Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 7191410, 11 pages
https://doi.org/10.1155/2022/7191410

Research Article

Research on Product Design Strategy Based on User Preference and Machine Learning Intelligent Recommendation

Jie Wu

Design Department, Taiyuan Normal University, Taiyuan 030619, China

Correspondence should be addressed to Jie Wu; wujie@tynu.edu.cn

Received 15 February 2022; Revised 18 March 2022; Accepted 9 April 2022; Published 28 April 2022

Academic Editor: Lisheng Fan

In the machine learning model, intelligent recommendation system can select valuable information from a lot of data to help users find the products or services they need, which has been more and more widely used in recent years. However, there are still many problems in machine learning recommender systems, such as data sparsity, natural noise, and cold start, which leads to the fact that machine learning recommender systems cannot obtain accurate user preferences. When a project is rated, the quality of the recommendation is greatly affected. In order to solve the problem that the existing recommendation algorithms have poor recommendation results in sparse data sets, this paper proposes a machine learning method for recommendation rating prediction based on user interest concept lattice. Firstly, the nearest neighbors are divided into direct nearest neighbors and indirect nearest neighbors by user interest concept lattice. Then, different methods are used to calculate the similarity between the direct "nearest neighbor" and the target user, and the similarity between the indirect "nearest neighbor" and the target user. Finally, the invisible item score of the target user is calculated by the similarity value. Experiments are carried out on real data sets, and the experimental results show that the CFCNN-CL algorithm and RRP-UI CL algorithm proposed in this paper have high recommendation accuracy and still have good performance in the case of sparse data.

1. Introduction

With the development of the Internet, users can access it through various devices and services. Users are more involved in the project selection process by directly controlling the items to be accessed (such as film and television dramas, music, clothing, websites, travel, accommodation, e-learning materials, gadgets, and applications), and there are many different items to choose from around each user. With the increase of information and data scale in the Internet, it is difficult for users to find interesting projects in a reasonable time, and the project selection process may become tedious and complicated [1]. In order to prevent users from choosing items among tens of millions of items and recommending items to people according to their preferences, recommender system for machine learning is introduced [2]. The recommendation system tracks the interaction information between users and their selected items and then uses this information to process into a user model through recommendation algorithm, which is used to filter out the items that users are interested in and recommend the results to users in the form of personalized list [3]. According to user's needs, interests, etc., create a list of items that users are interested in, without a lot of interaction with users [4]. Recommendation system helps users to solve the problem of too many products and difficult to choose and provides them with personalized services. Users can make appropriate purchase decisions and explore new products from the best product evaluation through the minimum online search cost. Now, recommendation system has been fully mined, it has appeared in any services that require users to make decisions, including e-commerce, information retrieval, navigation information services, social networks, and other fields [5].

The two most commonly used technologies in the development of recommendation system are content-based technology and collaborative filtering technology. Among them, content-based technology extracts the features of items first and then can provide items with similar features selected by users in the past [6]. The technology based on collaborative filtering mainly relies on the historical records provided by users to predict the items they are interested in and mainly depends on the scoring data, which is easy to implement and has high recommendation accuracy [7]. Collaborative filtering has become the most popular recommendation algorithm at present [8]. It uses user scores to build user-user or item-item similarity index and identifies the "nearest neighbor" of users or items to generate recommendations. Collaborative filtering mainly includes neighborhood-based and model-based methods, both of which have their own advantages and disadvantages. Neighborhood-based recommendation has high accuracy, but if new users join, it will reduce performance. The model-based model has better scalability and makes up for the shortcomings of the neighborhood-based model, but the recommendation accuracy is low [9]. Compared with the traditional recommendation method, this paper adopts the nearest neighbor similarity comparison method. Firstly, the nearest neighbor is divided into direct nearest neighbor and indirect nearest neighbor by user interest concept lattice. Then, different methods are used to calculate the similarity between the direct "nearest neighbor" and the target user, and the similarity between the indirect "nearest neighbor" and the target user. Finally, the invisible item score of the target user is calculated by the similarity value. On the basis of direct nearest neighbor, an indirect nearest neighbor similarity comparison method is proposed to further obtain the optimal recommended value. Compared with traditional methods, the recommended methods in this paper are better in integrity.

2. Recommendation System Theory

The purpose of researching recommendation system is to retrieve the most relevant products and services from a large amount of data, so as to reduce information overload and provide personalized services [10]. In 1990s, recommendation system was first applied to e-commerce and Web services. In recent years, people have developed various recommendation system software for social networks, digital libraries, e-commerce, and online advertising [11]. This section mainly summarizes the commonly used recommendation algorithms and common problems in the current recommendation system.

