r_{a,m} - \bar{r}_a)}{\sum_{a \in S(u)} Sim(a,u)}\right) + (1-\lambda) \cdot \left(\bar{r}_m + \frac{\sum_{i \in S(m)} Sim(i,m) \cdot (r_{u,i} - \bar{r}_i)}{\sum_{i \in S(m)} Sim(i,m)}\right) \tag{5}$$

In Eq. (5), the constant λ is the ratio for combining the rating values predicted both by user-based filtering and by item-based filtering. In user-based filtering, the prediction for rating is calculated as the weighted average of the difference between the rating of user a who belongs to the similar group of the user u for the item m and the average rating of user a. Here, $Sim(a,u)$, the similarity between user a and user u, is used as the weight of the term for user a. In item-based filtering, the prediction for rating is calculated as the weighted average of the difference between the rating of the user u for item i that belongs to the similar group of the item m and the average rating for item i. Here, $Sim(i,m)$, the similarity between item i and item m, is used as the weight of the term for item i.

2.4 Algorithm Analysis

According to the experiments, the method proposed in [6] appears to be more accurate than the various movie recommendation methods like similarity fusion [8], cluster-based smoothing [9], aspect model [10], personality diagnosis [11], and user-based Pearson correlation coefficient [12] that are relatively accurate.

The algorithm used in this method can be described in Fig. 1.

1. for all data tuple (u, m, *rate, time*)
2. $user\_set = \{a | I_a \cap I_u \neq \emptyset\}$;
3. for all $a \in user\_set$
4. calculate $Sim(a,u)$ using equation (1);
5. select $S(u)$ using equation (3);
6. $item\_set = \{j | U_m \cap U_j \neq \emptyset\}$;
7. for all $i \in item\_set$
8. calculate $Sim(i,m)$ using equation (2);
9. select $S(m)$ using equation (4);
10. calculate $\hat{r}_{u,m}$ using equation (5);
11. update: I_u, U_m, $\bar{r}_u$, $\bar{r}_m$;

Fig. 1. Algorithm used in former method.

This method has two fundamental problems. First, lots of parameters need to be calculated each time. For example, there are U^2 parameters for $Sim(a,u)$, M^2 parameters for $Sim(i,m)$, U parameters for $\bar{r}_a$ and M parameters for $\bar{r}_i$, where U is the number of users, M is the number of items. This leads to the problem of sparsity. In addition, there is a problem in the equation itself. The denominators of Eqs. (1) and (2) can be zero. In this case, the similarity value cannot be calculated. In fact, the denominators of the equations can be zero not only in the early stage when there are not enough data to calculate the equation but also in the later whenever a new user or item is added, and the set $S(u)$ and $S(m)$ in Eqs. (3) and (4) become empty sets frequently because of low similarity even if the denominators are not zero. In these cases, only $\bar{r}_u$ and $\bar{r}_m$ are the basis for prediction of the rating in Eq. (5). As a result, the accuracy of prediction is degraded. Actually, 99.78% of $S(u)$ and 98.4% of $S(m)$ are found to be empty in Eqs. (3) and (4) when we experiment not modifying the threshold values used in [6].

Second, several constants are used to keep the adequacy of the calculated results in intermediate steps such as Eqs. (1), (2), (3), and (4). Also a constant is used as the ratio of combining the finally calculated terms in Eq. (5). There is no absolute criterion for determining these constants. Thus, these can only be determined experimentally. This may be improper partially. Therefore, the performance of the prediction for rating is limited.

In [13], two methods were proposed in order to partially solve these problems as follows. First, the constants γ and δ in Eqs. (1) and (2) were omitted. However, the similarity calculated by each method may be overestimated due to this in the beginning. Second, the user-based prediction for rating is calculated as the weighted average of the rating of user a who belongs to the similar group of the user u

for the item m. Here, $Sim(a,u)$ is used as the weight. In the same way, the calculation of the item-based prediction of rating is simplified. However, there seems to be a limit in enhancing the accuracy due to such a simplification.

3. Proposed Method

In this paper, we propose a method that effectively combines the user-based and item-based methods for movie recommendation by solving the two problems of the paper [6] as analyzed previously. The basic idea of the paper is briefly described in [14].

3.1 Solution to Problem of Sparsity

First, we have to find a way to replace the $Sim(a,u)$ and $Sim(i,m)$ calculated in Eqs. (1) and (2) to solve the problem of sparsity. These values are used as weights for the prediction of rating in Eq. (5). The $Sim(a,u)$ is used as the weight of user a's rating in the user-based prediction. Likewise, the $Sim(i,m)$ is used as the weight of the rating for item i in item-based prediction. In many cases, the prediction result is distorted by an improper constant value or by a denominator of zero in Eqs. (1) and (2). Therefore, these values are replaced by 1 to prevent this problem in this paper. That is to prevent that incomplete data are used for prediction.

