Collaborative Filtering Recommendation based on Rating Habits and Items' Attributes

Ghadah Alabduljabbar [1], Nouf Alrowais [2], Hafida Benhidour [3]

Dept. of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia.

[1] 439204033@student.ksu.edu.sa; galabduljabbar@kacst.edu.sa
[2] 439204064@student.ksu.edu.sa; nalrowais@kacst.edu.sa
[3] hbenhidour@ksu.edu.sa

***Abstract* - Collaborative filtering (CF) is one of the most widely used algorithms in recommendation systems. It uses the ratings of similar users to predict the ratings of target items. Due to the sparsity of the rating matrices, the number of co-rated items becomes small which makes the computation of the similarity between users inaccurate or even impossible, thus the quality of the recommendation is affected. This paper aims at solving the problem of data sparsity in collaborative filtering. A new similarity measure that uses the users' preferences for items' attributes and the users' personal rating habits is proposed in order to improve the predicted ratings. The proposed similarity gives more weight to the users' rating habit when the number of co-rated items is high, otherwise the users' preferences for items' attributes are given a higher weight. The experimental results show that the proposed similarity has a good prediction accuracy of the ratings.**

Keywords - collaborative filtering, recommendation system, similarity measure, rating prediction, data sparsity, personal rating habit, preference for item attributes.

I. INTRODUCTION

With the rapid development of Internet and the fast growth rate of data resources, satisfying people's needs in finding an information has become a real challenge. Recommender systems are effective tools for solving information overload problems by finding the most suitable items for users and transform the situation of "people looking for information" into "information looking for people". Users, thus, spend less time and energy. At present, various Internet platforms, such as Amazon, Netflix, and Facebook, use various types of recommendation algorithms to varying degrees.

Collaborative Filtering (CF) is one of the most widely used algorithm. The basic idea of CF algorithm is to calculate the similarities between users and a target user based on their ratings, then recommend an item based on the ratings of the similar users. To find the similar users of a target user, there are different traditional similarity measures that have been used, such as, Jaccard similarity (Jaccard) [1], Cosine similarity (COS) [2] and Pearson Correlation Coefficient (PCC) similarity measure [3].

When the user-item rating matrix is very sparse, the traditional similarity measures may not find other users who share the same interests as the target user. This is due to the fact that the number of co-rated items between any two users is small which makes the computation of the similarity between these users inaccurate or even impossible. Therefore, the recommender system will not be able to recommend items that are suitable. Improving the way the similarity measures are computed is thus crucial in improving the performance of the Collaborative Filtering.

Traditional similarity measures, usually, consider the set of users who have co-rated the same items as a target user. This approach, however, ignores the users' preferences for the items' attributes. Usually, users prefer items with certain characteristics rather than one or several specific items. In addition, everyone has a personal habit to express their preferences. Some users tend to give high rating values to items, others prefer low values. This habit in rating items affects the relationship between users, and traditional similarity measures ignore this factor. To overcome these limitations, this paper proposes a similarity measure model that considers users' preferences for items' attributes and takes into account the users' personal rating habits.

This paper is structured as follows: Section 2 reviews the most important Collaborative Filtering (CF) algorithms that are related to the proposed approach. Section 3 presents the problem statement. Section 4 describes the details of the proposed approach and the similarity measures that have been used. Section 5 presents the obtained results and, finally, Section 6 concludes the paper.

Problem Statement: Given an $m \times n$ sparse User-Item-Rating matrix, where m is the total number of users and n is the total number of items. The items are described by a set of attributes, but no ratings are provided by the users for these attributes. The aim is to improve the similarity measure by taking into consideration the users' rating habits and the users' preferences to certain attributes of the items in order to select the similar users to a certain target user. Since the attributes are not rated, an $m \times k$ User-Attribute-Rating matrix is built from the User-Item-Rating matrix, where k is the total number of attributes used to describe the n initial items.

