An Enhanced Memory-Based Collaborative Filtering Algorithm based on User Similarity for Recommender Systems

Ramil G. Lumauag, Ariel M. Sison, Ruji P. Medina

Abstract--- The determination of user similarity in a memory-based collaborative filtering is the most crucial part of the process since the result of this will greatly influence the prediction rating in generating an accurate and valuable recommendation. This paper presents an enhanced memory-based collaborative algorithm by formulating a similarity measure to identify the number of co-rated items, compute user similarity by selecting the nearest neighbor. The experimental results on dataset show that the proposed algorithm decreases the Mean Absolute Error and improves the accuracy of the algorithm.

Keywords—collaborative filtering algorithm, memory-based collaborative filtering, recommender systems, user similarity

I. INTRODUCTION

Personal computers, mobile devices, and other intelligent systems already dominate today's modern society. With the availability of huge information over the Internet, people tend to spend more time to search for their expected information in various platforms like social media, e-commerce, and streaming media sites.

With these trends, Recommender Systems (RS) were developed by several online platforms to track user behavior and analyze their preferences to provide personalized services. With the availability of a lot of information over the Internet that causes an information overload problem, a technique to deal and create a personalized recommendation for customers to add dimension to user experience can now be managed by RS [1]. Online sites such as Amazon, Netflix, eBay, etc. use RS that are very useful in recommending items or products to the user according to their interests.

Collaborative filtering algorithms are used by recommender systems to find users with similar tastes and suggest to these related users items that were mostly selected by them [2]. In collaborative filtering approach, the system analyzed the items liked by both users and recommends new items to similar users [3].

There are two methods of collaborative filtering as cited by [4]: the memory-based method and model-based method. Memory-based method works by computing the user similarities, then select the most similar users based on the active users' neighbors, and compute the similarity scores to generate prediction and give the top N recommendations according to the predicted value, while the Model-based method uses a constructed model to describe the behavior of the users and predicts the ratings of the items.

The main focus of this paper is the memory-based collaborative filtering algorithm specifically the user-based approach that utilizes the entire user-item database to calculate user similarities based on neighborhood ratings. Recommender systems commonly used memory-based collaborative filtering algorithm due to its simplicity and high-quality predictions [5]. However, data sparsity [6] and overfitting [7] are among the drawbacks associated with the memory-based collaborative filtering.

Data sparsity occurs when most of the items in the user-item rating matrix where not rated by the users, this will cause inaccuracy that affects the quality and efficiency of the recommendation. Overfitting occurs when most of the users only have interest in very few items.

Therefore, the aim of this paper is to enhance a memory-based collaborative filtering algorithm based on user similarity on co-rated items by identifying the co-rated items of the active and similar users to remove noise and eliminate dissimilar users and solve data sparsity; compute the user similarity that will evaluate the variation of ratings among users to identify common ratings; select the nearest neighbors with significant contribution in the recommendation process.

II. RELATED WORK

A. Collaborative Filtering Process

In collaborative filtering, the rating data of the active user is compared to rating data of the other users using the user-item matrix (rating table). The user rating similarity is calculated using similarity measures and the prediction score is obtained to predict the item to be recommended to the active user. Fig. 1 shows the collaborative filtering process.

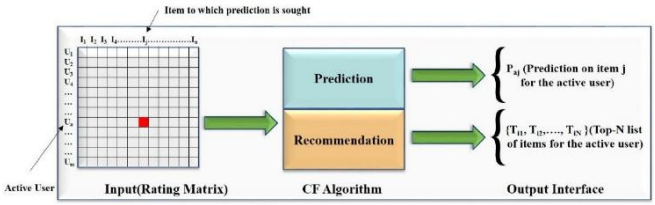

Fig. 1. Collaborative filtering process.

Revised Manuscript Received on March 08, 2019.

Ramil G. Lumauag, Technological Institute of the Philippines, Quezon City Campus, Quezon City, Philippines. (E-Mail: ramilglumauag@gmail.com)

Ariel M. Sison, Emilio Aguinaldo College, Manila, Philippines. (E-Mail: ariel.sison@eac.edu.ph)

Ruji P. Medina, Technological Institute of the Philippines, Quezon City Campus, Quezon City, Philippines.(E-mail: ruji.medina@tip.edu.ph)

B. Similarity Measure

Collaborative filtering process utilizes similarity measure and it assumes that "similar users like similar items". Among the various similarity measures commonly used in collaborative filtering are Pearson Correlation and Cosine Distance [8].

