Collaborative Filtering Recommendation Algorithm Based on User Characteristics and User Interests

Shangsong Li, Xuesong Li
School of computer science, Inner Mongolia University, 010021, China
shangsongli1994@163.com

Abstract. Among many e-commerce platforms, collaborative filtering recommendation algorithm is currently the most widely used recommendation technology. In order to alleviate the deficiencies of the traditional user-based collaborative filtering algorithm in cold start and recommendation accuracy, this paper proposes a collaborative filtering recommendation algorithm based on user characteristics and user interests. The similarity of the algorithm in this paper is composed of user score similarity, user attribute feature similarity and user interest similarity, in which user registration information is used to extract attribute features to calculate user feature similarity; use the number of user evaluations of project attributes to measure users’ interest in different project attributes, use the similarity calculation formula to calculate the interest similarity value between users. The user attribute feature similarity and user interest similarity are combined with the user rating similarity to obtain the final similarity for recommendation. Finally, a simulation experiment is performed on the MovieLens movie data set. Through the experimental results, it can be seen that the improved collaborative filtering algorithm based on user characteristics and user interests not only solves the cold start problem, but also improves the recommendation accuracy.

1. Introduction
In recent years, with the rapid development of information technology, the Internet produces a lot of data every day. Taking the Douyin App, which is well known in recent years, as an example, as long as we register an account, we have the opportunity to become a video producer and publisher, recording our own lives and also sharing our lives with others; another example is microblog, each of us can post our blog on the Internet to express our personal feelings and share with others. We can also spread our knowledge and communicate with others through blog. In the face of so much data, researchers have come up with a number of ways, for example, information retrieval technology that requires manual input of key information saves search time to a certain extent, but if the retrieval content cannot be given accurately, the searched product often cannot fully meet the user's interests and hobbies, moreover, with the exponentially increasing amount of information every year, the amount of data retrieved has also increased rapidly, so this technology is facing severe challenges[1]. The emergence of recommendation technology has solved this problem well. It uses some behaviors of users and some mathematical algorithms to predict what users may be interested in, and actively recommends to users[2], which is equivalent to intelligent recommendation. The application field of the recommendation algorithm is very wide. It can be applied to book websites, music websites, video websites, map websites, and news portal[3,4].

There are many types of recommendation algorithms, such as content-based recommendations, collaborative filtering (referred to as CF)-based recommendation algorithms and association rules-
based recommendations[5]. Among the many recommendation algorithm technologies, one of the earliest and most successful technologies is the CF[6].

The CF includes model-based algorithm and memory-based algorithm[7]. Memory-based CF technology is divided into two categories, one based on users and the other based on items[8]. CF usually uses the nearest neighbor technology, count the distance between users through the acquired user's historical preference data, and then forecasts the intent user's preference for characteristic products based on the weighted value of the goal user's neighbor users' product evaluation degree, and finally recommend to the target user based on this preference. The strongest point is that it doesn’t pose any restrictions on the recommended objects and it can process unstructured complex objects such as movies and music. However, collaborative filtering recommendations still have some shortcomings, such as sparse problems, scalability problems, new user problems, poor quality of recommendations at the beginning of the system, cold start[9].

2. Related Work
Various improved algorithms are emerging one after another. Literature[10] investigates how to leverage the heterogeneous information in a knowledge base to improve the quality of recommender systems. Literature[11] proposed a new user similarity model to improve the recommendation performance when only few ratings are available to calculate the similarities for each user, which improves the recommendation performance. Cao[12] and others uses knowledge graphs to analyze the reasons for user preferences, understand the user's points of interest, and recommend items they may like to the user based on the points of interest. Ma[13] proposed a consider the situations involving ratings and trust. Literature[14] proposed a weight-based time similarity calculation method, which effectively reflects the user ‘s interest, improved the accuracy of recommendations. Literature[15] proposed a movie recommendation model based on word vector features, improves the recommendation effect, but the input of a single text is Improving the accuracy of recommendations may not be very effective. Literature[16] proposed an ensemble based k-NN collaborative filtering, this work fuses the merits of the user based k-NN algorithm and item-based algorithm, and give better recommendation quality. Literature[17] proposed a similarity measure model considering users’ preferences for item attributes, which effectively improves the performance of the recommended algorithm. Literature[18] proposed a method of filling unrated data on the user-item rating matrix by using linear regression model, alleviate the data sparsity. Literature[19] introduced an enhanced recommendation algorithm based on modified user-based collaborative filtering, improve the recommendation quality.

