Paper:

Proposal of Personal Value-Based User Modeling Without Attribute Evaluation

Kaichi Nihira*1, Hiroki Shibata*2, and Yasufumi Takama*3

*123Graduate school of System Design, Tokyo Metropolitan University
6-6, Asahigaoka, Hino, Tokyo,
191-0065, Japan
E-mail : *1nihira-kaichi@ed.tmu.ac.jp, *2 hshibata@tmu.ac.jp, *3 ytakama@tmu.ac.jp

Abstract. This paper proposes a personal values modeling method that does not require attribute ratings. The proposed method is applied to memory-based collaborative filtering (CF) to demonstrate its effectiveness. Personal values modeling using rate matching rate (RMRate) has been proposed, which needs an evaluation of each item attributes in addition to the rating value to items. Therefore, it cannot be used when the evaluation of item attributes is not available. To obtain personal values model of users only from a rating matrix, this paper defines user's personal values as their tendency to select popular/unpopular items. Five attributes are proposed for modeling user's personal values, of which effectiveness is evaluated in combination with user-based CF. Experimental results with three datasets show the proposed method can achieve higher precision and recall than conventional user-based CF.

Keywords: Recommendation, Personal Values, Collaborative Filtering

1. Introduction

Recent advances in information technology have made it easier to disseminate information than before. In addition, collecting various information has also become easy owing to huge computer storage and faster high-speed network. Although it has brought great benefits, it has also caused new problems that it is difficult to obtain appropriate information from a large amount of information. To solve this problem, research of recommendation has been conducted to suggest appropriate information to users.

One of the challenges of recommendation system is to recommend niche items (long tail items or unpopular items). With the recent growth of e-commerce, the long tail theory has become important to online retailers: the sales volume of niche items accounts for a non-negligible proportion of the total sales [1]. Compared to brick-and-mortar retail stores, online retailers do not need space to store items while reducing product management costs, which allows them to offer a wider selection of products. Therefore, recommending niche items is expected to take such advantages of online retailers.

Collaborative filtering (CF), which is one of the most common recommendation approaches, can recommend items without considering item characteristics. However, it is known to have the cold-start problem and the sparsity problem. The cold-start problem is the difficulty of making appropriate recommendations to new users or the difficulty of recommending new items [2]. The sparsity problem is caused by the fact that users evaluate only a few of available items [3]. As a result, it is difficult to estimate relationship between users and items from such a sparse rating matrix. These two problems should be solved to improve the accuracy of recommendation.

As a solution for such problems, this paper focuses on personal values, which is supposed to have the strong relation to potential consumer preferences. In order to define personal values quantitatively, the rate matching rate (RMRate) has been proposed [4]. The effectiveness of employing RMRate has been shown by applying it to content-based filtering [4], neighborhood-based CF [5]. However, RMRate has the problem that it cannot be used for datasets that does not have the evaluation of item attributes.

To extend the applicability of the personal values-based approach, this paper defines user's personal values as their tendency to select popular/unpopular items. Five attributes are proposed for modeling user's personal values based on this definition, which can be calculated only from a rating matrix. The proposed method is evaluated by applying it to user-based CF. Experimental results with three datasets show that the proposed method outperforms the conventional user-based CF in terms of precision and recall.

This paper is organized as follows. Related work about introduction of personal values into recommendation algorithms is introduced in Section2. Section3 proposes personal value-based user modeling method that does not require attributes ratings. The effectiveness of the proposed method is shown based on experimental results with three datasets in Section 4.

2. Related Works

2.1. Personal Value-based Approach

Personal values are known as factors that affect user's decision making on evaluating/purchasing items. Hattori et al. have defined personal values as the effect of user's evaluation of item attributes on their evaluation (rating) of items in the context of recommendation [4]. When a target item is a movie, its attributes are story, cast, music, etc. If a user always highly evaluates the story of their favorite movies, a story is supposed to strong effect on their decision making. To define such personal values quantitatively, RMRate has been proposed. It is defined as Eq. (1).

$$RMR_{ik} = \frac{\sum_{x_j \in I(u_i)} \delta(p_{ij}, p_{ij}^k)}{|I(u_i)|}, \quad \ldots\ldots\ldots \quad (1)$$

where $I(u_i)$ is a set of items rated by user u_i, $p_{ij} \in \{pos, neg\}$ is the polarity of total evaluation (rating) of u_i on item x_j, $p_{ij}^k \in \{pos, neg\}$ is the polarity of the evaluation of u_i on the k-th attribute of x_j. If the rating or the evaluation of an item attribute is more than a predefined threshold, it is regarded as *pos*, and *neg* otherwise. The function $\delta(p_{ij}, p_{ij}^k)$ returns 1 if p_{ij} is equal to p_{ij}^k, and 0 otherwise. It has been reported that this method can estimate the values with small number of ratings [4].