2.1. Overview of Intelligent Recommendation Algorithms. Recommendation system can be defined as a program, which predicts users' interest in projects based on projects, users, and interaction information between projects and users, so as to recommend the most suitable projects (products or services) to specific users (target users). In recommendation system, the quality of recommendation has a great relationship with the performance of recommendation algorithm. The following will introduce the common intelligent recommendation algorithms [12, 13].

2.1.1. Collaborative Filtering (CF) Recommendation Algorithm. CF is to recommend target users by analyzing the scoring information of other users or other items, and

Table 1: User-item scoring matrix.

	I_1	I_2	I_3	I_4	I_5
U_1	3	0	0	0	1
U_2	0	4	0	5	0
U_3	0	2	4	0	0

the recommendation accuracy is higher. Two main recommendation algorithms will be introduced below.

(1) ***CF*** *Based on Neighborhood.* In the neighborhood-based collaborative filtering recommendation algorithm, finding similar users is a key step, and the main goal of similar users is to get the most suitable recommendation items for the target users. The user-based algorithm is mainly divided into three steps: first, calculating similarity; the second is to choose the "nearest neighbor" according to the similarity; the third is to calculate the score value and make prediction and recommendation. Next, we will introduce the most used methods to calculate similarity.

Adjusted Cosine (ACOS) similarity in user u and v is calculated using Equation (1).

$$\mathrm{sim}(u,v)=\frac{\sum_{i\in I_u\cap I_v}(r_{u,i}-\overline{r_u})(r_{v,i}-\overline{r_v})}{\sqrt{\sum_{i\in I_u}(r_{u,i}-\overline{r_u})^2}\sqrt{\sum_{i\in I_v}(r_{v,i}-\overline{r_v})^2}}. \quad (1)$$

Pearson's Correlation (PC) is used to calculate the similarity between u and v.

$$\mathrm{sim}(u,v)=\frac{\sum_{i\in I_u\cap I_v}(r_{u,i}-\overline{r_u})(r_{v,i}-\overline{r_v})}{\sqrt{\sum_{i\in I_u\cap I_v}(r_u,i-\overline{r_u})^2}\sqrt{\sum_{i\in I_u\cap I_v}(r_{vi}-\overline{r_v})^2}}. \quad (2)$$

Constrained Pearson's Correlation (CPC) is using Equation (3)to calculate the similarity between u and v.

$$\mathrm{sim}(u,v)=\frac{\sum_{i\in I_u\cap I_v}(r_{u,i}-r_{\mathrm{med}})(r_{v,i}-r_{\mathrm{med}})}{\sqrt{\sum_{i\in I_u\cap I_v}(r_{u,i}-r_{\mathrm{med}})^2}\sqrt{\sum_{i\in I_u\cap I_v}(r_{v,i}-r_{\mathrm{med}})^2}}, \quad (3)$$

where r_{med} is the median of the grade.

The Jaccard similarity between u and v is calculated by using Equation (4).

$$\mathrm{sim}(u,v)=\frac{|I_u\cap I_v|}{|I_u\cup I_v|}, \quad (4)$$

where $|I_u\cap I_v|$ is the same number evaluated by u and v together.

$$\mathrm{R_{u,i}}=\overline{r_u}+\frac{\sum_{v\in N_u} sim(u,v)(r_{v,i}-\overline{r_v})}{\sum_{v\in N_u} sim(u,v)}, \quad (5)$$