Based on the above research, this paper presents an improved collaborative filtering recommendation algorithm. It is fully considers the influence of the user's attributes characteristics on the cold start in terms of the recommended quality. By computing the resemblance of the user's characteristics, the problem of poor recommendation quality at the beginning of the system is alleviated; in terms of prediction accuracy, the user's personalized interest information is fully utilized to establish an interest matrix to calculate the interest similarity value. The user attribute feature similarity is merged with user’s interest resemblance and user’s rating resemblance to obtain the final resemblance value, and finally the similarity value is scored and predicted on the user-item rating matrix, and filtered using the Top-N algorithm[20], use the average absolute error value to measure whether the algorithm has been improved and become better, through the experimental results, we can see that the algorithm in this paper has better advantages than the traditional CF algorithm.

3. Steps of traditional collaborative filtering algorithm

3.1. Establish a scoring matrix
The data involved in the User-based CF algorithm is the user ‘s historical rating data for the item, and the rating matrix N is used to indicate the rating data, m indicates the number of users and n represents the count of items. N_{ij} is the rating value of the user U_i for the item I_j. The rating value is generally
represented by an integer of 1 to 5. The score is positively correlated with the user's likes. The rating matrix is shown below.

| User | Item | I_1 | ... | I_j | ... | I_n |
|------|------|-----|-----|-----|-----|-----|
| U_1  | N_11 | ... | N_1j | ... | N_1n |
| ...  | ...  | ... | ...  | ...  | ...  |
| U_i  | N_i1 | ... | N_ij | ... | N_in |
| ...  | ...  | ... | ...  | ...  | ...  |
| U_m  | N_m1 | ... | N_mj | ... | N_mn |

### 3.2. Select nearest neighbor

This article takes the user-based CF as an example to illustrate the process of generating the nearest neighbor set. First, using the following formula, count the resemblance between different users and the intent user a. From this, the nearest neighbor set $K_a = \{a_1, a_2, ..., a_k\}$ of the target user a is found, which is composed of the K most similar neighbors of a. This article gives three common methods of calculating similarity.

**Cosine similarity:**

The score vector of user a in the n-dimensional object space is denoted by $i$. The score vector of user b in the n-dimensional object space is denoted by $j$, the calculation is shown in (1):

$$Sim(a, b) = \cos(i, j) = \frac{i \cdot j}{\|i\| \|j\|}.$$  

**Modified cosine similarity:**

For the user rating scale, the cosine similarity is not considered, explain in detail, for example, if it is [1-5] in the scoring area, for user a, a rating of 3 or higher is what you like, and for user b, a rating of 4 or higher is your favorite. The modified cosine similarity measurement method can balance the difference of users' ratings, mainly by the user's rating minus the user's average score to balance, which improves the above problem. The modified cosine similarity is shown in (2):

$$Sim(a, b) = \frac{\sum_{i \in Nab} (N_{ai} - \overline{Na}) (N_{bi} - \overline{Nb})}{\sqrt{\sum_{i \in Na} (N_{ai} - \overline{Na})^2 \times \sum_{i \in Nb} (N_{bi} - \overline{Nb})^2}},$$  

In the modified cosine similarity formula $N_{ab}$ represents the common rating set of user a and user b, the rating set of a is represented by $N_a$, the rating set of user b is represented by $N_b$ and $N_{a,i}$ represents the rating of a on item i. $N_{b,i}$ represents the rating value of b for i and the average value of all ratings of user a is represented by $\overline{Na}$, similarly user b’s ratings average is represented by $\overline{Nb}$.

**Pearson related similarities:**

The strength of the correlation between the two variables can be described by the Pearson correlation coefficient. The larger its absolute value, the greater the correlation. The mathematical definition of Pearson's related similarity is shown in (3):

$$Sim(a, b) = \frac{\sum_{i \in Nab} (N_{ai} - \overline{Na}) (N_{bi} - \overline{Nb})}{\sqrt{\sum_{i \in Nab} (N_{ai} - \overline{Na})^2 \times \sum_{i \in Nab} (N_{bi} - \overline{Nb})^2}},$$  

3
Among them, $N_{a, b}$ stand for the set of items that users $a$ and $b$ have rated together; $N_{a, i}$ and $N_{b, i}$ represent the ratings of $a$ and $b$ for item $i$, and $\overline{Na}$ and $\overline{Nb}$ represent the average of all ratings of user $a$ and user $b$. This article only selects the Pearson correlation similarity with relatively superior performance under the same conditions for explanation.