Misawa et al. have proposed user-based neighborhood CF employing the RMRate [5]. This method represents a user's characteristics as the vector of RMRate and calculates an inter-user similarity based on the Pearson correlation coefficient (Eq. (2)).

$$\rho_{it} = \frac{\sum_{a_l \in A} (v_{il} - \bar{v}_i)(v_{tl} - \bar{v}_t)}{\sqrt{\sum_{a_l \in A} (v_{il} - \bar{v}_i)^2} \sqrt{\sum_{a_l \in A} (v_{tl} - \bar{v}_t)^2}}, \quad \ldots \quad (2)$$

where $a_l \in A$ is an attribute of items, v_{il} is the RMRate of a_l given by u_i, and $\bar{v}_i$ is the average RMRate of all attributes given by u_i.

After calculating the similarities, the predicted rating $\hat{r}_{ij}$ of user u_i on item x_j is calculated by Eq. (3).

$$\hat{r}_{ij} = \frac{\sum_{u_\tau \in U(x_j)} \rho_{i\tau} r_{\tau j}}{\sum_{u_\tau \in U(x_j)} \rho_{i\tau}}, \quad \ldots\ldots\ldots\ldots \quad (3)$$

where $U(x_j)$ is the set of u_i's neighborhood users who evaluated x_j, and $r_{\tau j}$ is the rating of u_τ on x_j. It is reported that this method can recommend long tail items while maintaining the precision of the recommendation [5].

While it is shown that recommender systems employing the RMRate are effective for mitigating the cold-start problem [4] and for recommending long tail items [5], there is a problem that RMRate cannot be calculated on datasets that do not have the evaluation of item attributes.

3. Personal Value-based User Modeling Without Attribute Evaluation

This paper proposes a method for obtaining personal values model of users only from a rating matrix. The proposed method defines user's personal values as their tendency to select popular/unpopular items. Based on this definition, the following five attributes are proposed instead of the RMRate.

1. Ratio of popular items among all items rated by a user
2. Average number of ratings of popular items rated by a user
3. Average number of ratings of unpopular items rated by a user
4. Average rating of popular items rated by a user
5. Average rating of unpopular items rated by a user

The procedure of the proposed method is mainly composed of three processes: getting personal values, calculating inter-user similarity, and predicting the rating value. These are described in the subsequent subsections.

3.1. Getting Personal Values from the Rating Matrix

Popularity and unpopularity of an item is determined based on Pareto principle, which is also known as the 80:20 principle. In this paper, the top 20% items in terms of the number of ratings are classified as popular, and the other items are classified as unpopular items. The sets of popular items I_{pop} and unpopular items I_{unpop} are defined as Eq. (3), (4), respectively.

$$I_{pop} = \{x_j | x_j \in I, n_j \geq n\star\}, \quad \ldots\ldots\ldots \quad (4)$$

$$I_{unpop} = \{x_j | x_j \in I, n_j < n\star\}, \quad \ldots\ldots\ldots \quad (5)$$

where I is a set of all items, n_j is number of ratings of item x_j, $n\star$ is the number of ratings of the $\lfloor 0.2|I| \rfloor$-th item in the descending order of the number of ratings.

The above-mentioned five attributes of user u_i are defined by Eq. (6), (7), (8), (9), (10), respectively.

