

Retraction

Retracted: Optimization of the Hybrid Movie Recommendation System Based on Weighted Classification and User Collaborative Filtering Algorithm

Complexity

Received 23 January 2024; Accepted 23 January 2024; Published 24 January 2024

Copyright © 2024 Complexity. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] Z. Yuan, J. H. Lee, and S. Zhang, "Optimization of the Hybrid Movie Recommendation System Based on Weighted Classification and User Collaborative Filtering Algorithm," *Complexity*, vol. 2021, Article ID 4476560, 13 pages, 2021.

Research Article

Optimization of the Hybrid Movie Recommendation System Based on Weighted Classification and User Collaborative Filtering Algorithm

Zhenning Yuan,^{1,2} Jong Han Lee,² and Sai Zhang ³

¹Academic Affairs Office, Qingdao Agricultural University, Qingdao 266109, China

²Department of Formative Convergence Arts, Huxi University, Seoul 31499, Republic of Korea

³Animation and Communiton College, Qingdao Agricultural University, Qingdao 266109, China

Correspondence should be addressed to Sai Zhang; 201301049@qau.edu.cn

Received 29 April 2021; Revised 17 May 2021; Accepted 28 May 2021; Published 3 June 2021

Academic Editor: Zhihan Lv

Copyright © 2021 Zhenning Yuan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aiming at the problem that the single model of the traditional recommendation system cannot accurately capture user preferences, this paper proposes a hybrid movie recommendation system and optimization method based on weighted classification and user collaborative filtering algorithm. The sparse linear model is used as the basic recommendation model, and the local recommendation model is trained based on user clustering, and the top- N personalized recommendation of movies is realized by fusion with the weighted classification model. According to the item category preference, the scoring matrix is converted into a low-dimensional, dense item category preference matrix, multiple cluster centers are obtained, the distance between the target user and each cluster center is calculated, and the target user is classified into the closest cluster. Finally, the collaborative filtering algorithm is used to predict the scores for the unrated items of the target user to form a recommendation list. The items are clustered through the item category preference, and the high-dimensional rating matrix is converted into a low-dimensional item category preference matrix, which further reduces the sparsity of the data. Experiments based on the Douban movie dataset verify that the recommendation algorithm proposed in this article solves the shortcomings of a single algorithm model to a certain extent and improves the recommendation effect.

1. Introduction

With the rapid development of information technology and social networks, the data generated by the Internet has risen exponentially in recent years, and the era of big data is coming. With the increase of data, it is more and more difficult for people to find the information they really want from the massive data. At this time, the recommender system can play the maximum application value [1, 2]. According to the user information and user historical behavior data, the recommendation system can accurately predict the user's preferences, personalize the things that users may be interested in, and greatly reduce the cost of finding target information. Content-based collaborative filtering algorithm (CBF) and traditional collaborative

filtering algorithm (CF) have their own shortcomings [3, 4]. When a new project is added to the system, but its project characteristics cannot be obtained or described, CBF cannot be used at this time. Recommender system makes up for the deficiency of search engine. It does not need users to put forward clear requirements. Instead, it recommends information that meets users' personalized needs to users by analyzing users' historical behavior, applying recommendation algorithm, or establishing users' interest model [5, 6]. In the research of recommender system, people focus on the improvement of recommender algorithm. At present, the mainstream recommendation algorithms include the following categories: collaborative filtering-based recommendation, content-based recommendation, and hybrid recommendation. Although

collaborative filtering algorithm has been widely used in movie recommendation, music recommendation, and other fields, it still has some problems such as data sparsity, user interest change, cold start, and scalability.

Different user groups often form their own unique behavior patterns. For example, the users who love military war tend to score more in military movies, but their scoring behavior in literary movies is relatively sparse. On the contrary, the users who love literature and art have more scoring behavior in literature and art movies, but less in military movies [7]. For some popular movies, no matter what the theme is, users in each user group are usually very active in their scoring behavior. The neighborhood-based collaborative filtering is to calculate the similarity between items by analyzing the user's behavior. It thinks that item A and item B have great similarity because most users who like item A also like item B. This means that the similarity between popular movies and war movies is different in the above two user groups, similar in the military enthusiast group, and not similar in the literature and art enthusiast group [8, 9]. There are also many people who have studied the hybrid recommendation method. Froelich and Hajek combined the domain model-based algorithm with the matrix decomposition based algorithm in collaborative filtering algorithm [10]. Su et al. used photos uploaded by users and preference photos as mixed preferences to study the characteristics of users [11]. Bobadilla et al. proposed the multidimensional matrix factorization model, combined with collaborative filtering (CF) algorithm, and used user and project attributes to make score prediction, which improved the accuracy of prediction score [12]. The first mock exam algorithm usually constructs a single global model for all users [13–15]. It is considered that the similarity of two identical items in any group is the same. Obviously, this single model cannot find the difference of similarity among items in different user groups and thus cannot accurately capture user preferences and recommend the general effect.

The recommendation algorithm proposed in this paper aims to solve the above problems by constructing a local model. In order to solve the problem that a single model cannot accurately capture user preferences, this paper proposes a local model weighted fusion recommendation algorithm based on user clustering [16]. The local model is trained by user subgroup partition, and finally the global model weighted fusion is used to improve the quality of recommendation. Because the local model only uses the local training data in the training process, it will lose the same important global information, so the fusion algorithm proposed in this paper is the linear weighted fusion between the local model and the global model. The local model plays an auxiliary and correction role in the prediction of the global model. The importance of different models to the fusion model is controlled by adjusting the weight parameters. In addition, in order to achieve the division of user subgroups, this paper uses the text content information of the movie, proposes to use LDA topic model to calculate the user feature vector, and uses spectral clustering algorithm to achieve user clustering based on the feature vector. The recommendation

algorithm proposed in this paper is an effective combination of content-based recommendation, neighborhood-based collaborative filtering, and model-based collaborative filtering, so it has the advantages of interpretable recommendation results, fast recommendation speed, and high recommendation quality. The effectiveness of the proposed recommendation algorithm is proved by several experiments using the movie dataset from douban.com.

2. User Collaborative Filtering Recommendation Algorithm Based on Weighted Classification

2.1. Recommendation Based on Weighted Classification. The core of recommendation based on weighted classification is to explore the explicit or implicit relationship between items. E-commerce recommendation system uses this method to explore the relationship between the combinations of goods that users are interested in and generate recommendation list according to the purchase rules of users. "Beer and diapers" is a classic case of this method [17]. When fathers go to the supermarket to buy diapers for their babies, they usually buy some beer by the way. After the supermarket finds this rule, they put diapers and beer in nearby places, and eventually their sales have been increased. The supermarket finds the potential combination of beer and diaper by studying the user's behavior, so the weighted classification method realizes the cross selling of goods by predicting the items that the user may be interested in.