- Pearson Correlation Coefficient is a similarity measures that determines the linear correlation between two variables X and Y with an inclusive value between +1 and −1, where 1 refers to a total positive correlation, 0 means no correlation, and −1 refers to a total negative correlation [9]. A positive high value represents a high correlation, while a negative high value represents an inversely high correlation, and lastly, the uncorrelated samples are indicated by zero correlation.

The equation of User-Based Pearson Correlation similarity is defined as:

$$s(u,v) = \frac{\sum_{i \epsilon I_{uv}} (r_{ui} - r'u)(r_{vi} - r'v)}{\sqrt{\sum_{i \epsilon I_{uv}} (r_{ui} - r'u)^2 \sum_{i \epsilon I_{uv}} (r_{vi} - r'v)^2}} \quad (1)$$

where s (u,v) is the similarity of users u and v, I_{uv} are the set of items rated by users u and v, r_{ui} and r_{vi} is the item's rating by user u or v, and r'u and r'v is the mean rating of user u or v on all items.

- Cosine similarity is a smilarity measure that determines the similarity between two vectors by getting the cosine of the angle of an inner product space. It is a measure of orientation and not magnitude: the similarity is measured if two vectors with the same orientation have a cosine of 1, a similarity of 0 if two vectors are at 90°, and 1 if two vectors diametrically opposed have a similarity independent of their magnitude [10]. A high correlation is represented by +1 and -1, while 0 represents no correlation.

The equation of User-Based Cosine similarity is defined as:

$$s(u,v) = \frac{\sum_{i \epsilon I_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \epsilon I_u} r_{ui}^2 \sum_{i \epsilon I_v} r_{vi}^2}} \quad (2)$$

where s (u, v) is the similarity of users u and v ratings on the items.

C. Evaluation Metrics

The evaluation metrics are used to determine the accuracy of the algorithm to predict the user's preference over a list of items. The following evaluation metrics are the standard metric used to evaluate recommender systems: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) [11].

The formula are defined as follows:

Mean Absolute Error:

$$MAE = \frac{1}{|Q|} \sum_{(u,i) \in Q} |r_{ui} - r'_{ui}| \quad (3)$$

Root Means Square Error:

$$RMSE = \sqrt{\frac{1}{|Q|} \sum_{(u,i) \in Q} (r_{ui} - r'_{ui})^2} \quad (4)$$

where Q is the test set, r_{ui} represents the user's true ratings, r'_{ui} represents the prediction ratings of the recommender system.

III. PROPOSED METHOD

A. The Enhancement

Fig. 2 illustrates the enhancement that focuses on the essential component of recommender systems which is determination of user similarity and selection of nearest neighbor. The enhancement contains three phases; In Phase 1, the numbers of the co-rated items are identified by determining the items rated by the active user and other users. This is done by sorting out the items rated by the active user. Next step is to group the co-rated items for easy identification. In a large and sparse matrix, it is difficult to locate co-rated items that are fragmented, in this step grouping of the co-rated item will be essential to easily identify them.

Phase 2 is the finding of similar users; the first step is to eliminate dissimilar users from the group of co-rated items. The exclusion of dissimilar user will remove the noisy users that are irrelevant in the process and will alleviate the data sparsity problem. The next step is the classification of the similar user; this process ensures that only significant users will be involved in the process. The next step is the computation of user similarity; in this step, the formulated similarity measure is used to determine the most similar users. The last step is to determine the similarity value derived from the computation to find the nearest neighbor.

Phase 3 is the finding of nearest neighbors; the nearest neighbors refer to the most similar users who are the influencing factor in the rating prediction. In this phase, the derived similarity value is sorted based on the highest possible value. Users with high similarity value are considered the most similar users; these are users with high positive correlation. The next step is to define the similarity threshold to determine the number of K nearest neighbor to restrict the number of bad neighbors, and lastly list down the most similar users.

The result of this enhancement process specifically the similarity value will be used to get the prediction rating and generate the top N recommendations. Fig. 3 illustrates the enhanced collaborative filtering process.