3.3. Predict scores and generate recommendations

After obtaining the $K$ nearest neighbor set, we can predict the unrated items of the goal user according to the data of similar users in the set, and sort the predicted scores from high to low to generate Top-N recommendations to the target user for selection. The calculation is as shown in (4), the set $K_a$ is the neighbor set of the target user. $N_{b,i}$ represents user $b$'s rating of item $i$, $\overline{Na}$ and $\overline{Nb}$ represent the average of all ratings of $a$ and $b$, respectively. $P_{a,i}$ is user $a$'s predicted rating value of unrated items.

$$P_{a,i} = \overline{Na} + \frac{\sum_{b \in K_a} Sim(a,b) \times (N_{b,i} - \overline{Nb})}{\sum_{b \in K_a} Sim(a,b)},$$

(4)

4. Algorithm improvements

4.1. User feature similarity calculation

Generally, when a user starts to register an account on a website, some basic information needs to be filled in[21]. The information to be filled in on different websites may not be exactly the same, but most of them will include an account name, age, gender, occupation, and postcode. These seemingly unimportant basic information can actually reflect users' interests and hobbies. Users with the same characteristics often have similarities in interests and hobbies and become similar users. When new users are coming and there is no historical data, user characteristics are particularly important, so introducing the user's basic characteristics into the calculation of similarity can well avoid the problem of poor recommendation results brought by the user's cold start. This article uses the user's gender, age, occupation, zip code to calculate the user's feature similarity.

Age similarity:

As users of different ages, their favorite items will be different. As far as film and television works are concerned, children prefer cartoons, young people prefer idols, and old people tend to prefer time movies. Therefore, age similarity is a key factor of user feature similarity. We set the age of user $a$ to $A_a$ and the age of user $b$ to $A_b$. The following (5) is used to calculate the similarity of age attributes:

$$age(a,b) = 1 - \frac{|A_a - A_b|}{100},$$

(5)

Gender similarity:

The user's gender is different, there will be a big difference in the preferences of the project. Since the gender of a person is only a man and a woman. This article uses the following (6) to count the gender resemblance of user $a$ and user $b$:

$$Sex(a,b) = \begin{cases} 0, & a \text{ and } b \text{ have different genders} \\ 1, & a \text{ and } b \text{ have the same gender} \end{cases},$$

(6)

label similarity:

This article only considers the effect of occupational labels and postal codes on the similarity of user characteristics. Different occupations often have different people's preferences. At the same time, people's regions are different, and they are affected by different customs and habits. Therefore, their
interest preferences may also be different, and the postal code can accurately reflect people's geographical location. We use the following (7) to compute the label resemblance of user a and user b:

\[ \tag{7} \]

This paper adds the above three feature similarities to the weight coefficients for addition, where \( L_1, L_2, L_3 \in [0-1] \), the sum of \( L_1, L_2, L_3 \) is 1. Formula (8) is the user attribute similarity \( \text{Sim}_f(a, b) \) of user a and user b:

\[ \text{Sim}_f(a, b) = L_1 \text{age}(a, b) + L_2 \text{sex}(a, b) + L_3 \text{tag}(a, b), \] (8)

4.2. User interest similarity calculation

Human interest is composed of subjective expression and objective description, which are explicit interest and implicit interest. Explicit interest usually obtains the user's interest in the form of conversation or questionnaire. For example, after registering on Sina Weibo, it will recommend many bloggers to let users choose the bloggers they need to pay attention to. Niuke.com is a famous professional platform for programmers to learn and grow, will also allow new users to choose their favorite language and technology when they just register. However, because some users may not have enough patience, give up or fill in at will, it may lead to inaccurate final results. The user's hidden interest is mainly reflected in the user's behavior preference data, mostly user back-end data. The back-end data is mostly user behavior information, such as user click information on commodities, evaluation data on movies. In this way, the recommendation system can calculate the recommendation results from all the user's various behavior record data, which can greatly reduce the user's time, obtain and calculate the user's potential interests and hobbies in the background. Therefore, this paper uses implicit interest to compute the user's interest resemblance. This article defines the user's level of interest by counting the sum of the number of item attributes evaluated by the user. For example, a person has seen many movies (a movie contains more than one attribute). If you want to count the movies of the love attribute that the person has seen, then you can measure the person's preference for love movies from the number of times the movie contains love attributes. The number of visits is positively related to the degree of interest. Therefore, an interest matrix is established to express the user's interest in each item attribute.