Support value is an important index in weighted classification, which represents the proportion of the union of items contained in set X and set Y in the whole transaction set, expressed as $Sp(Y \geq X)$. P is the set of trading goods, and $P \in I$. Suppose X is a subset of set I . If $X \in P$, then x is included in the trading set. Y is another subset of set I , and $X \cap Y = \Phi$; the weighted classification of X , Y can be expressed by $Y \geq X$.

$$Sp(Y \geq X) = \frac{|P: Y \cup X \subset P|}{|P| + 1}. \quad (1)$$

Another index of weighted classification is confidence, which represents the percentage of transaction set P containing both set X and set Y , expressed as Confidence($Y \geq X$):

$$\text{Confidence}(Y \geq X) = \frac{|N: Y \cup X \subset N| + 1}{|N: Y \subset N| - 1}. \quad (2)$$

Firstly, the weighted classification technology sets a threshold to extract the items whose frequency is higher than the threshold in the original item set, that is to find out all the items whose support is higher than the threshold. Secondly, the weighted classification is found from the selected items, and the confidence between items is calculated by corresponding rules. If the confidence is higher than the preset confidence threshold, the two items are considered to be related.

The trial range of weighted classification technology is relatively wide, but it also has some defects; that is, the quality of the recommendation results has a great

relationship with the accuracy of the selection of support and confidence [18–20]. If the above indicators are selected accurately, it will produce a better recommendation effect; on the contrary, the effect is poor. Content-based recommendation is widely used in the field of recommendation, such as e-commerce recommending products for customers, movie and video websites recommending movies for users, and news websites recommending news for users. The core idea of CBR is to recommend the items closest to users' favorite content [21–24]. Taking the movie recommendation system as an example, the system constructs a feature vector for the user through the movie type features (such as love, science fiction, action, and other information) that the user has evaluated or watched and then recommends several movies with the highest similarity to the feature vector to the user. The schematic diagram of content-based recommendation is shown in Figure 1.

User A is interested in love and romance, while users B and C are interested in horror and horror. When using the content-based recommendation algorithm to recommend to user A, it is found that the feature vectors of movies A and C are close, so movie C is finally recommended to user A.

2.2. Weight Calculation of Score Difference Degree. Simple cosine similarity calculation and modified cosine similarity calculation are the approximation calculation methods often used in user-based collaborative filtering algorithms [25–27]. In the calculation of user approximation, the traditional cosine similarity is not sensitive to distance because it represents the difference in direction. If two users keep the same trend of scoring items, but the score is one high and one low, it will lead to a certain deviation of user similarity. The following is an example to illustrate the disadvantages of traditional cosine similarity calculation. The user item score matrix is shown in Table 1.

For the scoring matrix in Table 1, the traditional modified cosine similarity is used to calculate the similarity between U_1 and U_2 . The result shows that the similarity between them is $\text{sim}(U_1, U_2) = 1$; that is, the user preferences between U_1 and U_2 are completely consistent. However, in fact, U_1 score is generally low, and U_2 score is generally high. So, their user preferences are not exactly the same. To sum up, the traditional cosine similarity is not sensitive to distance, and it cannot accurately represent the approximation between users. In view of the above problems, this paper proposes an improved approximation calculation method based on score difference weight, which can effectively solve the problem that the traditional cosine similarity calculation is not sensitive to distance.

The steps of the improved similarity calculation method are as follows:

- (1) Define the set of all items $I = \{I_1, I_2, \dots, I_n\}$, and user u scores all items $U_1 = \{\text{Ru}1I1, \text{Ru}1I2, \dots, \text{Ru}1In\}$. The score set of user U_2 for all items is $U_2 = \{\text{Ru}2I1, \text{Ru}2I2, \dots, \text{Ru}2In\}$. The formula for calculating the score difference between U_1 and U_2 is defined as follows:

$$\begin{aligned} \{D_1, D_2 \dots D_n\} &= f(u_1, u_2) \\ &= \{Pu_1I_i - Pu_2I_i\}, \quad i = 1, \dots, n. \end{aligned} \quad (3)$$

- (2) The standardized Euclidean distance is used to replace the score difference between U_1 and U_2 , and the calculation formula is shown in the following equation:

$$\text{dif}(u_1, u_2) = \lim_{n \rightarrow \infty} \sqrt{\sum_{s=1}^n \left(\frac{d_s^2}{t_s} \right)^2 + \mu s}, \quad (4)$$

where sk is the variance of components. If the reciprocal of variance is regarded as a weight, this formula can be regarded as a weighted Euclidean distance.

- (3) Considering the influence of the common scoring items of users U_1 and U_2 on the similarity between users, when the number of common scoring items is more, the user's approximation is greater, and the number of common scoring items n is taken into account. When $\text{dif}(u_1, u_2)$ is larger, the user approximation is lower. Considering that the denominator cannot be 0, that is, there is no common scoring item, the formula still has the same meaning. According to equation (5), the range of $\text{dif}(u_1, u_2)$ is $(0, +\infty)$.

$$\text{dif}(u_1, u_2) = \frac{\lim_{n \rightarrow \infty} \sqrt{\sum_{s=1}^n (d_s^2/t_s)}}{k+1}. \quad (5)$$

- (4) $\text{dif}(u_1, u_2)$ is normalized, and the corresponding calculation formula is shown in the following equation:

$$\text{dif}(u_1, u_2) = e^{-(k+1)} \lim_{n \rightarrow \infty} \sqrt{\sum_{s=1}^n (d_s^2/t_s) + \lambda e}. \quad (6)$$

The improved similarity calculation formula is shown in the following equation:

$$\text{sim}(m, n) = \text{dif}(m, n) \times \text{cosine}(m, n). \quad (7)$$

$\text{Sim}(m, n)$ is the similarity of users I and j calculated by the traditional cosine similarity formula.

2.3. Implementation of Movie Recommendation Results.

Based on user collaborative filtering recommendation algorithm, the user's network operation or rating behavior records are used to find the nearest neighbor user set consistent with the target user's interests and preferences and recommend the items that the user is most interested in but the target user has not browsed to the target user. If the interest preferences of user N and user n^* are the same, and user n^* 's item recommendation list (the list is sorted by preference, the higher the order, the more user's preference) is c^* , then user n^* of the first several items in list c^* may also have preferences, and they can be recommended to user n^* .

The flow of recommendation algorithm based on user collaborative filtering is shown in Figure 2.

2.3.1. Getting the Nearest Neighbor Set of the Target User. According to the similarity between the target user and other users, other users are sorted according to the similarity from large to small. Obviously, the higher the ranking, the more similar the interest preferences of other users and target users, and the lower the ranking, the less similar the interest preferences. Therefore, the top k users in other users' ranking are selected as the nearest neighbor set of target users $A_n = \{A_1, A_2, \dots, A_k\}$, $k = 1, 2, \dots, K$.