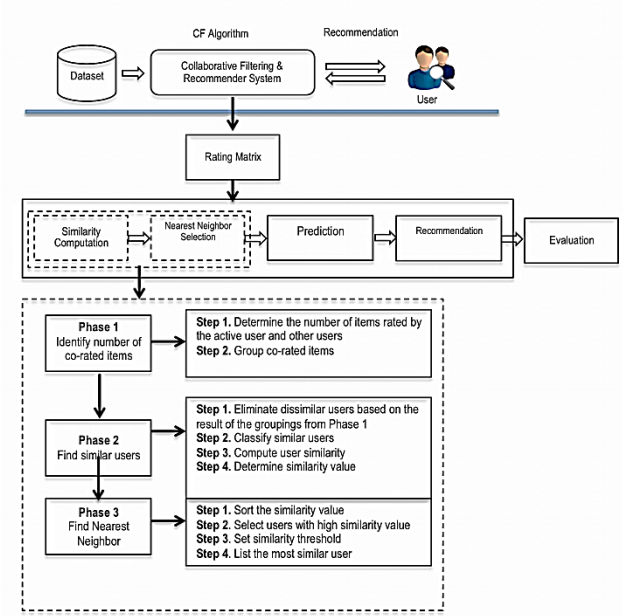

Fig. 3. Enhanced collaborative filtering process.

B. The Enhanced Algorithm

The enhanced algorithm is illustrated in the following steps:

Input: rating matrix of user-item
Output: recommendation in top N
Step 1: Read the user-item matrix
Step 2: Identify and group co-rated items
Step 3: Classify similar users and remove sparse ratings
Step 4: Compute user similarity
Step 5: Sort similarity value
Step 6: Select the most similar users
Step 7: Set the similarity threshold
Step 8: List nearest neighbor
Step 9: Compute prediction rating
Step 10: Produce top N recommendation

C. Processing the Rating Matrix

The first step is to read the user-item matrix and determine the common items that are rated by both the active user the other users. The rating matrix is shown in Table 1.

Table i. The rating matrix

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Item 9	Item 10
User 1	2	3	4	4	5	2	3	3	5	3
User 2	3	3	5	4	3	4	5	2	4	4
User 3	5	5	1	5	1	5	3	4	1	0
User 4	0	2	0	1	2	1	1	0	2	4
User 5	1	1	2	0	3	0	5	1	4	1
User 6	4	4	3	3	0	1	5	3	3	2
User 7	0	0	5	4	3	3	1	5	4	3
User 8	2	2	3	5	4	5	3	2	1	4
User 9	3	3	1	3	5	4	2	5	2	0
User 10	5	4	2	0	2	5	2	0	3	3
Active User	0	2	4	3	0	2	4	1	0	0

The table shows the users rating on the items with a numeric rating from 1-5. Zero value means that the user has not rated an item and it is considered as sparse data.

D. Identifying Number of Co-Rated Items

The number of co-rated items is identified in this phase. The items rated by the active user are determined by sorting out the items rated by the active user and grouping the co-rated items for easy identification. The identification of the co-rated items is done by getting the cardinality of the items rated by the active user and other users. The notation of the co-rated item is defined as:

$$CR = |I_U \cap I_v| \qquad (5)$$

where $|I_U \cap I_v|$ are the number of items, which user u and v commonly rated. Table 2 shows the grouping of co-rated items.

Table ii. Grouping of co-rated items

	Item 3	Item 7	Item 4	Item 2	Item 6	Item 8	Item 1	Item 5	Item 9	Item 10
User 1	4	3	4	3	2	3	2	5	5	3
User 2	5	5	4	3	4	2	3	3	4	4
User 3	1	3	5	5	5	4	5	1	1	0
User 4	0	1	1	2	1	0	0	2	2	4
User 5	2	5	0	1	0	1	1	3	4	1
User 6	3	5	3	4	1	3	4	0	3	2
User 7	5	1	4	0	3	5	0	3	4	3
User 8	3	3	5	2	5	2	2	4	1	4
User 9	1	2	3	3	4	5	3	5	2	0
User 10	2	2	0	4	5	0	5	2	3	3
Active User	4	4	3	2	2	1	0	0	0	0

E. Finding Similar Users

To find similar users, dissimilar users are removed from the group of co-rated items in order to remove the noisy users that are irrelevant in the process. The next step is the classification of the similar user; this process ensures that only significant users will be involved in the process.

Table 3 shows the rating matrix of similar users without the sparse data.