The $S(u)$ in Eq. (3) potentially includes all the users because it is hard to distinguish the users similar to user u. In fact, $S(u)$ includes all the users when the threshold η is low, and becomes empty if the threshold η is high. Conceptually, however, if $S(u)$ is defined to be the set of the users who are helpful for the user-based filtering, it can includes the users who evaluated item m. Therefore, it is appropriate to use U_m instead of $S(u)$. In the same way, $S(m)$ in Eq. (4) can be defined to be the set of the items that are helpful for the item-based filtering. Therefore, I_u, the set of all the items that have evaluated by user u, would be appropriate to use instead of $S(m)$.

Eventually, the Eq. (5) is simplified as the following equation by replacing $Sim(a,u)$ with *1*, $Sim(i,m)$ with *1*, $S(u)$ with U_m, and $S(m)$ with I_u.

$$\hat{r}_{u,m} = \lambda \cdot \frac{\sum_{a \in U_m} (\bar{r}_u + r_{a,m} - \bar{r}_a)}{|U_m|} + (1-\lambda) \cdot \frac{\sum_{i \in I_u} (\bar{r}_m + r_{u,i} - \bar{r}_i)}{|I_u|} \tag{6}$$

3.2 Ratio of Combining User-Based and Item-Based Methods

In order to combine the results of the user-based and the item-based methods, it is necessary to determine the ratio based on the reliability of the two results. For this, the rational and simple way is to use the ratio of the number of data used for calculation of each result. This is because we can improve the reliability of prediction results when we use more data in general.

Therefore, the combination ratio is calculated by the following equation.

$$\lambda = \frac{|U_m|}{|U_m|+|I_u|}, 1-\lambda = \frac{|I_u|}{|U_m|+|I_u|} \tag{7}$$

Applying this ratio, the Eq. (6) can be eventually replaced by the following.

$$\hat{r}_{u,m} = \frac{\sum_{a \in U_m} (\bar{r}_u + r_{a,m} - \bar{r}_a) + \sum_{i \in I_u} (\bar{r}_m + r_{u,i} - \bar{r}_i)}{|U_m| + |I_u|} \tag{8}$$

The Eq. (8) calculates the prediction for rating in two ways. First, in the user-based method, the difference between the average rating of user *u* and user *a* is calculated, and the rating of user *a* for item *m* is added to the difference. Then, the average of these values is calculated for all users *a* who have rated item *m*. Second, in the item-based method, the difference between the average rating for item *m* and item *i* is calculated, and the rating of user *u* for item *i* is added to the difference. Then, the average of these values is calculated for item *i* that user *u* have rated for. Finally, these two prediction results are combined using the ratio of the number of data used in each method. The prediction for rating can be calculated faster and more accurate using the Eq. (8) than that using the Eq. (5).

3.3 Algorithm Analysis

The algorithm used in the proposed method is described in Fig. 2.

1. for all data tuples (*u*, *m*, *rate, time*)
2. calculate $\hat{r}_{u,m}$ using equation (8);
3. update: I_u, U_m, $\bar{r}_u$, $\bar{r}_m$;

Fig. 2. Algorithm used in proposed method.

This new algorithm is compared with the existing algorithms described in Fig. 1 as follows. First, the steps 2–9 of the existing algorithm are not necessary in the new algorithm. The time complexity of the steps is O(R*U*M*log$R$). In addition, the step 11 of the existing algorithm is corresponding to the step 3 of the new algorithm. The time complexity of this part is O(R*(U+M)). The step 10 of the existing algorithm is corresponding to the step 2 of the new algorithm. The formula to be calculated in this step has been changed from the Eq. (5) into the Eq. (8). The time complexity of these steps are both O(R*(U+M)*log$R$). Therefore, the time complexity of the new algorithm is O(R*(U+M)*log$R$), which is much lower than O(R*U*M*log$R$) of the existing algorithm.

4. Experimental Result

4.1 Experimental Environment

The experimental data used in this study are the MovieLens 100K dataset [15]. In the dataset, the number of users is 943, the number of movies is 1,682 and the number of evaluation data is 100,000. Particularly the evaluation data are composed of a tuple (a user ID, item ID, rating, and time stamp).

In [6], the dataset is divided into training data and testing data. After training the model, the performance of the model is measured using the test data. However, the experimental method of this

paper is not the same as the method used in [6]. Instead, in this paper, the prediction process for the currently read tuple is performed repeatedly, while referencing to the previously read tuples, reading each tuple one by one from the beginning. It is because pre-training of models with sufficient data is not practically possible when applied to the practical movie recommendation system. The reality is that new users and new movies are added continuously.