2.3.2. Calculating the Forecast Score of Target Users. The prediction score based on memory collaborative filtering recommendation algorithm is usually calculated by weighted average of similarity:

$$q_{mn} = \frac{\lim_{s \rightarrow \infty} \sum_{s=1}^t \text{SIM}(m, t) \times q_{sn}}{\lim_{s \rightarrow \infty} \sum_{s=1}^t \text{SIM}(m, s)}. \quad (8)$$

The specific process of the algorithm to generate movie recommendation list is as follows: using the user clustering method of item category preference to find the user space with similar category preference for the target user. Then, using the improved similarity calculation method of fusion score difference weight, the top k users with the highest similarity with the current user are selected as the nearest neighbors of the current user.

3. Implementation and Optimization of the Hybrid Movie Recommendation System

3.1. System Overall Design

3.1.1. System Architecture Design. The recommender system adopts the classic B/S architecture mode, which can be divided into three parts: presentation layer, business logic layer, and data access layer. The advantage of layering is that each layer can be separated from each other. Each layer does not know each other's internal information and only connects through the interface of each layer. When developers develop the system, they only need to pay attention to the function of this layer and what kind of output they provide to other layers through the interface and do not need to consider the internal structure of other layers. When the system has reused requirements or system functions need to be modified, we do not need to modify the whole system; we only need to modify the content of the corresponding layer to meet the requirements. Therefore, the system with three-tier architecture has the advantages of easy reuse, loose coupling, and high degree of cohesion, which has been widely used in industry.

Presentation layer is the front-end page visible to users. Users interact with the server through the presentation layer, and the input data is transferred to the business logic layer through this layer. Ordinary users can register, log in, evaluate, and get recommendation in the presentation layer.

The administrator can add, delete, modify movies, and log off users in the presentation layer.

Business logic layer is, also known as the business layer; its main function is to control business data. Business data mainly include scoring information, user basic information, and project basic information. The recommendation module of this layer uses the above data to calculate the recommendation results and recommend them to the target users. The main functions of this layer are offline computing and online recommendation.

Data access layer, also known as the data layer, implements the operation of the database, mainly including the addition, deletion, modification, and query of the database. The business layer extracts data from the data layer for calculation, then accesses the data layer, and stores the calculated results in the database. The data processed by the data layer is mainly composed of scoring information, user information, project information, recommendation list, and so on. The overall architecture of the system is shown in Figure 3.

3.1.2. The Overall Module Design of the System. This movie recommendation system mainly includes four parts: user function module, administrator function module, offline calculation module, and online recommendation module. The overall system module is shown in Figure 4.

3.2. Database Design. According to the function requirement analysis of the system and the algorithm design of this paper, we can make the E-R diagram of the system as shown in Figure 5.

According to the E-R diagram of the system, we can design the database into the following five tables.

User information table: this table stores user related information, mainly including user ID, user name, age, and occupation. User ID is the primary key. The specific design of the table is shown in Table 2.

Project information table: this table stores the relevant information of the movie, mainly including movie, movie name, movie type, release time, and picture path. The movie ID is the primary key. The specific design of the table is shown in Table 3.

Scoring table: this table stores the scoring information of users for movies, mainly including user ID, movie ID, scoring, and scoring time. User ID and movie ID are the primary keys. The specific design of the table is shown in Table 4.

Similarity table: this table stores cosine similarity and other information between items, mainly including movie ID1, movie ID2, and similarity. The combination of movie ID1 and movie ID2 is the primary key. The specific design of the table is shown in Table 5.

3.3. Design of the Recommendation Function Module. Recommendation function module mainly includes offline data calculation module and online intelligent recommendation module.

The main function of the offline data calculation module is to calculate the results in advance for the online intelligent

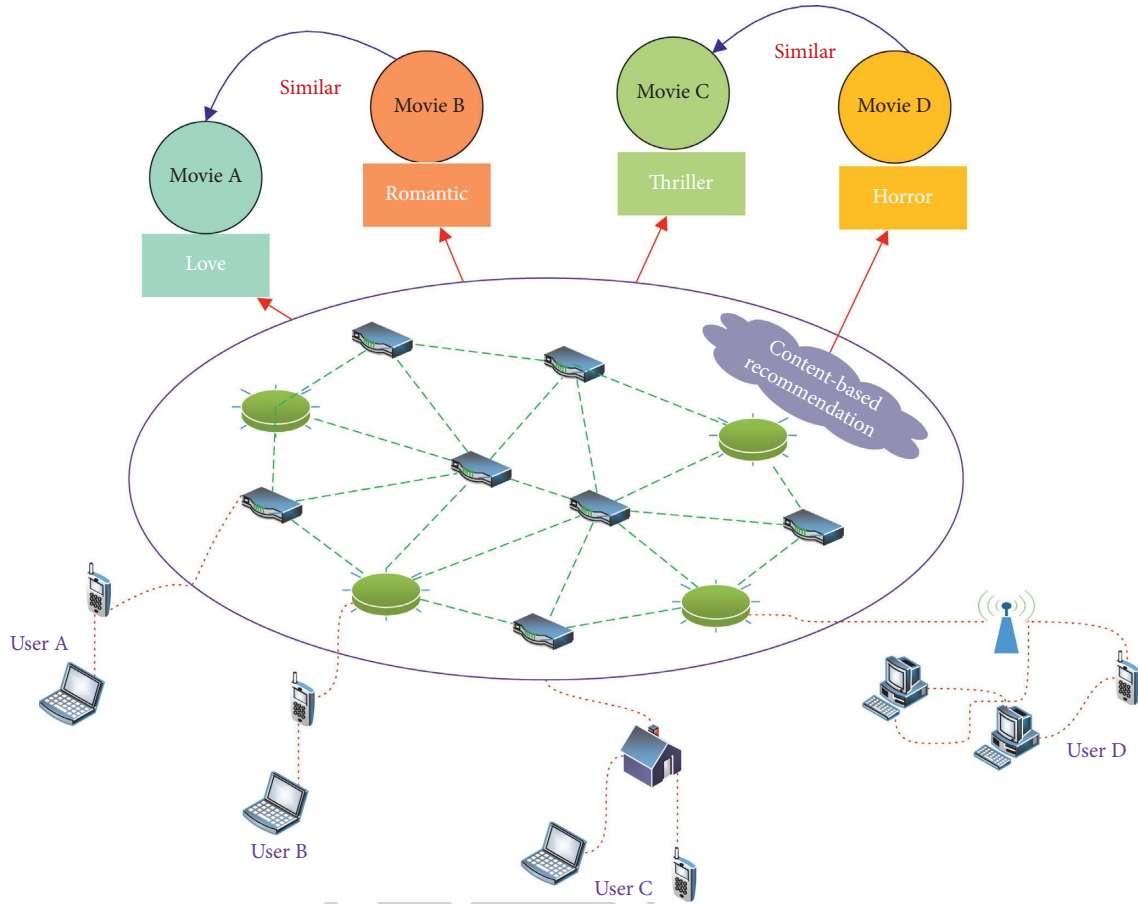


FIGURE 1: Content-based recommendation principle.

TABLE 1: User project rating matrix.