where $\mathrm{sim}(u,v)$ is the similarity of user u and v, and n_u is the

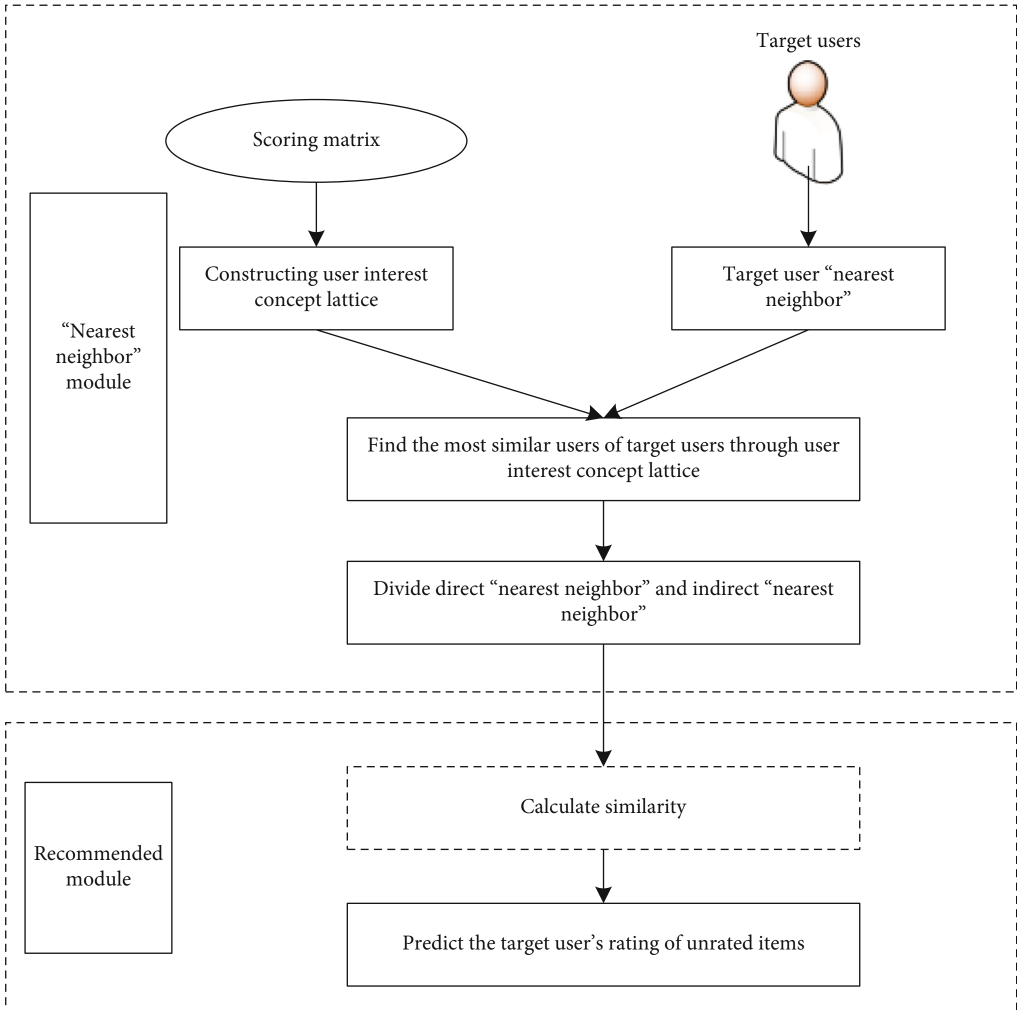

Figure 1: RRP-UICL algorithm model.

Table 2: Scoring matrix.

	I_1	I_2	I_3	I_4	I_5	I_6	I_7
U_1	0	0	0	5	0	4	1
U_2	0	4	4	5	2	0	4
U_3	0	4	4	0	5	1	2
U_4	1	2	5	4	0	3	4
U_5	4	1	0	5	0	5	0
U_6	5	3	4	5	2	0	4

"nearest neighbor" set of user u. Project-based algorithm and user-based algorithm have the same calculation principle, but the calculation objects are different, and users need to be exchanged for projects.

(2) Collaborative Filtering Based on Model. The model-based algorithm is mainly divided into two main stages. In the first stage, we need to deal with the original scoring matrix and construct an effective model representing the original matrix. In the second stage, we use the generated model as an input matrix to predict the scoring of target users. The core of this algorithm is the establishment of model, which needs to use historical information to create and generate recommended models, among which singular value decomposition (SVD) is the most widely used model [14].

In the SVD model, the original scoring matrix R is decomposed into three matrices, and the decomposition form is

$$R_K = USV^T, \tag{6}$$

where u and v are two the orthogonal matrices, s is a diagonal matrix of size $r^* r$, and r is the rank of matrix R, which is composed of singular values of scoring matrix. The matrix can be reduced by discarding the minimum value, and finally, the matrix s is obtained, where $k < r$; then, the decomposition form of the reconstructed matrix is

$$R_K = U_K S_K V_K^T. \tag{7}$$

Table 3: Background of user interest form.

	I_1	I_2	I_3	I_4	I_5	I_6	I_7
U_1	0	0	0	1	0	1	0
U_2	0	1	1		0	0	1
U_3	0	1	1	0	1	0	0
U_4	0	0	1	1	0	0	1
U_5	1	0	0	1	0	1	0
U_6	1	0	1	1	0	0	1

The scoring prediction formula is

$$\mathrm{R}_{\mathrm{u,i}} = \overline{r_u} + \mathrm{U_K}\sqrt{\mathrm{S}_{\mathrm{k}}^{\mathrm{T}}(u)}\sqrt{\mathrm{S_K}}\mathrm{V}_{\mathrm{k}}^{\mathrm{T}}(i). \tag{8}$$

The recommendation algorithm based on collaborative filtering does not need detailed content. When the details of content cannot be accessed or it is difficult to collect or analyze the details, collaborative filtering method is very effective, and this method can find the items that target users want in a large number of items [15]. However, it will also face the problem of rating sparsity, and there will also be the problem of cold start of new users and new projects.