II. RELATED WORK

With the growth and the rapid increase of the number of users and the number of items that are available online, the problem of data sparsity in Collaborative Filtering (CF) algorithms and the inaccuracy of the similarity measurement caused by it are becoming more and more prominent, which directly leads to a lower quality of recommendations. For this reason, researchers proposed different approaches to improve the recommendation accuracy.

Xuansen et al. [4] proposed a similarity measure model that considers users' preferences for items' attributes. Their model consists of two parts: one computes the co-rated items similarity measure model between users based on the User-Item rating matrix, the other computes the similarity based on the preferences for items' attributes based on the User-Attribute rating matrix. They used Jaccard similarity as a weighting factor between the similarity of users based on the ratings of co-rated items and the similarity for items' attributes since Jaccard similarity represents the proportion of co-rated items. Based on their experimental results, they found that their proposed model achieved improvement in the accuracy of recommendation results.

On the other hand, Hongtao et al. [5] proposed a new user similarity method, that takes into account the fact that every user has a different rating habit, some of them prefer to give high or low ratings comparing to others. Their method first calculates the rating difference between all users, then to calculate the final similarity they combine it with the Cosine similarity and Jaccard similarity. Thus, when users with similar personal rating habits have similar rating values, the final similarity between them will be higher than users who have different rating habits but have similar rating values for specific item. They used MovieLens (MovieLens Dataset, grouplens.org/datasets/ movielens/) dataset for their experiments and showed that the proposed method can handle the problem of data sparsity and improve the recommendation accuracy.

People usually are influenced by their surrounding family, friends, and colleagues when they take personal decisions. Based on this idea Reshma et al. [6] tried to alleviate the data sparsity and cold start problems by proposing a new approach that calculates the similarities between users based on their social networks data, such as location, favorites, liked products, social profile, if the users' rating information is not available. Nguyen et al [7] proposed also to overcome the data sparsity and cold start problems by extracting social networks community preferences of target users. On the other hand, Yuan et al. [8] introduced two types of social relationships, membership and friendship, with their effects on users' choices and interests. Their experiment shows the impact of the data sparsity on the recommendation accuracy is reduced when using social relationships to compute the similarity between users.

Clustering the users based on specific aspects is also one of the proposed ways to deal with the problem of data sparsity. Mark et al. [9] proposes to cluster the users according to their ratings of a set of items, then computed the prediction for each cluster independently. While Zahra et al. [10] proposes to cluster users based on their "personality characteristics", users who have similar personalities are placed in the same cluster using K-means algorithm. Researchers proved that clustering improves the prediction quality and increases the scalability of collaborative filtering algorithms.

Keunho et al. [3] suggested that the neighbors need to be selected based on the target item that the recommender system aim to predict its rating. They proposed to use the similarity between the co-rated items and the target item as a weight for each rating of the co-rated items. They also proposed to compute the similarity based on three approaches, Pearson correlation coefficient, Cosine similarity and Distance-based similarity. While Jiumei et al [11] propose to adjust the deviation of the traditional similarity measures with an impact factor when the data is sparse. The impact factor represents the ratio of co-rated items between two users, and in case there are no common items the value is zero.

III. PROPOSED APPROACH

In this section, the proposed approach for calculating the similarity measure is presented. The new similarity measure consists of two parts. The first part is the co-rated items similarity measure CJH (Cosine-Jaccard-Habit Measure) that combines the Cosine, Jaccard, and users' rating habits similarities to compute the similarity between the users using the User-Item-Rating matrix. The second part uses the User-Attribute-Rating matrix to compute the similarity between the users based on their preferences for the items' attributes. The next sections provide more details on each part of the proposed similarity measure.

A. Cosine-Jacquard-Habit (CJH) Similarity Measure

In order to compute CJH similarity measure, the User-Item-Rating matrix is used. Table I shows an example of this matrix.