	I_1	I_2	I_3	I_4	I_5
U_1	3	3	3	3	3
U_2	2	2	2	2	2
U_3	4	4	4	4	4
U_4	4	4	4	4	4

recommendation module. The module first needs to extract the score information and other data from the database and store it in the relevant HashMap class objects. This system uses the improved fusion algorithm mentioned above. At this stage, it needs to generate a prediction score matrix by biased SVD for subsequent algorithms. During the operation of the algorithm, it needs to use the gradient descent method to generate new parameters, and the iterative error function is used to minimize it. Finally, the prediction score matrix is generated. In addition, the cosine similarity between items should be calculated according to the complete user item score matrix. Finally, the calculated results are stored in the database for online recommendation. The computation of the above two parts is very large and time consuming. If the complete algorithm is run online, the real-time performance of the system will be greatly affected. The usual solution is to calculate the results offline, which is convenient for the

online module to call. We can recalculate the prediction score matrix and similarity matrix every other period of time because the user's behavior has little effect on the above calculation results in a short time, and the offline calculation reduces the load of the server on the basis of ensuring the accuracy of the recommendation.

The online intelligent recommendation module can be divided into two parts: personalized online recommendation and the most popular movie recommendation. In the personalized recommendation part, based on the results of offline computing module, the final prediction score matrix is generated according to the improved algorithm, and the K movies with the highest prediction score are recommended as recommendation items. The most popular movie recommendation is to calculate the average score of all items in the scoring matrix and recommend the top k movies with the highest average score as the recommended items.

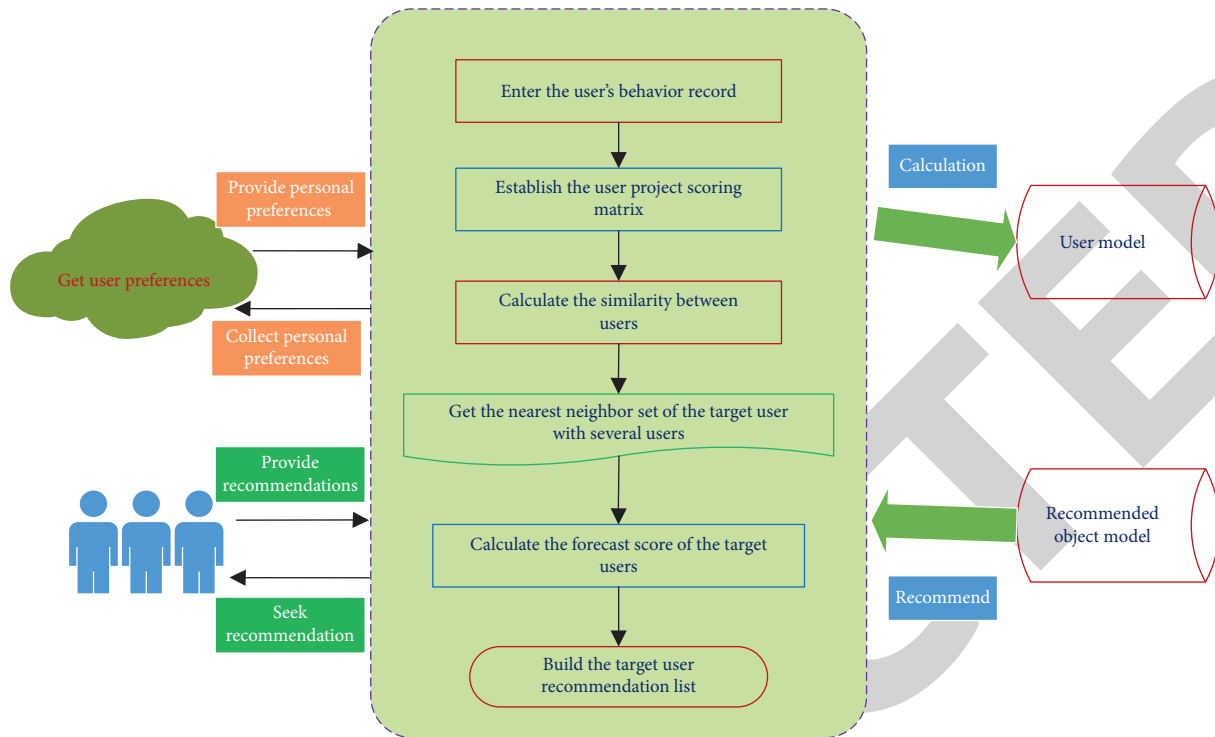


FIGURE 2: User collaborative filtering algorithm recommendation system model.

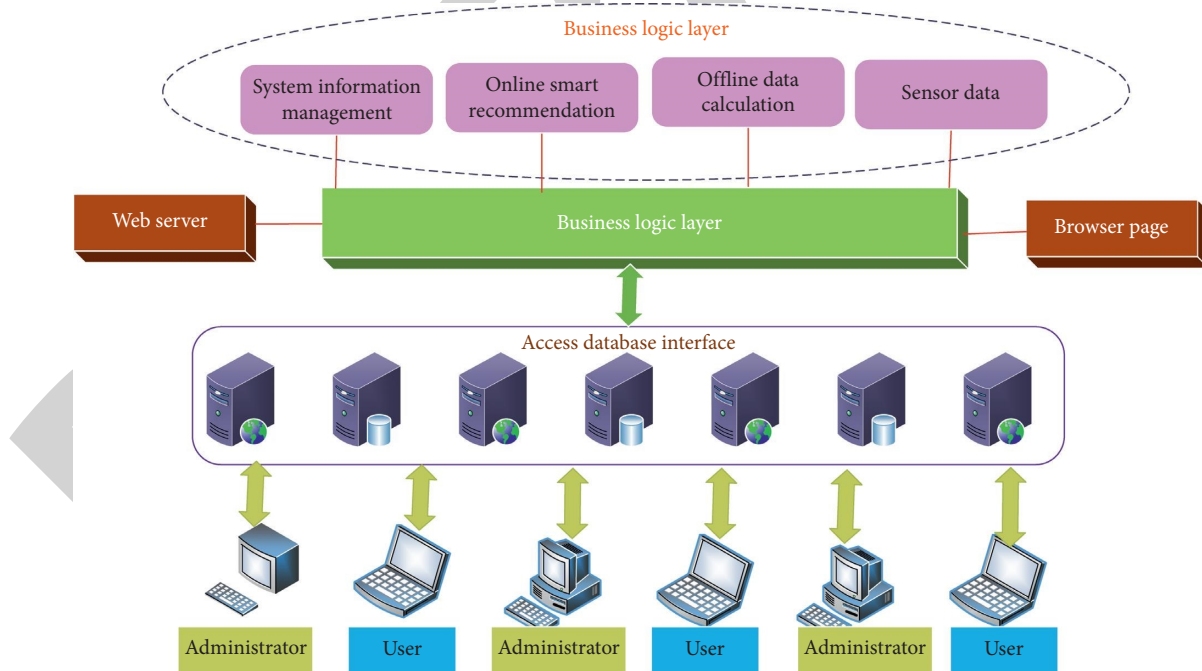


FIGURE 3: Overall system architecture.