\[ W_{m \times z} = \begin{pmatrix}
  w_{11} & \ldots & w_{1j} & \ldots & w_{1z} \\
  \vdots & \ddots & \vdots & \ddots & \vdots \\
  w_{m1} & \ldots & w_{mj} & \ldots & w_{mz}
\end{pmatrix} \]

In the matrix, the total count of users is represented by \( m \), and the total count of attributes is represented by \( z \), and the total number of times user i evaluates item attributes j is represented by \( w_{ij} \). Based on the user preference matrix, we can compute the similarity between users’ interest in item attributes. The calculation is revealed in (9):

\[ \text{Sim}_i(a, b) = \frac{\sum_{z \in Z_{ab}} (w_{a,z} - \bar{w}_a) (w_{b,z} - \bar{w}_b)}{\sqrt{\sum_{z \in Z_{ab}} (w_{a,z} - \bar{w}_a)^2} \times \sqrt{\sum_{z \in Z_{ab}} (w_{b,z} - \bar{w}_b)^2}}, \] (9)

In the formula, the total number of times the item evaluated by user a includes attribute \( z \) is represented by \( w_{a,z} \), and the total count of times the item evaluated by user b contains attribute \( z \) is
represented by $w_{b,z}$. The average value of the count of times the user $a$ evaluates all project attributes is denoted by $\overline{W}_a$, and the average value of the number of times user $b$ evaluates all project attributes is denoted by $\overline{W}_b$, and $Z_{ab}$ represents the collection of all item attributes evaluated by user $a$ and user $b$.

4.3. Similarity Fusion

We use (3) to calculate the user's rating similarity to get $\text{Sim}_r(a, b)$, the feature similarity is expressed by $\text{sim}_f(a, b)$ of (8), and the similarity of user interesting for project attributes is represented by $\text{sim}_i(a, b)$ of (9).

The three similarities are fused to get the final similarity $\text{sim}(a, b)$. The three weight parameters $d, e, f$ introduced here, where $d, e, f \in [0,1]$, and $d + e + f = 1$. As shown in (10):

$$\text{Sim}(a, b) = d\text{Sim}_r(a, b) + e\text{Sim}_f(a, b) + f\text{Sim}_i(a, b),$$

(10)

4.4. Improved algorithm flow

Input user rating information, item rating matrix, item attribute matrix, number of neighbors $k$, user feature information, and output target user's predicted rating. Here are the brief steps of the algorithm:

1. Convert the user rating information into a user rating matrix, and calculate the similarity between users using (3), obtain $\text{Sim}_r$;
2. Use user characteristic information to obtain various characteristics of users, and calculate the age similarity, gender similarity, label similarity between users;
3. Substitute the obtained similarity of various features into (8) to obtain the overall similarity $\text{Sim}_f$ of user features;
4. Scan the item attribute matrix, and use Equation (9) to count the interest resemblance value between users;
5. The results obtained in step 1 and the results obtained in steps 3 and 4 are calculated using (10) to obtain the end user similarity $\text{Sim}(a, b)$;
6. Using (4), predict the intent user score based on the set of $k$ neighbor users on the rating matrix.

5. Experimental results and analysis

5.1. Data Sources

In order to confirm the usefulness of the improved algorithm, this experiment uses the famous MovieLens data set for experiments, which provides a user message, a movie message and a score message. The user information table includes the user's gender, age, occupation, mailbox, etc. The movie message includes the movie release time and movie type. The rating information table includes 943 users, 1,682 movies, and 100,000 ratings. The rating range is [1-5] points. Each user has rated at least 20 movies. The score is positively correlated with the user's preference. The degree of data sparsity of this data set can be calculated according to the data sparsity formula, as shown in (11):

$$\text{spa} = (1 - \frac{\text{num}}{m \times n}) \times 100\%,$$

(11)

In the formula, $\text{spa}$ is the degree of sparsity of the data set, num represents how many ratings there are, $m$ represents how many users there are, and $n$ represents the number of items, use (11) to calculate the $\text{spa} = (1 - 100,000 / (943 \times 1,682)) = 93.69\%$. Here we separate the data set into a training set and a test set, and guarantee to be randomly assigned, the rate is 4:1. The training set is used for algorithm experiment and prediction estimation, and the test set is used for comparing the prediction estimation results.