$$b_{i1} = \frac{\sum_{x_j \in I(u_i)} f(x_j)}{|I(u_i)|}, \quad \ldots\ldots\ldots\ldots \quad (6)$$

$$b_{i2} = \frac{\sum_{x_j \in X_{pop}} n_j}{|X_{pop}|}, \quad \ldots\ldots\ldots\ldots \quad (7)$$

$$b_{i3} = \frac{\sum_{x_j \in X_{unpop}} n_j}{|X_{unpop}|}, \quad \ldots\ldots\ldots\ldots \quad (8)$$

$$b_{i4} = \frac{\sum_{x_j \in X_{pop}} r_{ij}}{|X_{pop}|}, \quad \ldots\ldots\ldots\ldots \quad (9)$$

$$b_{i5} = \frac{\sum_{x_j \in X_{unpop}} r_{ij}}{|X_{unpop}|}, \quad \ldots\ldots\ldots\ldots (10)$$

where $X_{pop/unpop}$ is the intersection of $I_{pop/unpop}$ and $I(u_i)$, the function $f(x_j)$ returns 1 if x_j belongs to I_{pop}, and 0 otherwise. b_{i2} and b_{i4} become missing value $\perp$ if $X_{pop}(u_i)$ is empty, and b_{i3} and b_{i5} become $\perp$ if $X_{unpop}(u_i)$ is empty. Since the proposed attributes have different ranges, min-max normalization is applied (Eq. (11)).

$$b'_{im} = \frac{b_{im} - \min B_m}{\max B_m - \min B_m}, \quad \ldots\ldots\ldots\ldots (11)$$

where B_m is defined as Eq. (12).

$$B_m = \{b_{\tau m} : u_\tau \in U\} - \{\perp\}. \quad \ldots\ldots\ldots\ldots (12)$$

3.2. Calculation of Inter-user Similarity : Pearson Correlation and Cosine Similarity

To calculate the inter-user similarity, the proposed method uses two types of similarity measures: the Pearson correlation coefficient and cosine. Using the Pearson correlation coefficient, the similarity between u_i and u_t is calculated by Eq. (13).

$$\rho^P_{it} = \frac{\sum_{m \in M_{it}} (b'_{im} - \bar{b}'_i)(b'_{tm} - \bar{b}'_t)}{\sqrt{\sum_{m \in M_{it}} (b'_{im} - \bar{b}'_i)^2} \sqrt{\sum_{m \in M_{it}} (b'_{tm} - \bar{b}'_t)^2}}, \quad \ldots (13)$$

where M_{it} is an index set of the proposed attributes satisfying that $b_{im} \neq \perp$ and $b_{tm} \neq \perp$. $\bar{b}'_i$ is defined as Eq. (14).

$$\bar{b}'_i = \sum_{m \in M_{it}} \frac{b'_{it}}{|M_{it}|}. \quad \ldots\ldots\ldots\ldots (14)$$

The cosine-based similarity is calculated by Eq. (15).

$$\rho^C_{it} = \frac{\sum_{m \in M_{it}} b'_{im} b'_{tm}}{\sqrt{\sum_{m \in M_{it}} {b'_{im}}^2} \sqrt{\sum_{m \in M_{it}} {b'_{tm}}^2}}. \quad \ldots\ldots\ldots\ldots (15)$$

3.3. Rating prediction

The predicted rating $\hat{r}_{ij}$ for u_i on x_j is calculated by Eq. (16).

$$\hat{r}_{ij} = \bar{r}_i + \frac{\sum_{u_\tau \in U(x_j)} \rho_{i\tau}(r_{\tau j} - \bar{r}_\tau)}{\sum_{u_\tau \in U(x_j)} \rho_{i\tau}}, \quad \ldots\ldots\ldots\ldots (16)$$

where $\bar{r}_i$ is average ratings of u_i and $\rho_{i\tau}$ is either $\rho^P_{i\tau}$ or $\rho^C_{i\tau}$.

Table 1. Overview of datasets.

Dataset	Range	Step size	Item
Movie Lens (ML)	1-5	1	movie
Yahoo! Movie (YM)	1-5	1	movie
4Travel (FT)	1-5	0.5	accommodation

Table 2. Summary of datasets.

Dataset	User	Item	Evaluation	Density
ML	6,040	3,706	1,000,209	0.04468
YM	11,465	6,827	473,339	0.00605
FT	16,578	12,039	90,111	0.00045

4. Experiment

4.1. Datasets

The proposed method is evaluated using 3 datasets: MovieLens[1], Yahoo! Movie[2] and 4Travel[3]. MovieLens dataset is one of the most popular datasets used for evaluating recommender systems. The other two datasets are generated from Japanese review sites. The overview of these datasets is shown in **Table 1**.