In the initial stage of the design, considering the system scalability and other issues, the prediction algorithm, the operation of the database, datasets, and so on are written in different classes, so that the classes are independent of each other, which is convenient for future modification and

replacement. At the same time, the interface of different prediction scoring algorithms is implemented, including the method based on the improved fusion model and the method of calculating the average score of the most popular movies. In the subsequent process of enriching the system

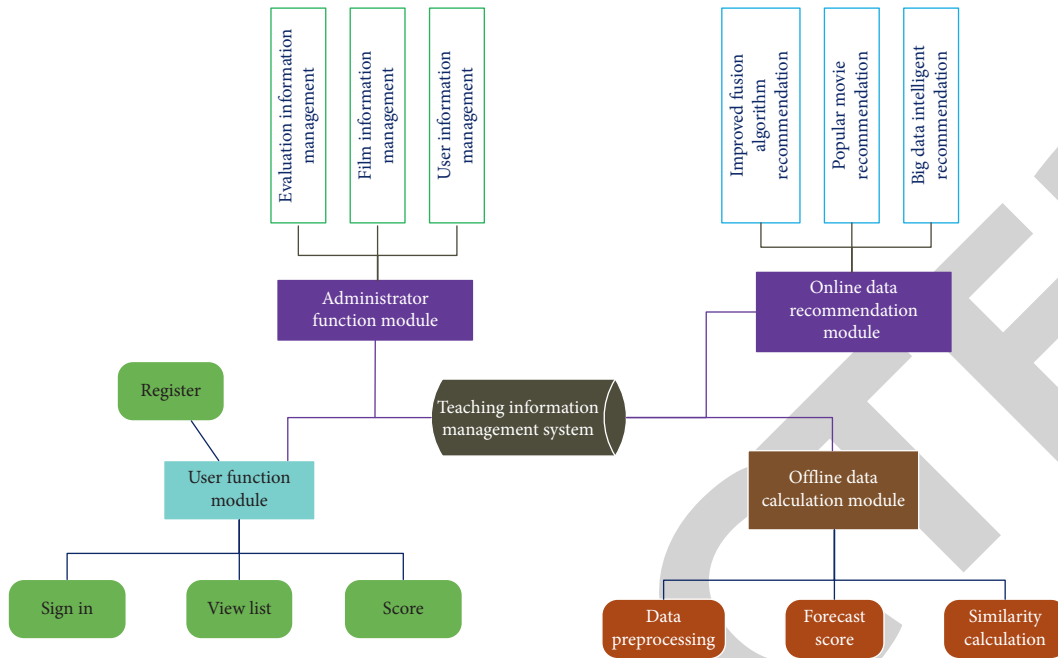


FIGURE 4: System general function module.

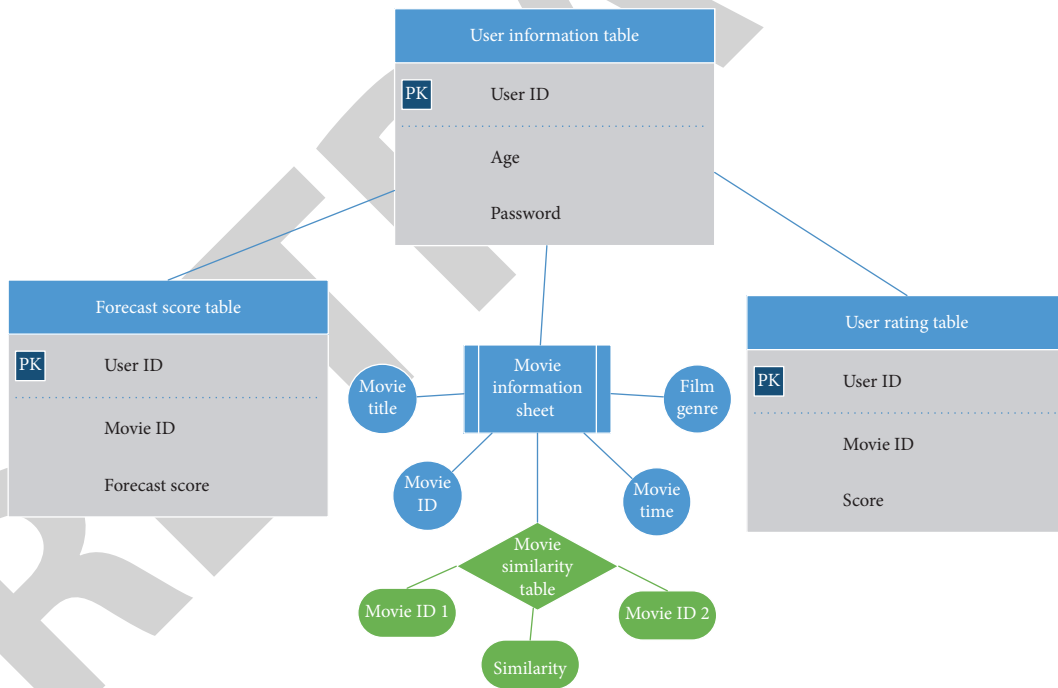


FIGURE 5: System E-R diagram.

TABLE 2: User information database table.

Title	Data type	Notes
ID	Int (12)	User ID
NM	Varchar (60)	User name
PW	Varchar (60)	Password
AG	Int (12)	Age
GD	Bool	Gender
OP	Varchar (30)	Occupation

TABLE 3: Movie information database.

Title	Data type	Notes
ID	Int (16)	Movie ID
TL	Varchar (255)	Movie title
RD	Date	Issue date
IU	Text	Viewing address
PC	Text	Picture path
TP	Text	Film genre
DT	Varchar (30)	Director
AT	Varchar (30)	Performer

content, different recommendation algorithms may be applied according to different scenes. This interface can be used to fill in the algorithm conveniently.

4. Experiment and Result Analysis

4.1. Experimental Datasets and Evaluation Methods

4.1.1. Introduction to Datasets. This paper uses the dataset crawled from douban.com to verify the proposed recommendation model. The initial dataset included 3328 users, 28615 movies, and 389184 ratings. In order to better evaluate the performance of the recommendation model, the dataset is cleaned to ensure that a movie has been watched by at least 20 users, and a user has watched at least 15 movies. Finally, the experimental dataset includes 3156 users, 3524 movies, 302673 ratings, and 4232 movie tags. The explicit scoring information is transformed into implicit 0-1 feedback information. As long as the user has seen the movie, the corresponding item is set to 1; otherwise, it is set to 0, and the original training matrix A is generated. Dimension reduction is one of the important research branches in many research fields, and its methods are diverse. According to the different dimension reduction methods, many clustering methods based on dimension reduction are produced, such as Kohonen self-organizing feature mapping, principal component analysis, multidimensional scaling, and so on.