5.2. Evaluation index
The average absolute error MAE is used as the criterion for judging the quality of recommendation results. This method of measuring recommendation accuracy is relatively easy to understand. Its essence is to count the average deviation between the forecast value and the true value.

The smaller the value of MAE, the higher the recommended quality, the corresponding calculation formula is shown in (12), where $p_i$ is the user's real rating value; $q_i$ is the rating value forecast by the algorithm; and the number of items predicted to score is $N$.

$$
MAE = \frac{\sum_{i=1}^{N} |p_i - q_i|}{N},
$$

(12)

5.3. Results and analysis

5.3.1. Experiment 1. Determine the number of nearest neighbors for tuning

In order to select the suitable number of nearest neighbors to achieve better recommendation results when adjusting parameters, this experiment analyzes the MAE values of different numbers of neighbors. First select the traditional CF recommendation algorithm, set the value range of the nearest neighbor to [5-50], the increment is 5, and then calculate MAE, the experimental results are as follows:

![Fig.1 The value of K on MAE](image)

Fig.1 shows, when the nearest neighbor user is 40, MAE tends to be stable, and the effect is better at this time, so the nearest neighbor user is selected as 40 in the subsequent parameter adjustment.

5.3.2. Experiment 2. Determination of user characteristic parameters

According to (8), $L_1$, $L_2$ and $L_3$ represent the weight of gender, age, label similarity in user attributes, respectively, and $L_1, L_2, L_3 \in [0-1]$, $L_1 + L_2 + L_3 = 1$, set the increment to 0.1, the abscissa is the value of the weight $L_1$, and then calculate the MAE value corresponding to different weights, the experimental results are as follows:
8

We can see from the chart above, when $L1 = 0.5$, $L2 = 0.4$, $L3 = 0.1$, the average absolute error (MAE) obtained is smaller than other values.

5.3.3. Experiment 3. Determination of $d$, $e$, $f$ parameters
According to (10), $d$, $e$, and $f$ represent the weight of user rating resemblance, user attribute resemblance, and user interest similarity, respectively, and $d$, $e$, $f \in [0-1]$, $d + e + f = 1$, set the increment to 0.1, the abscissa is the value of the weight $d$, and then calculate the MAE value corresponding to different weights, the experimental results are as follows:

It can be seen from Fig.3 that when $d = 0.1$, $e = 0.1$, $f = 0.8$, the average absolute error (MAE) obtained is smaller than other values.

5.3.4. Experiment 4. Comparison experiment of different recommendation algorithms
In order to better reflect the effect of the algorithm put forward in this article, the improved algorithm in this paper is compared with the traditional CF algorithm (Pearson Similarity) and the improved algorithms in literature[17], literature[21]. The values of the nearest neighbors are all in the range of [10-50], and the increment is 10. Compare the MAE values of the respective algorithms under different numbers of neighbors. The experimental results obtained are shown as follows:
In Fig.4, we can see that when the number of nearest neighbors is small, the algorithm results in this paper are good. With the increase in the number of nearest neighbors, although the MAE value obtained by the algorithm in this paper increases, at 35 points, same as the literature [17], when the nearest neighbor is greater than 35, it is slightly inferior to the literature [17], but still has a smaller MAE value than the traditional algorithm and the algorithm of the literature [21]. Therefore, the improved algorithm in this paper is more suitable for the case where the number of nearest neighbors is small.

6. Conclusions and Future Works
This paper aims at the shortcomings of traditional CF recommendation in such aspects as cold start and recommendation accuracy, a CF algorithm based on user characteristics and user interest is presented. The addition of user features effectively avoids the trouble caused by cold start, simultaneously, the introduction of user interest models enhances personalized recommendations and further improves the accuracy of similarity calculation; finally, the user rating similarity, user feature similarity and user interest similarity are used to find suitable fitting parameters through separate experiments. The final similarity value is obtained, and the reliability of the method is verified by experiments. Future work will study the impact of time changes on user interest.

Acknowledgments
Thanks to the MovieLens data set provided by the GroupLens Research team. In addition, I would also like to thank the reviewers and editors for taking the time to review our paper.