Table iii. Rating matrix of similar users

	Item 3	Item 7	Item 4	Item 2	Item 6	Item 8	Item 1	Item 5	Item 9	Item 10
User 1	4	3	4	3	2	3	2	5	5	3
User 2	5	5	4	3	4	2	3	3	4	4
User 3	1	3	5	5	5	4	5	1	1	0
User 6	3	5	3	4	1	3	4	0	3	2
User 8	3	3	5	2	5	2	2	4	1	4
User 9	1	2	3	3	4	5	3	5	2	0
Active User	4	4	3	2	2	1	0	0	0	0

The next step is the determination of the most similar users by computing of user similarity using the User-Based Pearson Correlation [10]. It is defined as:

$$s(u,v) = \frac{\sum_{i \epsilon I_{uv}} (r_{ui} - r'_u)(r_{vi} - r'_v)}{\sqrt{\sum_{i \epsilon I_{uv}} (r_{ui} - r'_u)^2 \sum_{i \epsilon I_{uv}} (r_{vi} - r'_v)^2}} \qquad (6)$$

Table 4 shows the result of the computation of user similarity using User-based Pearson Correlation.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Table iv. Result of user similarity computation

	Item 3	Item 7	Item 4	Item 2	Item 6	Item 8	PCC
User 1	4	3	4	3	2	3	0.51189
User 2	5	5	4	3	4	2	0.94176
User 3	1	3	5	5	5	4	-0.6528
User 6	3	5	3	4	1	3	0.41416
User 8	3	3	5	2	5	2	0.20146
User 9	1	2	3	3	4	5	-0.9342
Active User	4	4	3	2	2	1	

As shown in the table, there are 4 users with positive correlation these are users 1,2,6 & 8 and 2 users with negative correlation, these are users 3 & 9.

F. Finding the Nearest Neighbor

In finding the nearest neighbors, the users with high similarity value are considered the most similar users; these are users with high positive correlation. In this step, the K nearest neighbor is selected. Table 5 shows the most similar users.

Table v. Most similar users

	Item 3	Item 7	Item 4	Item 2	Item 6	Item 8
User 1	4	3	4	3	2	3
User 2	5	5	4	3	4	2
User 6	3	5	3	4	1	3
User 8	3	3	5	2	5	2

G. Computing the Prediction Rating

To compute the prediction rating, the weighted average of deviations from neighbors mean is added to active user's mean rating. Deviations are used to adjust for the user-associated biases. User biases occur when some users tend to give high or low ratings to all items. Once the most similar users are determined, the prediction rating is computed to generate the top N recommended items to the active user. The standard prediction metric [12] is used to compute the prediction rating as presented in (7):

$$P_{a,i} = r'_a + \frac{\sum_{u \in K}(r_{u,i} - r'_u).W_{a,u}}{\sum_{u \in K} W_{a,u}} \quad (7)$$

Where,

p (a,i) is the prediction rating of the target user a for item i

w (a,u) is the similarity of both users a and u

K is the number of neighbors of most similar users.

Table 6 shows the prediction rating of similar users to the target item to be recommended to the active user.

Table vi. Prediction Rating

	Item 3	Item 7	Item 4	Item 2	Item 6	Item 8	Item 1	Item 5	Item 9	Item 10
User 1	4	3	4	3	2	3	2	5	5	3
User 2	5	5	4	3	4	2	3	3	4	4
User 6	3	5	3	4	1	3	4	0	3	2
User 8	3	3	5	2	5	2	2	4	1	4
Active User	4	4	3	2	2	1				
Predicted Rating							2.73	3.07	3.37	3.23

Based on the result, the predicted rating for item 9 has the highest value with 3.37, followed by items 10 with 3.23, item 5 with 3.07, and item 1 with 2.73.

G. Top N Recommendation

This step arranges the target items from highest to lowest predicted rating, in this case, the result will be Item 9, Item 10, Item 5, and Item 1. Item 9 will most likely be recommended to the active user.

III. EXPERIMENTAL RESULTS

The Movie Lens 100K dataset was used in this experiment. The dataset includes 943 users, 1682 movies; it contains 100,000 rating records in a scale between 1 to 5 points, with 1 as the lowest and 5 as the highest. The User-Based Pearson Correlation [10] in (6) was used to compute the user similarity, it was tested using different numbers of neighbors, and the prediction metric [12] in (7) was used to compute the prediction rating. Lastly, in order to evaluate the enhanced algorithm, the MAE (Mean Absolute Error) [11] in (3) was used as a measure.