TABLE I. USER-ITEM-RATING MATRIX

User-Item Rating Matrix		**Items**				
		I_1	…	I_i	…	I_n
Users	U_1	3	…	5	…	4
	…	…	…	…	…	…
	U_i	0	…	1	…	3
	…	…	…	…	…	…
	U_m	2	…	4	…	0

The rating given by a user u to an item i is represented by R_(u,i) and its value ranges between 1 and 5. If the user u did not rate the item i, the value of R_(u,i) is 0.

A1. Cosine Similarity

The cosine similarity (COS) [2] measures the cosine of the angle between the two vectors. Given two users u and v, each represented by an n-dimensional vector, the cosine similarity measure between user u and user v is defined as:

$$Sim(u,v)^{Cos} = \frac{\sum_{i\epsilon I} R_{u,i} R_{v,i}}{\sqrt{\sum_{i\epsilon I} {R_{u,i}}^2}\sqrt{\sum_{i\epsilon I} {R_{v,i}}^2}} \quad (1)$$

Where I is the set of items that the user u and user v have co- rated, R_(u,i) is the rating that user u gave to item i, and R_(v,i) is the rating that user v gave to item i.

A2. Jacquard Similarity

In Jaccard similarity [1], the similarity between users depends on the number of co-rated items and is calculated as following:

$$Sim(u,v)^{Jaccard} = \frac{|I_u \cap I_v|}{|I_u \cup I_v|} \quad (2)$$

Where Iu is the set of items that the user u has rated, and Iv is the set of items that the user v has rated.

A3. Users-Habit Similarity

The users' rating habits similarity [5] takes into consideration the users' tendency in ratings the items. The user rating habit similarity is calculated as follow:

$$Sim(u,v)^{H} = \frac{1}{\sum_{i=m}^{M}\left|\frac{\theta_u^{\#i}}{N_u} - \frac{\theta_v^{\#i}}{N_v}\right| + 1} \quad (3)$$

Where θ _u is a vector representing the frequency of each rating value i, where i∈{1,2,3,4,5}, for the user u, m represents the minimum possible value for the ratings (m=1), and M represents the maximum possible value for the ratings (M=5), Nu and N_v represent the total number of items that the user u and v have rated. For example, if the ratings of 10 items given by user u are (1,5,3,2,0,2,4,1,4,2), where the rating scale is 1 to 5 (m=1 and M=5), then the vector θ_u is (2,3,1,2,1), and Nu = 9 since one item is not rated. Also, the value 1 is added to the denominator to avoid division by 0. Whenever the difference between users in their rating habits is large the user habit similarity value will be small, and if the difference is small the similarity will be large.

A4. Combining the Similarities

The users' rating habit similarity is combined with the traditional similarities Cosine Eq. (1) and Jaccard Eq. (2), the formula for calculating the combined similarity between users is as follows:

$$Sim(u,v)^{\mathrm{CJH}} = Sim(u,v)^{cos} + Sim(u,v)^{jaccard} + Sim(u,v)^{H} \quad (4)$$

B. User- Attribute Similarity

Each item is described by a set of attributes. For example, if the items are movies, the genres, such as action or comedy, are attributes describing the movies, a movie can have different genres. Since the users did not provide ratings for the genres, the User-Attribute-Rating matrix needs first to be built in order to be able to compute the similarity between the users based on their rating for the attributes. In order to compute the rating R_(u,A_k) representing the rating of the attribute A_k by user u, the ratings of all items described by attribute A_k and rated by u are aggregated. If, for example, a user u has rated only two items i and j and assuming that item i is described by the attribute A_1 and A_2 and item j is described only by the attribute A_1, the obtained ratings for user u are R_(u,A_1)=R_(u,i)+R_(u,j) and R_(u,A_2)=R_(u,i). An example of a User-Attribute-Rating matrix is shown in Table II with m users and k attributes.