References
[1] Yu Chuan, Xu Jieping, et al. Recommendation algorithm combining the user-based classified regression and the item-based filtering[C]. In Proceedings of the 8th international conference on Electronic commerce: The new e-commerce: innovations for conquering current barriers, obstacles and limitations to conducting successful business on the internet (ICEC '06). Association for Computing Machinery, New York, NY, USA, 574–578.
[2] Xiao Lin, Min Zhang, et al. Enhancing Personalized Recommendation by Implicit Preference Communities Modeling[J]. ACM Transactions on Information Systems. 2019, 37, 4, Article 48, 32 pages.
[3] Feng Xue, Xiangnan He, et al. Deep Item-based Collaborative Filtering for Top-N Recommendation[J]. ACM Transactions on Information Systems. 2019, 37, 3, Article 33, 25 pages.
[4] Wang Peng, Yang Jing, et al. A Strategy toward Collaborative Filter Recommended Location Service for Privacy Protection[J]. Sensors, 2018, 18(5), 1522.
[5] Rayan M. Nouh, Hyun-Ho Lee, et al. A Smart Recommender Based on Hybrid Learning Methods for Personal Well-Being Services[J]. Sensors, 2019, 19(2), 431.
[6] Xu Zhang. Research on Enterprise Knowledge Unified Retrieval Based on Industrial Big
Data[C]. In 2018 Sixth International Conference on Advanced Cloud and Big Data (CBD), Lanzhou, China. IEEE Computer Society. 2018, 130-134.

[7] Qibing Jin, Yue Zhang, et al. A New Similarity Computing Model of Collaborative Filtering [J]. IEEE Access. 2020, 8, 17594-17604.

[8] Peng Yu. Merging Attribute Characteristics in Collaborative Filtering to Alleviate Data Sparsity and Cold Start [C]. 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 2019, 569-573.

[9] Yu Zhu, Jinghao Lin, et al. Addressing the Item Cold-Start Problem by Attribute-Driven Active Learning [J]. IEEE Transactions on Knowledge and Data Engineering, 2020, 32(4):631-644.

[10] Fuzheng Zhang, Nicholas Jing Yuan, et al. Collaborative Knowledge Base Embedding for Recommender Systems[C]. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, 2016,353–362.

[11] Haifeng Liu, Zheng Hu, et al. A new user similarity model to improve the accuracy of collaborative filtering[J]. Knowledge-Based Systems, 2014, 56:156-166.

[12] Yixin Cao, Xiang Wang, et al. Unifying Knowledge Graph Learning and Recommendation: Towards a Better Understanding of User Preferences[C]. In The World Wide Web Conference (WWW ‘19). Association for Computing Machinery, New York, NY, USA, 2019,151–161.

[13] Ma Xiao, Lu Hongwei, et al. An exploration of improving prediction accuracy by constructing a multi-type clustering based recommendation framework[J]. Neurocomputing. 2016, 191:388-397.

[14] Liao Shaowen, Chen Yong. An Improved Collaborative Filtering Recommendation Algorithm[C]. 2017 International Conference on Smart Grid and Electrical Automation (ICSGEA), Changsha, 2017, 204-208.

[15] Gaojun Liu, Xingyu Wu. Using Collaborative Filtering Algorithms Combined with Doc2Vec for Movie Recommendation[C]. 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 2019, 1461-1464.

[16] P. Venil, G. Vinodhini, et al. Performance Evaluation of Ensemble based Collaborative Filtering Recommender System[C]. 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, 2019, 1-5.

[17] Xuansen He, Xu Jin. Collaborative filtering recommendation algorithm considering users’ preferences for item attributes[C]. 2019 International Conference on Big Data and Computational Intelligence (ICBDCI), Pointe aux Piments, Mauritius, 2019, 1-6.

[18] Meigen Huang, Yu Wang, et al. Collaborative Filtering Algorithm based on Linear Regression Filling[C]. 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 2019, 1831-1834.

[19] Raml G. Lumauag, Ariel M. Sison, et al. An Enhanced Recommendation Algorithm Based on Modified User-Based Collaborative Filtering[C]. 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), Singapore, 2019, 198-202.

[20] Yifan Chen, Yang Wang, et al. Local Variational Feature-Based Similarity Models for Recommending Top-N New Items[J]. ACM Transactions on Information Systems. 2019, 38, 2, Article 12, 33 pages.

[21] Qibing Jin, Xia Song, et al. Optimized CF Recommendation Algorithm Based on Users' Characteristics and Trust[C]. In Proceedings of the 2018 International Conference on Artificial Intelligence and Pattern Recognition (AIPR 2018). Association for Computing Machinery, New York, NY, USA, 120–125.