4.1.2. Evaluation Method and Index. In this paper, we use leave one method to verify the validity of the model. One movie is randomly selected from each user's movie score set and put into the test set. Other movies are used as the training set of the model. Then, use the trained model to recommend a top- N movie list for each user, observe whether the movie corresponding to the user in the test set appears in the recommended list and the specific location of the movie in the list. Finally, we use the hit rate (HR) and average rank hit rate (ARHR) to measure the recommendation quality of the model. The evaluation indexes of recommender system mainly include accuracy, coverage, and diversity. According to different recommendation methods, different evaluation indexes are selected. Because the ultimate goal of the improved algorithm in this paper is to generate a movie recommendation list for current users in line with their interests and preferences, rather than predict how much the target users will score for the movie, so this paper adopts the form of top- N recommendation list when

recommending movies for the target users. The commonly used important indicators to measure the accuracy of recommendation system are accuracy and recall, so this paper uses accuracy and recall as the evaluation metrics of recommendation algorithm.

First of all, we need to determine the number of specific topics. Therefore, at first, a group of topic numbers {5, 6, 7, 8, 9, 10, 11, 15} are randomly set to train multiple topic models, and then the average similarity between topic vectors generated by each topic model is calculated (cosine similarity is used here), and the model with the lowest average similarity is taken as the final model. After experimental verification, when the number of topics in the current dataset is set to 10, the average similarity is the lowest, which is 0.645, so the number of topics is 10. In order to improve the training speed of the model, the online LDA topic model proposed by Hoffman et al. is adopted, which reduces the training time of traditional LDA from hour level to second level. Finally, 3155 10-dimensional user feature vectors and 4221 10-dimensional topic vectors are obtained.

When spectral clustering algorithm realizes user clustering, it needs to determine the number of clusters first. Because each dimension of each user vector obtained by training represents the membership degree of the user belonging to the corresponding topic, in order to determine the importance of each topic in the current user group, the feature vectors of all users are accumulated according to the dimensions and then averaged to obtain a 10-dimensional topic intensity vector, which is visualized as shown in Figure 6. It can be seen that themes 2, 9, 3, 8, and 6 have the highest intensity in the current dataset, indicating that most people like to watch these types of movies. Therefore, this paper uses spectral clustering algorithm to cluster users into five categories (Figure 7).

4.2. Comparative Experiments with Other Recommendation Algorithms. For each user clustering, we train a local sparse linear recommendation model and a global sparse linear recommendation model and determine the optimal parameters through cross validation, so that the experimental indicators of each model are optimal. The training results are listed in Table 6.

Local model and global model are combined to recommend top- N movies' list to each user. In order to determine the optimal global weight parameter g , the 0-1 interval is divided into 101 parts, and the fusion experiments are carried out, respectively. The experimental results are

TABLE 4: Rating information data sheet.

Title	Data type	Notes
UID	Int (15)	User ID
MID	Int (15)	Movie ID
RT	Int (15)	Score
TM	Int (15)	Scoring time

TABLE 5: Similarity database.

Title	Data type	Notes
MI1	Int (12)	Movie ID1
MI2	Int (12)	Movie ID
SL	Double	Similarity

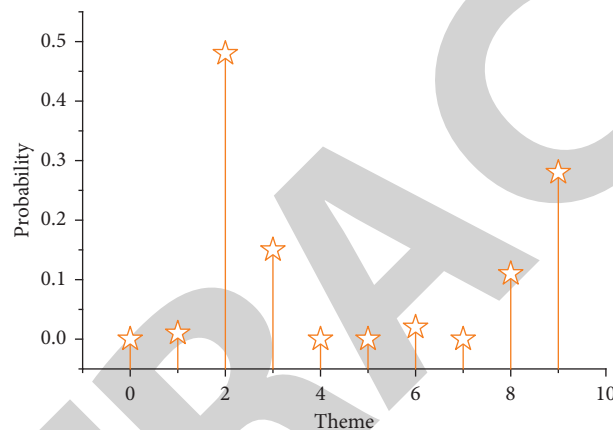


FIGURE 6: Theme intensity distribution.

shown in Figure 8. When $g = 1.0$, the fusion model is a single global model; when $g = 0$, the fusion model is a single local model. It can be seen from the figure that when G is 0.53, HR reaches the highest value, which is 0.205388, and ARHR is 0.091972, which surpasses the single global model recommendation algorithm. HR has been improved by 5.85% and ARHR has been improved by 4%.

In addition, when G is in the range of 0.5 ~ 0.8, the experimental effect is the best, which shows that in the process of model fusion, the global model is the main, and the local model is the auxiliary, and the local model corrects the deficiency of the global model in prediction. This experiment proves the importance and necessity of constructing local model with LM.

The proposed recommendation algorithm is compared with several other classical recommendation algorithms, namely, heat-based recommendation (TopPop), user-based recommendation (UserKNN), item-based recommendation (ItemKNN), weighted regularized matrix factorization (WRMF), and sparse linear model recommendation (SLIM). The first three recommendation algorithms are the classic recommendation algorithms in the development of recommendation system. Because of their simple implementation, they are widely used. WRMF and SLIM are new recommendation algorithms proposed in recent years. They

are matrix factorization and linear fitting training recommendation model, respectively, which can predict user preferences more accurately and achieve very good recommendation results.

TopPop recommends the top n movies with the highest popularity to users. The popularity here is determined by the number of people who score the movies. The more people who score the movies, the more popular they are. This paper uses HR and ARHR to measure the recommendation quality of the model. The experimental results on the Douban dataset are shown in Tables 7 and 8 and Figures 9 and 10.

It can be seen from the figure that the recommendation effects of WRMF, SLIM, and LM are far better than TopPop, UserKNN, and ItemKNN in both HR and ARHR. Among them, the recommendation effect of TopPop is the worst because it is not personalized recommendation, and it recommends the most popular movies to every user. The UserKNN and ItemKNN are not modeled by machine learning and only get the surface user similarity or movie similarity, which leads to the poor recommendation quality. Among the three model-based recommendation algorithms WRMF, SLIM, and LM, WRMF implements recommendation by matrix decomposition, SLIM is the basic recommendation model of LM, and each group of experiments shows that the recommendation effect of the proposed local

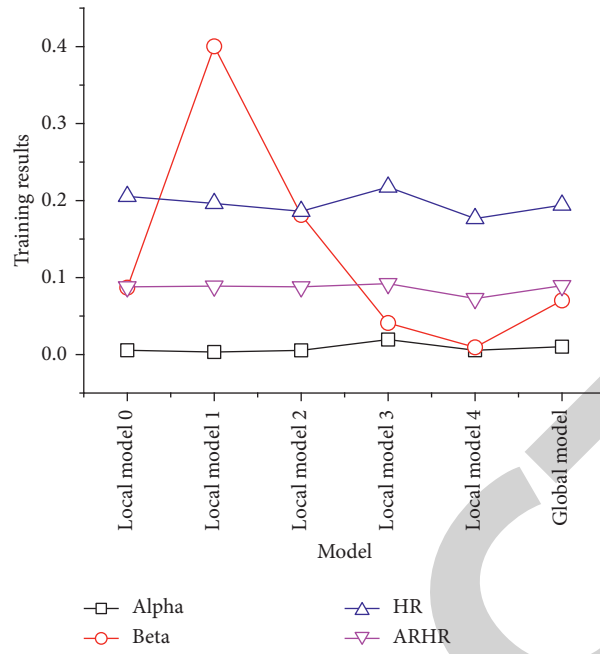


FIGURE 7: Training results of the model.