TABLE II. USER-ATTRIBUTE-RATING MATRIX

User-Attribute-Rating Matrix		Items' attributes				
		A_1	…	A_j	…	A_k
Users	U_1	5	…	71	…	65
	…	…	…	…	…	…
	U_i	0	…	82	…	34
	…	…	…	…	…	…
	U_m	0	…	74	…	0

B1. Pearson Correlation Coefficient

To compute the similarity between the users based on the User-Attribute-Rating matrix, the Pearson correlation coefficient (PCC) [12] is used as follows:

$$Sim(u,v)^{PCC} = \frac{\sum_{i=1}^{k}(R_{u,A_i} - \bar{R}_u)(R_{v,A_i} - \bar{R}_v)}{\sqrt{\sum_{i=1}^{k}(R_{u,A_i} - \bar{R}_u)^2}\sqrt{\sum_{i=1}^{k}(R_{v,A_i} - \bar{R}_u)^2}} \quad (5)$$

Where R ̅u is the mean value of the ratings of the user u given to all the attributes, and R ̅v is the mean value of the ratings of the user v given to the attributes of the items.

C. Habit- Item Attribute Similarity

Finally, the formula for calculating the user similarity is as follows:

$$Sim(u,v)^{\mathrm{CJHA}} = Sim(u,v)^{jaccard}\, Sim(u,v)^{\mathrm{CJH}} +$$

$$(1 - \ Sim(u,v)^{jaccard}\ )Sim(u,v)^{PCC} \qquad (6)$$

Jaccard similarity Eq. (2) is used as a weighting factor since Jacquard similarity measures the proportion of co-rated items, the higher the ratio means the higher weight for Sim(u,v)^CJH, and if the ratio is small more weight is given to the ratings of the attributes.

D. Rating prediction

The rating prediction for item i by user u is calculated according to the following equation:

$$P_{u,i} = \bar{r}_u + \frac{\sum_{v\epsilon N} Sim(u,v)^{CJHA}(r_{v,i} - \bar{r}_v)}{\sum_{v\epsilon N} |Sim(u,v)^{CJHA}|} \qquad (7)$$

Where, u is the target user, and i is the target item that we want to predict its rating value, and N is the set of the K most similar neighbors to the user u selected based on Sim(u,v)^CJHA.

In summary, the proposed CF recommendation algorithm CJHA is as follows:

Algorithm 1: CJHA Algorithm

Input:

User-Item-rating matrix: a dataset containing the ratings given by the users to certain items. The items need to have attributes.

K: the number of nearest neighbors.

Output: The MAE and $RMSE$ evaluation scores for the predicted ratings

1 Build the User-Attribute-Rating matrix.
2 Split the data set into a training set (90%) and a test set (10%).
3 **for** each user u
4 Calculate the Cosine similarity between u and the others based on Eq.(1)
5 Calculate the Jaccard similarity between u and the others based on Eq.(2)
6 Calculate the User-Habit similarity between u and the others based on Eq.(3)
7 Calculate the CJH similarity between u and the others based on Eq.(4)
8 Calculate the Pearson similarity (from the User-Attribute-Rating matrix) between u and the others based on Eq.(5)
9 Calculate CJHA similarity between u and the other users based on Eq.(6)
10 **end**
11 Find the K similar neighbors to each user.
12 Calculate the predicted scores for all users in the test set based on Eq.(7)
13 Calculate the MAE and $RMSE$ evaluation scores based on Eq.(8) and (9)
14 **return** MAE and $RMSE$ scores

IV. EXPERIMENT

In this section, we present the conducted experiments on publicly available dataset to assess the reliability of the proposed method to alleviate data sparsity problem.

A. Dataset

The proposed algorithm has been tested on MovieLens 100k datasets provided by the United States Group Lens project team. It is currently the most commonly used and authoritative test data in the quality evaluation of recommended algorithms. This dataset includes two data files: u.data includes ratings for 943 users on 1682 movies and u.item includes the attribute information of the movies like movie genre, and rating information with a scale ranging from 1 to 5. The sparseness of this dataset is 93.69%, which means that only 6.31% of movies are scored by users.