As for preprocessing, users who rated less than 20 ratings were removed from MovieLens dataset. Those who had less than 10 ratings were removed from Yahoo! Movie dataset. To evaluate the effectiveness of the proposed method for cold-start problem, no user was removed from 4Travel dataset. **Table 2** shows the statistics of the datasets after the preprocessing. Compared with MovieLens dataset, the other datasets are smaller but much more sparse. Due to the sparsity problem, it is expected that performance of recommendation for those datasets is lower than that for MovieLens dataset.

Table 3 and **Table 4** respectively show the number of users and items per the number of ratings. MovieLens dataset has many users/items with many ratings. Yahoo! Movie dataset has much users with few ratings than MovieLens dataset. Most of users and items in 4travel dataset have 10 or less ratings. Due to the cold-start problem, it is expected that performance of recommendation decreases for those datasets compared with MovieLens dataset.

4.2. Outline

The proposed method is compared with the conventional user-based CF [6]. Corresponding to the similarity calculation of the proposed method (Eq. (13) and (15)), both the Pearson correlation coefficient and cosine are employed. The similarity between u_i and u_t is calculated

1. https://www.movielens.org/
2. https://movies.yahoo.co.jp/
3. https://4travel.jp/

Table 3. Number of users per number of ratings. 2-4 columns show the range of the number of ratings. ML: Movie Lens, YM: Yahoo! Movie, FT: 4Travel.

Dataset	1–10	11–20	21–40	41–
ML	0 (0.0%)	86 (1.4%)	1,265 (20.9%)	4,716 (77.6%)
YM	1,053 (9.2%)	4,933 (43.0%)	2,818 (24.6%)	2,661 (23.2%)
FT	14,984 (90.4%)	905 (5.5%)	398 (2.4%)	291 (1.8%)

Table 4. Number of items per number of ratings. 2-4 columns show the range of the number of ratings.

Dataset	1–10	11–20	21–40	41–
ML	473 (12.8%)	222 (6.0%)	358 (9.7%)	2,653 (71.6%)
YM	663 (9.7%)	1,837 (26.9%)	1,648 (24.1%)	2,679 (39.2%)
FT	9,856 (81.9%)	1,242 (10.3%)	628 (5.2%)	313 (2.6%)

by Eq. (17).

$$\rho_{it}^{P2} = \frac{\sum_{x_j \in X_{it}} (r_{ij} - \bar{r}'_i)(r_{tj} - \bar{r}'_t)}{\sqrt{\sum_{x_j \in X_{it}} (r_{ij} - \bar{r}'_i)^2} \sqrt{\sum_{x_j \in X_{it}} (r_{tj} - \bar{r}'_t)^2}}, \quad \ldots (17)$$

where X_{it} is the intersection of $I(u_i)$ and $I(u_t)$, $\bar{r}'_i$ is the average rating of items in X_{it}.

The cosine-based similarity is calculated by Eq. (18).

$$\rho_{it}^{C2} = \frac{\sum_{x_j \in X_{it}} r_{ij} r_{tj}}{\sqrt{\sum_{x_j \in X_{it}} r_{ij}^2} \sqrt{\sum_{x_j \in X_{it}} r_{tj}^2}}. \quad \ldots\ldots\ldots (18)$$

Predicted rating is calculated with Eq. (16).

The experiment supposes top-k recommendation task and evaluates the performance by P@k (precision at k, Eq. (19)), R@k (recall at k, Eq. (20)), and D@k (diversity at k, Eq. (21)).

$$\mathrm{P@}k = \frac{1}{|S_U|} \sum_{u_i \in S_U} \frac{\sum_{j=1}^{k} s(u_i, x_j)}{k}, \quad \ldots\ldots\ldots (19)$$

$$\mathrm{R@}k = \frac{1}{|S_U|} \sum_{u_i \in S_U} \frac{\sum_{j=1}^{k} s(u_i, x_j)}{|I^+(u_i)|}, \quad \ldots\ldots\ldots (20)$$

$$\mathrm{D@}k = \left| \bigcup_{u_i \in S_U} \hat{I}^k(u_i) \right|, \quad \ldots\ldots\ldots\ldots (21)$$

where S_U is the set of users in the test data, the function $s(u_i, x_j)$ returns 1 if r_{ij} is more than a predefined threshold, and 0 otherwise. The threshold is set to 4 for MovieLens dataset and Yahoo! Movie dataset, and 3.5 for 4Travel dataset. $I^+(u_i)$ is the set of items u_i gave higher rating than the threshold. $\hat{I}^k(u_i)$ is the set of k items recommended to u_i. The k is set to 3.