Therefore, the experimental data were taken out orderly to predict the user's rating from the dataset sorted by the time stamp. That is, the prediction at time t is made based on the data received from time 1 to time t–1.

4.2 Accuracy of Prediction

The performance measure used in the experiment is the mean absolute error (MAE). It is defined as the following equation.

$$\text{MAE} = \frac{\sum_{u,i} |r_{u,i} - \hat{r}_{u,i}|}{N} \tag{9}$$

It is the average of the absolute value of difference between $r_{u,m}$ and $\hat{r}_{u,m}$, where $r_{u,m}$ is obtained from each sample in the dataset, and $\hat{r}_{u,m}$ is the prediction result calculated by the Eq. (8).

Figs. 3 and 4 show the result of comparing the prediction accuracy of the method used in [6] and the method proposed in this paper.

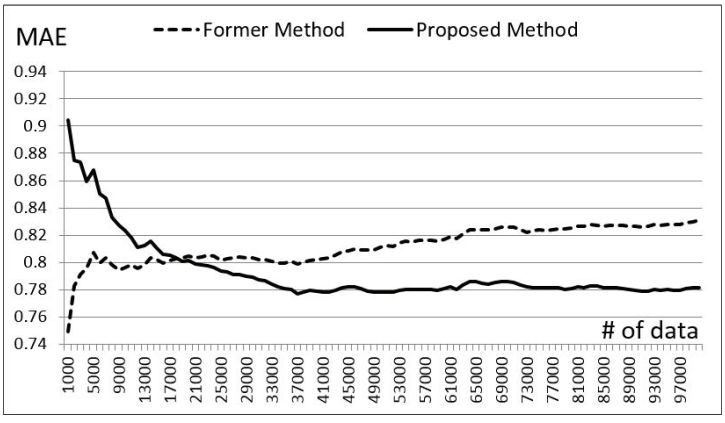

Fig. 3. Prediction accuracy of former method versus proposed method.

In Fig. 3, the vertical axis is the MAE calculated from the data in the range from the beginning up to each point in the horizontal axis. The curve labeled 'former method' represents the performance of the method used in [6], and the curve labeled 'proposed method' represents the performance of the method proposed in this paper.

In the graph, the MAE of the 'former method' continues to increase over time. On the other hand, the MAE of the 'proposed method' is maintaining a constant level. Also the 'proposed method' curve shows that the accuracy of prediction is low in the early stage. It is because the amount of data used for the prediction is small. While accumulating experience gradually, however, more accurate predictions

become possible based on it. As a result, the 'proposed method' appears 0.0494 lower MAE compared to the 'former method' based on the evaluation results for the 100,000 item set. This reduction in prediction error is about 6%.

4.3 Efficiency of Prediction Algorithm

The efficiency of the prediction algorithm is also important no less than the accuracy. This is because the prediction algorithm can be applied to a system that many people deal with many items such as Internet shopping mall. Therefore, it is very important how fast the processing time is increasing with respect to the amount of the data accumulating over time. The following graph compares the cumulative processing time of the 'former method' and the 'proposed method'.

In Fig. 4, the processing time of the 'former method' is increasing rapidly as the amount of data increases. On the other hand, the processing time of the 'proposed method' is increasing very slowly. The time needed to process all of the 100,000 data is 472.496 seconds for the 'former method', and it is 26.142 seconds for the 'proposed method' which is about 1/18 comparing with the 'former method'.

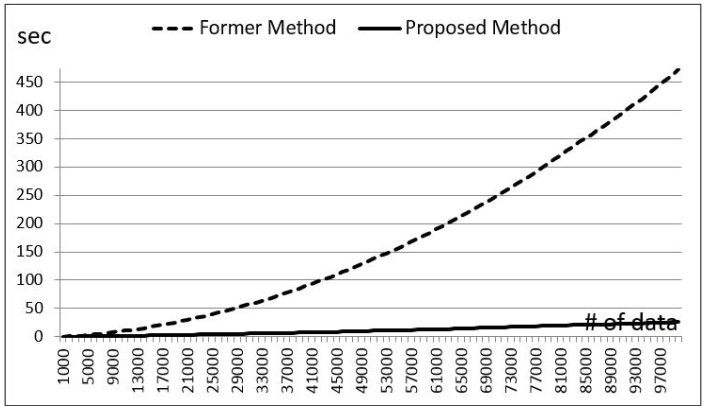

Fig. 4. Cumulative processing time of former method versus proposed method.