In this paper, the proposed algorithm (PA) was compared with the traditional collaborative filtering algorithm (TA) presented by [13] in order to test the accuray. Fig. 4 shows the MAE value of the two algorithms.

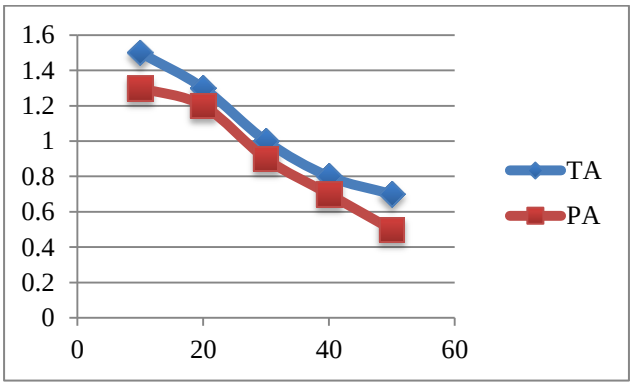

Fig. 4. MAE comparison of the traditional and proposed algorithm.

As presented in Fig. 4, the MAE values of the two algorithms presents a downward trend given of an extremely sparse rating data. The MAE of the proposed algorithm contains a small prediction error compared with the traditional collaborative filtering algorithm given of a number of neighbors between 10 and 50. It implies that the enhanced memory-based collaborative algorithm is more accurate compared with the traditional method.

IV. CONCLUSION

This paper analyzes the weakness of a traditional collaborative filtering algorithm through experimental evaluation. An enhanced memory-based collaborative filtering algorithm based on user similarity was proposed to resolve the data sparsity issues. Based on the experimental

result, the proposed algorithm has a small prediction error compared to the traditional algorithm and under data sparse condition, the proposed method has a better recommendation quality compared to the traditional method.

V. FUTURE WORK

Comparing the proposed algorithm with other enhanced algorithm is being considered in the future work.

ACKNOWLEDGMENT

The authors would like to thank Iloilo Science and Technology University for the Support.

REFERENCES

1. G. M. Dakhel and M. Mahdavi, "A new collaborative filtering algorithm using K-means clustering and neighbors' voting," 2011 11th International Conference on Hybrid Intelligent Systems (HIS), Melacca, 2011, pp. 179-184.
2. H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, "A new user similarity model to improve the accuracy of collaborative filtering," Knowledge-Based Systems, 56, 156–166.
3. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," In Proceedings of the 10th international conference on World Wide Web (WWW '01). ACM, New York, NY, USA, 285-295.
4. G. Tseng and W. Lee, "An enhanced memory-based collaborative filtering approach for context-aware recommendation," Proceedings of the World Congress on Engineering 2015 Vol I WCE 2, London, U.K.
5. R. Sharma, D. Gopalani, and Y. Meena, "Collaborative Filtering-Based Recommender System: Approaches and Research Challenges," 3rd International Conference on Computational Intelligence and Communication Technology (CICT), India.
6. Y. Wang, Y. Liu and X. Yu, "Collaborative Filtering with Aspect-Based Opinion Mining: A Tensor Factorization Approach," 2012 IEEE 12th International Conference on Data Mining, Brussels, 2012, pp. 1152-1157.
7. W. Ma, J. Shi, and R. Zhao, "Normalizing Item-Based Collaborative Filter Using Context-Aware Scaled Baseline Predictor," Mathematical Problems in Engineering. 2017. 1-9.
8. D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, "Recommender Systems: An Introduction," International Journal of Human–Computer Interaction, Cambridge University Press, 2012, (Vol. 28), pp. 72-73.
9. J. Herlocker, J. Konstan, and A. Borchers, "An algorithmic framework for performing collaborative filtering," Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley. California, United States, Berkeley. California, United States, pp. 230-237
10. G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," IEEE Transactions on Knowledge and Data Engineering, 2015, 17(6), 734–749.
11. M. Chen, "Performance Evaluation of Recommender Systems". International Journal of Performability Engineering, 2017, 13(8), 1246–1256.
12. B. Zhang, and B. Yuan, "Improved collaborative filtering recommendation algorithm of similarity measure," AIP Conference Proceedings, 2017, 1839