Let NP be the proposed method that uses Eq. (13) for user similarity, NC be the proposed method that uses Eq. (15), OP be the conventional method that uses Eq. (17), and OC be the conventional method that uses Eq. (18).

The experiment is carried out by repeating double cross validation, which is described as follows.

(1) Divide a dataset D into 3 subsets D_i.

(2) Do the following process for all combinations of hyperparameters.
   (a) Divide a dataset $D' = D \backslash D_i$ into 3 subsets D'_j.
   (b) Calculate the RMSE of the predicted rating of the validation data D'_j using $D'' = D' \backslash D'_j$ as training data. The similarity method is either OP or OC.
   (c) Calculate the average RMSE over D'_1, D'_2, D'_3, which is denoted as $\overline{\mathrm{RMSE}}$.

(3) With the hyperparameters that achieve the minimum $\overline{\mathrm{RMSE}}$ in (2), calculate the predicted rating of test data D_i using D' as training data. If OP is used in (2)(b), either NP or OP are used for prediction; if OC is used, either NC or OC are used.

(4) Do the top-k recommendation and calculate P@k, R@k, and D@k.

(5) Calculate the average P@k, R@k, and D@k over D_1, D_2, D_3, which are respectively denoted as $\overline{\mathrm{P@3}}$, $\overline{\mathrm{R@3}}$, and $\overline{\mathrm{D@3}}$, as final result.

There are two types of hyperparameters which related to Eq. (15): α and β respectively specifies the maximum and minimum number of similar users to be included in $U(x_j)$. If the size of $U(x_j)$ is smaller than β, the second term on the right-hand side of Eq. (15) is set to 0.

4.3. Result

Tables 5-7 respectively show $\overline{\mathrm{P@3}}$, $\overline{\mathrm{R@3}}$, $\overline{\mathrm{D@3}}$ for MovieLens, Yahoo! Movie, and 4travel datasets. In these tables, the results in boldface indicate the best performance among the four methods. It is shown that for all datasets, the proposed method (OP and NP) achieved higher $\overline{\mathrm{P@3}}$ and $\overline{\mathrm{R@3}}$ than the conventional methods. However, $\overline{\mathrm{D@3}}$ of the proposed methods were lower than the conventional methods. When comparing the result of different datasets, $\overline{\mathrm{P@3}}$ in MovieLens dataset is higher than other datasets regardless of the method used. It is supposed that the reason for this result is due to its higher density than other datasets.

Table 5. Experimental result (Movie Lens). OP: conventional method with Eq. (17), NP: proposed method with Eq. (13), OC: conventional method with Eq. (18), NC: proposed method with Eq. (15).

Method	$\overline{\text{P@3}}$	$\overline{\text{R@3}}$	$\overline{\text{D@3}}$
OP	0.8343	0.1891	**1,018**
NP	0.8354	0.1900	958
OC	0.8207	0.1863	954
NC	**0.8367**	**0.1904**	951

Table 6. Experimental result (Yahoo! Movie).

Method	$\overline{\text{P@3}}$	$\overline{\text{R@3}}$	$\overline{\text{D@3}}$
OP	0.6654	0.5061	**4,582**
NP	0.6891	0.5267	4,350
OC	0.6860	0.5236	4,374
NC	**0.6900**	**0.5277**	4,319

Table 7. Experimental result (4Travel).

Method	$\overline{\text{P@3}}$	$\overline{\text{R@3}}$	$\overline{\text{D@3}}$
OP	0.5371	0.7819	**3,485**
NP	0.5462	**0.7879**	3,184
OC	0.5418	0.7842	3,285
NC	**0.5464**	0.7879	3,148

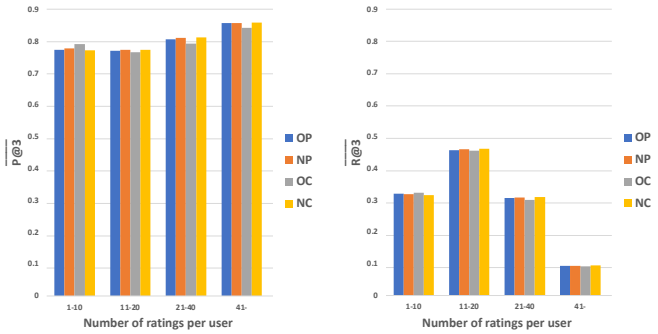


Fig. 1. $\overline{\text{P@3}}$ and $\overline{\text{R@3}}$ for MovieLens dataset. OP: conventional method with Eq. (17), NP: proposed method with Eq. (13), OC: conventional method with Eq. (18), NC: proposed method with Eq. (15).