5. Conclusion

In this study, we propose a very simplified combining method for the item-based and user-based methods used in movie recommendation. Due to the small number of parameters to be calculated, this method is relatively free from the problem of sparsity, and it can predict new cases more accurately using the user's rating to calculate the parameters. In the experimental results, the performance of the proposed method is quickly stabilized though it is not accurate in the early stage. The average error of the proposed method is 6% lower than the former method that uses similarity measures. When compared in terms of processing speed, it is 18 times faster than the former method in predicting 100,000 cases while learning each case. The graph for the processing time shows that the difference in speed compared to the former method, which is getting larger as more data are accumulated. Our recommendation method can be applied to any type of item in e-Commerce as well as movies. However, the processing speed is very important as well as the accuracy to apply the method to a common e-Commerce system with a vast

number of users and items. In this respect, the recommendation method proposed in this paper has obvious advantages. It is necessary to apply this method to various recommendation systems in the future.

Acknowledgement

This research was supported by a Research and Development of Dual Operating System Architecture with High-Reliable RTOS and High-Performance funded by ETRI (No. R0101-16-0081).

References

[1] S. H. Jo, "Weight recommendation technique based on item quality to improve performance of new user recommendation and recommendation on the web," Ph.D. dissertation, Hannam University, Daejeon, Korea, 2008.

[2] S. J. Lee, T. R. Jeon, G. D, Baek, and S. S. Kim, "A movie rating prediction system of user propensity analysis based on collaborative filtering and fuzzy system," *Journal of Korean Institute of Intelligent Systems*, vol. 19, no. 2, pp. 242-247, 2009.

[3] H. C. Lee, S. J. Lee, and S. O. Kim, "A study on improvements of prediction accuracy using additional information in collaborative filtering," in *Proceedings of the Korean Accounting Association 2009 Spring Conference*, Seoul, Korea, 2009, pp. 349-352.

[4] G. Lekakos and G. M. Giaglis, "Improving the prediction accuracy of recommendation algorithms: approaches anchored on human factors," *Interacting with Computers*, vol. 18, no. 3, pp. 410-431, 2006.

[5] K. R. Kim, J. H. Byeon, and N. M. Moon, "Collaborative filtering design using genre similarity and preffered genre," *Journal of the Korea society of Computer and Information*, vol. 16, no. 4, pp. 159-168, 2011.

[6] H. Ma, I. King, and M. R. Lyu, "Effective missing data prediction for collaborative filtering," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Amsterdam, The Netherlands, 2007, pp. 39-46.

[7] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," in *Proceeding of the 1994 ACM Conference on Computer Supported Cooperative Work*, Chapel Hill, NC, 1994, pp. 175-186.

[8] J. Wang, A. P. de Vries, and M. J. Reinders, "Unifying user-based and item-based collaborative filtering approaches by similarity fusion," in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, WA, 2006, pp. 501-508.

[9] G. R. Xue, C. Lin, Q. Yang, W. Xi, H. J. Zeng, Y. Yu, and Z. Chen, "Scalable collaborative filtering using cluster-based smoothing," in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Salvador, Brazil, 2005, pp. 114-121.

[10] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 89-115, 2004.

[11] D. M. Pennock, E. Horvitz, S. Lawrence, and C. L. Giles, "Collaborative filtering by personality diagnosis: a hybrid memory- and model-based approach," in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, Stanford, CA, 2000, pp. 473-480.

[12] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the 14th conference on Uncertainty in Artificial Intelligence*, Madison, WI, 1998, pp. 43-52.

[13] D. S. Park, "Improved movie recommendation system based-on personal propensity and collaborative filtering," *KIPS Transactions of Computer and Communication System*, vol. 2, no. 11, pp. 475-482, 2013.

[14] S. C, Oh and M. Choi, "Effective combination of user-based and item-based methods for movie recommendation," in *Proceedings of the 2013 Korean Society of Internet Information (KSII) Fall Conference*, Seoul, Korea, 2013, pp. 135-136.

[15] GroupLens, "MovieLens datasets," [Online]. Available: http://www.grouplens.org/node/73.

Se-Chang Oh https://orcid.org/0000-0003-0899-7207

He received the M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST) in 1990 and 1997, respectively. He worked at LG Corporation Institute of Technology for four years and worked at Ajou University for three years. Now he is a professor in Department of Computer Software at Sejong Cyber University. His research interests include machine learning, pattern recognition and data science.

Min Choi https://orcid.org/0000-0002-9204-5665

He received the M.S. and Ph.D. degrees in Computer Science from the Korea Advanced Institute of Science and Technology (KAIST) in 2003 and 2009, respectively. From 2008 to 2010, he worked for Samsung Electronics as a Senior Engineer. Since 2011, he has been a faculty member of Department of Information and Communication of Chungbuk National University. His current research interests include embedded system, microarchitecture, and cloud computing.