B. Evaluation Metrics

The two most popular evaluation metrics, the mean absolute error (MAE) and the root mean square error (RMSE), are used to evaluate the prediction accuracy of the proposed similarity measure.

B1. Mean Absolute Error (MAE)

The Mean Absolute Error (MAE) [13] computes over all users the average of the difference between user's predicted rating and user's actual ratings by using the following equation:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{l} |p_i - r_i|}{l} \qquad (8)$$

Where pi is the i^th predicted rating, ri is its corresponding actual rating, and l represents the total number of ratings. The smaller MAE, the more accurate recommendation results.

B2. Root Mean Square Error (RMSE)

The Root Mean Square error (RMSE) [14] reflects also the deviation of the predicted ratings from the real value but also penalizes the large errors since the error term is squared. The smaller the RMSE, the higher the accuracy is. It is calculated by the following equation:

$$\mathrm{RMSE} = \sqrt{\frac{1}{l}\sum_{i=1}^{l}(p_i - r_i)^2} \qquad (9)$$

Where pi is the i^th predicted rating, ri is its corresponding actual rating, and l represents the total number of ratings.

V. RESULTS AND DISCUSSIONS

The proposed approach has been implemented in python programing language, and the following packages (pandas, numpy and sklearn) have been used. The number of the most similar neighbors (K) used when computing the predicted ratings ranges from 10 to 100. In addition, the obtained MAE and RMSE for the proposed approach have been compared to the results of three other approaches (the algorithm based on users' preferences for items' attributes that was proposed by Xuansen et al. [4] (we named it ITEM_ATT_SIM), the User-Rating-Habit-based algorithm that was proposed by Hongtao et al. [5] (we named it H_SIM) and the Cosine and Jaccard similarity (we name it COS_JACC_SIM and is computed by summing the Cosine and Jaccard similarities).

First, the dataset is split into a training set and a test set with 90% for the training and 10% for the test. After that, the previous algorithms are applied on the training set, then evaluated using the test set. The experiment results are shown in Figure 1, Figure 2, and Figure 3.

Figure 1 and Figure 2 show the comparison of the different approaches based on the MAE and RMSE respectively. Different values for the number of the nearest neighbors are tested.

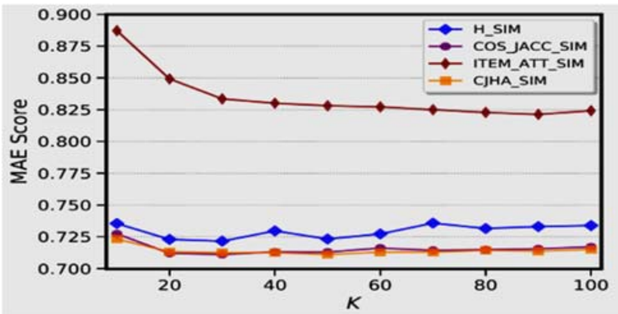

Figure 1 – MAE-based comparison of different similarity models: CJHA _SIM (proposed CJHA similarity), ITEM_ATT_SIM [4], H_SIM [5] and COS_JACC_SIM (Cosine-Jaccard similarity)

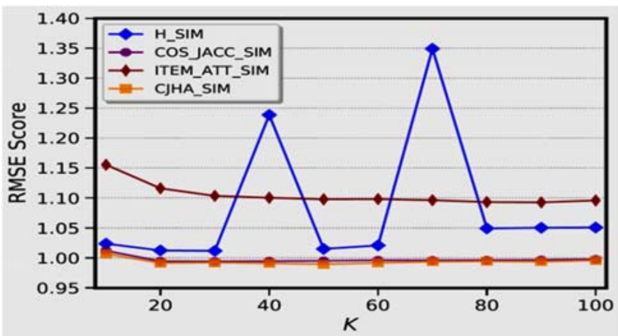

Figure 2 – RMSE-based comparison of different similarity models: CJHA_SIM (proposed CJHA similarity), ITEM_ATT_SIM [4], H_SIM [5] and COS_JACC_SIM (Cosine-Jaccard similarity)