TABLE 6: Training results of the local model and global model.

	Number of users	α	ρ	HR	ARHR
Local model 0	383	0.005521	0.086881	0.205412	0.087889
Local model 1	667	0.003212	0.400263	0.196233	0.088848
Local model 2	982	0.005421	0.181228	0.185994	0.087893
Local model 3	447	0.019458	0.040953	0.217533	0.092136
Local model 4	723	0.005522	0.009372	0.176673	0.072726
Global model	3266	0.010204	0.0702015	0.194146	0.089525

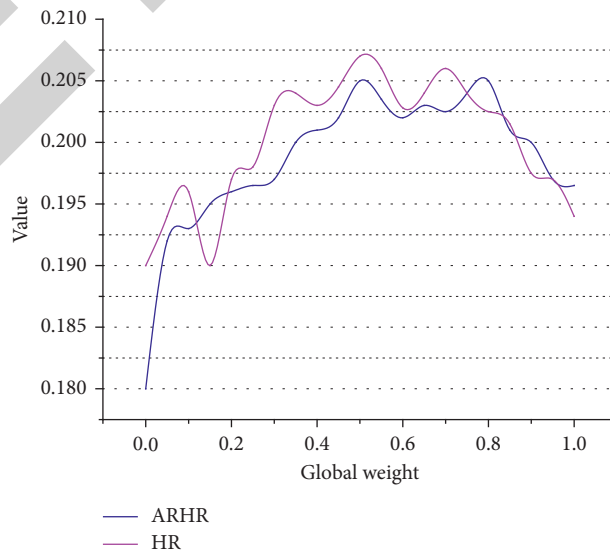


FIGURE 8: Fusion results using different global weights.

TABLE 7: Comparison of HR indicators on datasets.

	TopPop	UserKNN	ItemKNN	WRMF	SLIM	LM
Top-10	0.01	0.02	0.05	0.12	0.13	0.15
Top-15	0.02	0.035	0.07	0.2	0.21	0.23
Top-20	0.04	0.06	0.1	0.26	0.255	0.27

TABLE 8: Comparison of ARHR indicators on datasets.

	TopPop	UserKNN	ItemKNN	WRMF	SLIM	LM
Top-10	0.005	0.005	0.021	0.08	0.081	0.0818
Top-15	0.007	0.007	0.03	0.085	0.088	0.094
Top-20	0.008	0.01	0.035	0.095	0.09	0.098

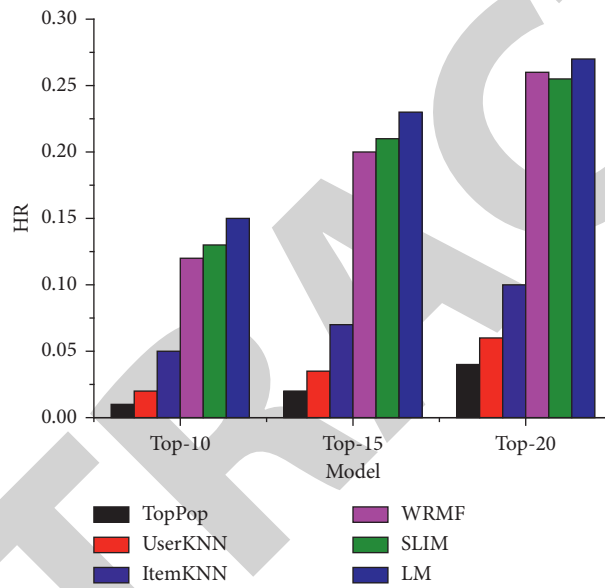


FIGURE 9: The contrast experiment of the HR index.

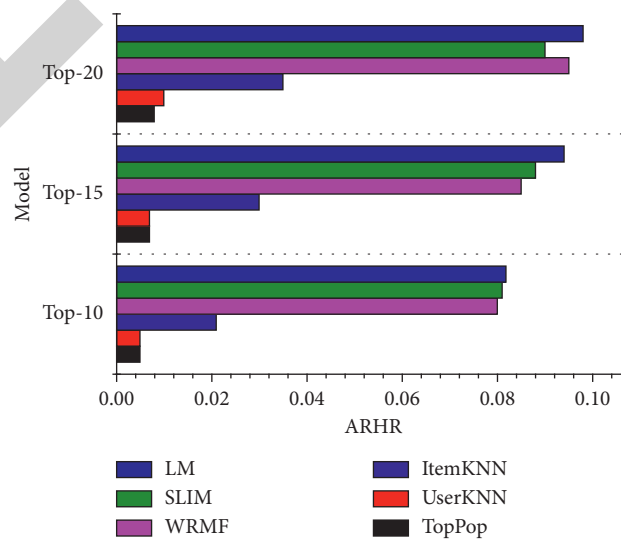


FIGURE 10: Comparison of ARHR indexes.

weighted fusion recommendation algorithm is better than the other two recommendation algorithms, which further verifies the effectiveness of the proposed recommendation algorithm.

5. Conclusion

With the rapid development of Internet technology, the amount of information is growing at an explosive speed. Users are usually helpless in the face of how to obtain effective information more efficiently. It is difficult for them to find the information they are interested in simply and quickly. The birth of the personalized recommendation system provides users with a passive way to obtain information and makes up for the lack of search engine which can only provide the same information, which can provide personalized information for users. At present, personalized recommendation system has been widely used in video websites, music websites, e-commerce, news reading websites, and other fields and has attracted more and more attention from scholars and industry. This paper proposes a hybrid movie recommendation system optimization based on weighted classification and user collaborative filtering algorithm. The research focus of the algorithm is to consider the user's behavior information and item category preference information at the same time. Firstly, the user's web log is obtained. At the same time, according to the access time of the item obtained in the web log, the user's recent behavior information is obtained. The recent behavior information reflects the user's current interest. The behavior information is transformed into user's score of the item, and the score matrix is filled with the transformed score. The sparsity of the filled score matrix is reduced to a certain extent compared with the previous one. Secondly, according to the item category preference, the scoring matrix is transformed into a low-dimensional and dense item category preference matrix to obtain multiple clustering centers. The distance between the target user and each clustering center is calculated, and the target user is classified into the nearest clustering. Finally, the collaborative filtering algorithm is used to predict the score of the target user's unsatisfied items and form a recommendation list. The innovation of the first mock exam is that the traditional recommendation system cannot capture user preferences accurately. A hybrid movie recommendation system and optimization method based on weighted classification and user collaborative filtering algorithm are proposed. The sparse linear model is used as the basic recommendation model, and the local recommendation model is trained based on user clustering. Finally, the top- N personalized recommendation of movies is realized by fusing with the weighted classification model. In this paper, user behavior information is used to fill the scoring matrix, which alleviates the data sparsity to a certain extent. By clustering items by item category preference, the high-dimensional rating matrix is transformed into a low-dimensional item category preference matrix, which further reduces the sparsity of data, and finally improves the recommendation accuracy of the recommendation algorithm. Based on the analysis of functional requirements, the overall