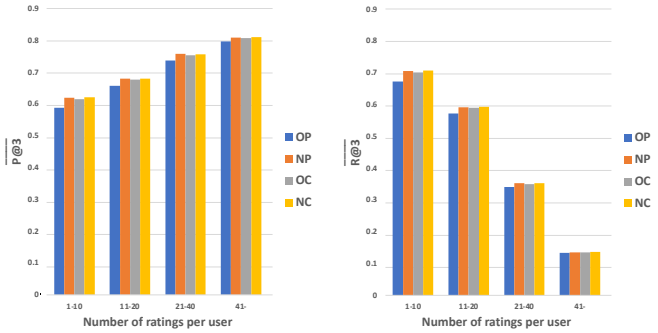


Fig. 2. $\overline{\text{P@3}}$ and $\overline{\text{R@3}}$ for Yahoo! Movie dataset.

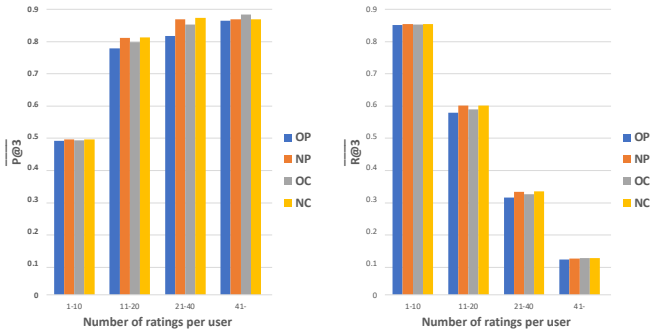


Fig. 3. $\overline{\text{P@3}}$ and $\overline{\text{R@3}}$ for 4travel dataset.

To examine the effect of the number of ratings a user gave on the recommendation performance, users are grouped based on their number of ratings, and $\overline{\text{P@3}}$ and $\overline{\text{R@3}}$ are calculated for each group. **Figure 1-3** respectively show the result for MovieLens, Yahoo! Movie, and 4travel datasets. The figures have the number of ratings per user on the horizontal axis and $\overline{\text{P@3}}$ or $\overline{\text{R@3}}$ on the vertical axis. **Figure 1** shows that $\overline{\text{P@3}}$ of NP were higher than OP for users with 1 to 40 ratings, and that of NC were higher than OC for users with more than 10 ratings in MovieLens dataset. $\overline{\text{R@3}}$ of the proposed methods were higher than the conventional methods for users with 11 to 40 ratings. **Figure 2** shows that $\overline{\text{P@3}}$ of the proposed methods were higher than the conventional methods regardless of the number of ratings in Yahoo! Movie dataset. $\overline{\text{R@3}}$ of the proposed methods were higher than or equal to the conventional methods regardless of the number of ratings. **Figure 3** shows that $\overline{\text{P@3}}$ of NP were higher than OP regardless of the number of ratings, and that of NC were higher than OC for users with 1 to 40 ratings in 4travel dataset. $\overline{\text{R@3}}$ of the proposed methods were higher than or equal to the conventional methods regardless of the number of ratings.

Summarizing the results, the proposed methods tend to be more effective than the conventional methods regardless of the number of ratings per user. As the common tendency of both the proposed and conventional methods, the precision decreases as the number of ratings per user decreases. Even so, the proposed methods achieved higher precision than the conventional methods when the number of ratings per user is small. Therefore, it is supposed that the proposed methods are able to reduce the cold-start problem to some extent.

To examine what items are recommended, items are grouped based on their number of ratings, and the frequency of recommending items in each group is counted. **Table 8-10** respectively show the results for MovieLens, Yahoo! Movie, and 4travel datasets. **Table 8** shows that the proposed methods tend to recommend popular items than the conventional methods. However, **Table 9** shows

Table 8. Frequency each item was recommended (Movie Lens). 2-4 columns show the range of the number of ratings. OP: conventional method with Eq. (17), NP: proposed method with Eq. (13), OC: conventional method with Eq. (18), NC: proposed method with Eq. (15).