H_SIM similarity didn't provide any improvement comparing to COS_JACC_SIM, in contrast the error increased slightly by increasing the number of similar neighbors. Thus, using users' rating habits alone to deal with data sparseness problems is not a solution, and can make the prediction worse than using the traditional similarity measure alone. ITEM_ATT_SIM similarity produce the highest error rate in term of MAE and RMSE. This high error is due to the loss of information when combining the ratings of items based on their attributes.

The proposed similarity model (CJHA_SIM) combines the previous models to overcome their disadvantages. In case the dataset doesn't have enough ratings, the similarity is calculated based on the ratings of items' attributes since more weight is given to them, and if the dataset is not sparse, the model calculates the similarity by giving more weight to the Cosine-Jaccard-Habit similarity. The Jaccard similarity is used as a weighting coefficient since its value increases when the number of co-rated items is high, otherwise it decreases. From both Figures 1 and 2, the proposed similarity (CJHA_SIM) has the best prediction accuracy compared to other models, even though the traditional similarity (COS_JACC_SIM) produces an error that is close to the one obtained from the proposed approach.

For the nearest neighbors numbers (K), we can see that as the number of nearest neighbors increases, the error scores of approximately all models except H_SIM decreased. For H_SIM similarity, increasing the number of the similar neighbors results in increasing the error rate. Since RMSE amplifies large errors, H_SIM produces a very large RMSE score (between 1.2 and 1.4), even higher than ITEM_ATT_SIM model, when K is 40 and 70. For the proposed approach, the best result is obtained when K is equal to 50.

We also compared the proposed approach (CJHA_SIM) with the approach that was proposed by Junmei et al. [15] (Junmei_SIM), after splitting the dataset into a training set and a test set with proportions equal to 80% for the training and 20% for the test set. The comparison was based on MAE score in the MovieLens dataset as shown in Figure 3.

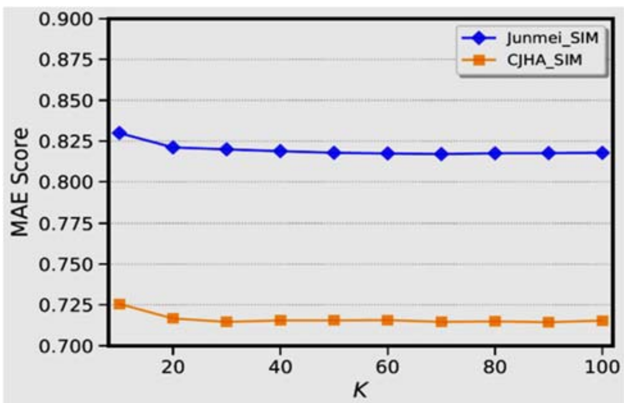

Figure 3 - MAE comparison between CJHA similarity and the similarity proposed in [15]

Comparing CJHA_SIM with the model in [15] (Junmei_SIM) shows that using users' rating habits with users' preferences for the items' attributes yields a significantly better prediction performance than the other model that uses the users' rating similarity and the proportion of the co-rated items to calculate the overall similarity between users. However, Junmei et al. [15] mentioned that their model works well when the dataset is extremely sparse (the rating sparsity is more than 98%). Movielens 100K dataset has a sparsity of 93.69%.

The results show that the similarity model that is proposed in this paper has a small prediction rating error in term of RMSE (0.988 < RMSE < 1.008) and MAE (0.711< MAE < 0.725) and is the best among the four mentioned similarity.