framework of the system and the design of each functional module are completed, and the basic functions such as personalized recommendation, popular movie recommendation, evaluation movie, and input movie are realized, and the above functions are displayed through the page.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] R. Shukla, S. Silakari, and P. K. Chande, "An improved recommendation approach based on user's standard and interests," *International Journal of Computer Applications*, vol. 119, no. 1, pp. 6–14, 2015.
- [2] R. Bhuvanya, K. Vijayalakshmi, and S. Uma, "Movie recommender system using two way filtering and agglomerative hierarchical clustering," *Journal of Computational and Theoretical Nanoscience*, vol. 15, no. 6, pp. 2269–2272, 2018.
- [3] Y. Yun, D. Hooshyar, J. Jo, and H. Lim, "Developing a hybrid collaborative filtering recommendation system with opinion mining on purchase review," *Journal of Information Science*, vol. 44, no. 3, pp. 331–344, 2018.
- [4] H. Alhoori and R. Furuta, "Recommendation of scholarly venues based on dynamic user interests," *Journal of Informetrics*, vol. 11, no. 2, pp. 553–563, 2017.
- [5] R. Chen, Q. Hua, Y.-S. Chang, B. Wang, L. Zhang, and X. Kong, "A survey of collaborative filtering-based recommender systems: from traditional methods to hybrid methods based on social networks," *IEEE Access*, vol. 6, pp. 64301–64320, 2018.
- [6] F. Pemisindo and N. Legowo, "Hybrid recommendation system with clustering and classification method based on case of clinic XYZ," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 2, pp. 1352–1357, 2020.
- [7] M. Li, C. Wang, and Y. Sun, "A dual-clustering based collaborative filtering recommendation algorithm with confidence estimation of user," *Journal of Computational Information Systems*, vol. 11, no. 12, pp. 4245–4254, 2015.
- [8] N. Tohidi and C. Dadkhah, "Improving the performance of video collaborative filtering recommender systems using optimization algorithm," *The International Journal of Non-linear Analysis and Applications (IJNAA)*, vol. 11, no. 1, pp. 283–295, 2020.
- [9] J. Shi, Y. Tang, and P. Xu, "A collaborative filtering recommendation method based on clustering and user trust," *Journal of Computational Information Systems*, vol. 11, no. 18, pp. 6845–6852, 2015.
- [10] W. Froelich and P. Hajek, "Combining rough set-based relevance and redundancy for the ranking and selection of nominal features," *Procedia Computer Science*, vol. 176, pp. 1459–1468, 2020.
- [11] J.-H. Su, B.-W. Wang, C.-Y. Hsiao, and V. S. Tseng, "Personalized rough-set-based recommendation by integrating multiple contents and collaborative information," *Information Sciences*, vol. 180, no. 1, pp. 113–131, 2010.

- [12] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, "A collaborative filtering approach to mitigate the new user cold start problem," *Knowledge-Based Systems*, vol. 26, pp. 225–238, 2012.
- [13] B. S. Neysiani, N. Soltani, N. Soltani, R. Mofidi, and M. H. Nadimi-Shahraki, "Improve performance of association rule-based collaborative filtering recommendation systems using genetic algorithm," *International Journal of Information Technology and Computer Science*, vol. 11, no. 2, pp. 48–55, 2019.
- [14] I. Kim and V. Gupta, "Using social tags and user rating patterns for collaborative filtering," *International Journal of Information Systems and Social Change*, vol. 8, no. 2, pp. 25–49, 2017.
- [15] N. Yin, "A big data analysis method based on modified collaborative filtering recommendation algorithms," *Open Physics*, vol. 17, no. 1, pp. 966–974, 2019.
- [16] A. Sang, "Design and implementation of collaborative filtering approach for movie recommendation system," *International Journal of Computer Applications*, vol. 167, no. 12, pp. 18–24, 2017.
- [17] S. Hattori, R. Misawa, H. Ishikawa, and Y. Takama, "Hybrid recommender system based on collaborative filtering employing personal values-based user model," *Journal of Japan Society for Fuzzy Theory and Intelligent Informatics*, vol. 29, no. 4, pp. 628–636, 2017.
- [18] S. K. Raghuvanshi, R. K. Pateriya, f. n. m. au, and R. K. Pateriya, "Movie recommendation system content-based and collaborative filtering," *International Journal of Computer Sciences and Engineering*, vol. 6, no. 4, pp. 476–481, 2018.
- [19] N. Ifada, N. F. D. Putri, and M. K. Sophan, "Normalization based multi-criteria collaborative filtering approach for recommendation system," *Rekayasa*, vol. 13, no. 3, pp. 234–239, 2020.
- [20] S. L. Jadhav, M. P. Mali, and M. P. Mali, "Pre-recommendation clustering and review based approach for collaborative filtering based movie recommendation," *International Journal of Information Technology and Computer Science*, vol. 8, no. 7, pp. 72–80, 2016.
- [21] X. Liu, "An improved clustering-based collaborative filtering recommendation algorithm," *Cluster Computing*, vol. 20, no. 2, pp. 1281–1288, 2017.
- [22] L. Chao, K. Zhang, Z. Li, Y. Zhu, J. Wang, and Z. Yu, "Geographically weighted regression based methods for merging satellite and gauge precipitation," *Journal of Hydrology*, vol. 558, pp. 275–289, 2018.
- [23] H. Xu, C. Zhang, H. Li, and J. Ou, "Real-time hybrid simulation approach for performance validation of structural active control systems: a linear motor actuator based active mass driver case study," *Structural Control and Health Monitoring*, vol. 21, no. 4, pp. 574–589, 2014.
- [24] W. Wang, Z. Gong, and J. Ren, "Venue topic model-enhanced joint graph modelling for citation recommendation in scholarly big data," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, vol. 20, no. 1, pp. 1–15, 2020.
- [25] J. Yang, M. Xi, and B. Jiang, "FADN: fully connected attitude detection network based on industrial video," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 2011–2020, 2020.
- [26] K. Sim, J. Yang, W. Lu, and X. Gao, "Blind stereoscopic image quality evaluator based on binocular semantic and quality channels," *IEEE Transactions on Multimedia*, vol. 17, p. 1, 2021.
- [27] Y. Li, R. Liang, W. Wei, W. Wang, J. Zhou, and X. Li, "Temporal pyramid network with spatial-temporal attention for pedestrian trajectory prediction," *IEEE Transactions on Network Science and Engineering*, vol. 1, 2021.