Method	1–10	11–20	21–40	41–
OP	43	**55**	101	17,914
NP	39	31	78	17,965
OC	**58**	44	**103**	17,909
NC	42	24	71	**17,976**

Table 9. Frequency each item was recommended (Yahoo! Movie). 2-4 columns show the range of the number of ratings.

Method	1–10	11–20	21–40	41–
OP	840	1,859	3,310	**27,175**
NP	1,300	**2,231**	**3,450**	26,248
OC	1,177	2,135	3,421	26,450
NC	**1,309**	2,180	3,371	26,324

Table 10. Frequency each item was recommended (4Travel). 2-4 Columns show the range of the number of ratings.

Method	1–10	11–20	21–40	41–
OP	**5,101**	2,332	2,171	2,364
NP	5,043	**2,355**	2,222	2,348
OC	4,847	**2,335**	**2,272**	**2,514**
NC	4,974	**2,355**	2,218	2,422

that the proposed methods tend to recommend unpopular items than the conventional methods. Furthermore, **Table 10** also shows that NC tends to recommend unpopular items than OC.

As shown in Table 5-7, $\overline{\text{D@3}}$ of the proposed methods was worse than the conventional methods. However, it was found that the proposed method tended to recommend unpopular items for some datasets.

5. Conclusion

This paper proposed a personal values modeling method that does not require attribute ratings. This paper defines a personal values as users tendency to select popular/unpopular items and applied it to memory-based CF. The proposed personal values can be applied to datasets that do not have the ratings of item attributes, which is essential advantage. Experiments using 3 datasets of actual review sites were conducted to compare the proposed methods with the conventional methods using Pearson correlation coefficient and cosine as inter-user similarity. As an overall tendency, $\overline{\text{P@3}}$ and $\overline{\text{R@3}}$ of the proposed methods were comparable to the conventional methods. On the other hand, $\overline{\text{D@3}}$ of the proposed methods was lower than the conventional methods. Looking at the number of ratings, the proposed methods tend to achieve higher $\overline{\text{P@3}}$ and $\overline{\text{R@3}}$ than the conventional methods for users who rated a small number of items. It was also confirmed that the proposed methods tended to recommend unpopular items for some datasets.

As future works, we plan to add new attributes that can be calculated only from rating matrix. In this paper, the number of ratings was used to classify items into popular/unpopular ones. Another idea is to judge popular/unpopular based on the average ratings of an item.

Another future direction is to introduce the proposed attributes to matrix-based CF. It has been reported the matrix-based CF with RMRate [7] achieved comparable recommendation accuracy to typical matrix-based CFs while recommending more unpopular items. It is expected that those benefits can be obtained by applying the proposed attributes to matrix-based CF.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Numbers 19K22896, 21H03553, 22H03698.

References:

[1] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and Metrics for Cold-Start Recommendations," Proc. of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 253-260, 2002.

[2] C. Anderson, "The Long Tail: Why the Future of Businesses Selling less of more," Hachette Books, 2006.

[3] S. Lee, J. Yang, and S. Y. Park, "Discovery of Hidden Similarity on Collaborative Filtering to Overcome Sparsity Problem," Proc. of Discovery Science: 7th International Conference, pp. 396-402, 2004.

[4] S. Hattori, and Y. Takama, "Recommender System Employing Personal-Value-Based User Model," J. of Advanced Computational Intelligence and Intelligent Informatics, Vol.18, No.2, pp. 157-165, 2014.

[5] Y. Takama, Y. Chen, R. Misawa, and H. Ishikawa, "Analyzing Potential of Personal Values-Based User Modeling for Long Tail Item Recommendation," J. of Advanced Computational Intelligence and Intelligent Informatics, Vol.22, No.4, pp. 506-513, 2018.

[6] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," Proc. of the Conference on Computer Supported Cooperative Work, pp. 175-186, 1994.

[7] Y. Takama, H. Shibata, and Y. Shiraishi, "Matrix-Based Collaborative Filtering Employing Personal Values-Based Modeling and Model Relationship Learning," J. of Advanced Computational Intelligence and Intelligent Informations, Vol.24, No.6, pp. 719-727, 2020.