VI. CONCLUSION

Many approaches for Collaborative Filtering have been proposed in recent years to improve the accuracy of the predicted ratings. When the user-item rating matrix is very sparse, the number of co-rated items between any two users becomes small, making thus the computation of the similarity between these users inaccurate or even impossible. This paper proposes a similarity measure that is suitable for sparse datasets. It takes into account the users' preferences for items' attributes and the users' personal rating habits to predict users' ratings for unrated items. The proposed similarity gives more weight to the users' rating habit when the number of co-rated items is high, otherwise the users' preferences for items' attributes are given a higher weight. From the experiment, the proposed similarity has shown to have a better accuracy in predicting users ratings than other similarity measures.

As future work, we will focus on improving the prediction accuracy by training the model using a larger size dataset such as 25M MovieLens. Another possible research axis consists in adding extra information on the users to the proposed similarity measure in order to improve its accuracy when the dataset is sparse. This information can be extracted for example from a questionnaire or from a social network, such as the users' profiles, their locations and their liked pages.

REFERENCES

[1] Koutrika, Georgia, Benjamin Bercovitz, and Hector Garcia-Molina. "FlexRecs: expressing and combining flexible recommendations." Proceedings of the 2009 ACM SIGMOD International Conference on Management of data. ACM, 2009.

[2] Tuzhilin, Alexander. "Towards the next generation of recommender systems." Proceedings of the 1st International Conference on E-Business Intelligence (ICEBI2010). Atlantis Press, 2010.

[3] Choi, Keunho, and Yongmoo Suh. "A new similarity function for selecting neighbors for each target item in collaborative filtering." Knowledge-Based Systems 37 (2013): 146-153.

[4] He, Xuansen, and Xu Jin. "Collaborative filtering recommendation algorithm considering users' preferences for item attributes." 2019 International Conference on Big Data and Computational Intelligence (ICBDCI). IEEE, 2019.

[5] Liu, Hongtao, et al. "Effective Similarity Measures of Collaborative Filtering Recommendations Based on User Ratings Habits." 2018 14th International Conference on Semantics, Knowledge and Grids (SKG). IEEE, 2018.

[6] Reshma, R., G. Ambikesh, and P. Santhi Thilagam. "Alleviating data sparsity and cold start in recommender systems using social behaviour." 2016 International Conference on Recent Trends in Information Technology (ICRTIT). IEEE, 2016.

[7] Nguyen, Van-Doan, Songsak Sriboonchitta, and Van-Nam Huynh. "Using community preference for overcoming sparsity and cold-start problems in collaborative filtering system offering soft ratings." Electronic Commerce Research and Applications 26 (2017): 101-108.

[8] Yuan, Quan, Li Chen, and Shiwan Zhao. "Factorization vs. regularization: fusing heterogeneous social relationships in top-n recommendation." Proceedings of the fifth ACM conference on Recommender systems. 2011.

[9] O'Connor, Mark, and Jon Herlocker. "Clustering items for collaborative filtering." Proceedings of the ACM SIGIR workshop on recommender systems. Vol. 128. UC Berkeley, 1999.

[10] Hafshejani, Zahra Yusefi, Marjan Kaedi, and Afsaneh Fatemi. "Improving sparsity and new user problems in collaborative filtering by clustering the personality factors." Electronic Commerce Research 18.4 (2018): 813-836.

[11] Mao, Jiumei, et al. "An improved similarity measure method in collaborative filtering recommendation algorithm." 2013 International Conference on Cloud Computing and Big Data. IEEE, 2013.

[12] Schafer, J. Ben, et al. "Collaborative filtering recommender systems." The adaptive web. Springer, Berlin, Heidelberg, 2007. 291-324.

[13] Goldberg, Ken, et al. "Eigentaste: A constant time collaborative filtering algorithm." information retrieval 4.2 (2001): 133-151.

[14] Cotter, Paul, and Barry Smyth. "Ptv: Intelligent personalised tv guides." AAAI/IAAI. 2000.

[15] Feng, Junmei, et al. "An improved collaborative filtering method based on similarity." PloS one 13.9 (2018): e0204003.

.