2.1.2. Content-Based. Content-based is the earliest recommendation algorithm, by comparing the characteristic information contained in the project with the characteristic information interested by the target user, the project is recommended to the target user, and the foundation is to find similar items by the target users before. For providing appropriate recommendations to the target users, accurate user characteristics, preferences, and demand models are needed. Firstly, the system extracts the feature information contained in each item, then classifies the items used by the target users before, extracts the feature information of the items, and then learns the feature information to obtain the user's preference characteristics. Finally, compare the user's preference characteristics with the feature information contained in the items and recommend the users through the correlation.

At present, *TF-IDF* is the most commonly used computational method in information retrieval, which is used to develop vector space model in content-based recommendation algorithm. In this method, the project content is regarded as a document D, and then, the keyword T is extracted from it, and the calculation formula of the *TF* value of the keyword T in the document D is shown in (9).

$$\mathrm{TF}_{\mathrm{t.d}} = \frac{N_{t.d}}{\sum_k N_{k.d}}, \tag{9}$$

where $N_{t,d}$ represents the number of times the keyword t appears in the document d, and the calculation formula of *IDF* value corresponding to the keyword is shown in Equation (10).

$$\mathrm{IDF_t} = \log \frac{|D|}{1 + |d \in D : t \in d|}, \tag{10}$$

where D denotes the set of documents, and $1 + |d \in D : t \in d|$ denotes the number of keywords t contained in document d.

The Rocchio algorithm is usually used to deal with the relevance feedback in the process of information retrieval and extract the interesting feature information of the target users. Decision tree algorithm, linear classification algorithm, and Naive Bayes method are used to classify documents, and documents are interested or uninterested.

Content-based recommendation algorithm does not have the problem of data sparsity, and new items can be recommended immediately. However, because the algorithm needs to extract the feature information of the project, At the same time, the algorithm only relies on the behavior information of the target users to recommend and does not involve the behavior information of other users. There are many problems in diversity. When new users enter the recommendation system, they also face the cold start problem when selecting items.

2.1.3. Hybrid. Hybrid combines two or more recommendation algorithms to predict and recommend and improve the recommendation accuracy.

In the recommendation algorithm based on the combination of content and collaborative filtering, the prediction value based on content algorithm can be used to supplement the user's historical scoring data, adding data to form a pseudoscore matrix, in which the observed scores remain unchanged, and then, using collaborative filtering algorithm based on weighted Pearson's correlation to predict the pseudoscore matrix, the recommendation algorithm has better prediction performance and also overcomes the cold start problem and data sparsity problem.

2.2. Frequently Asked Questions on Recommendation Systems. There are some problems in the current recommendation system. At present, the common problems in the recommendation system are as follows:

2.2.1. Data Sparsity Problem. Data sparse refers to the lack of useful scoring data when recommending items to target users, which leads to the error between recommended items and users' needs.

In most recommendation systems, each user only evaluates a part of the available items, so most of the evaluation information is empty. When users only grade a few items, there will be great errors in the similarity between different users or items, and at this time, the recommendation quality of recommendation algorithm will be greatly affected.

Sparsity is related to the scoring data hidden in the recommendation system, the number of scores can be measured by sparsity, which indicates the ratio of the number of unscored data to the whole matrix space in a scoring matrix. Assuming that a scoring matrix has U users, I items and R scores in total, and S is used to represent the sparsity

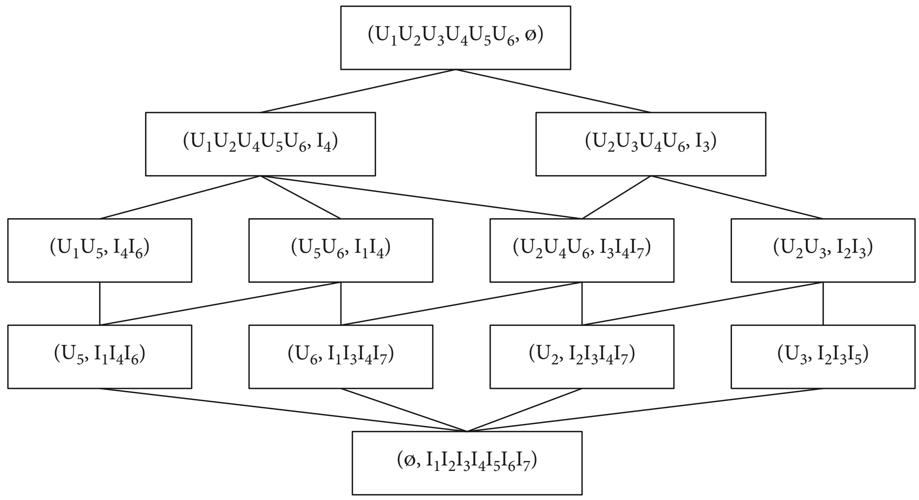

Figure 2: User interest concept lattice.

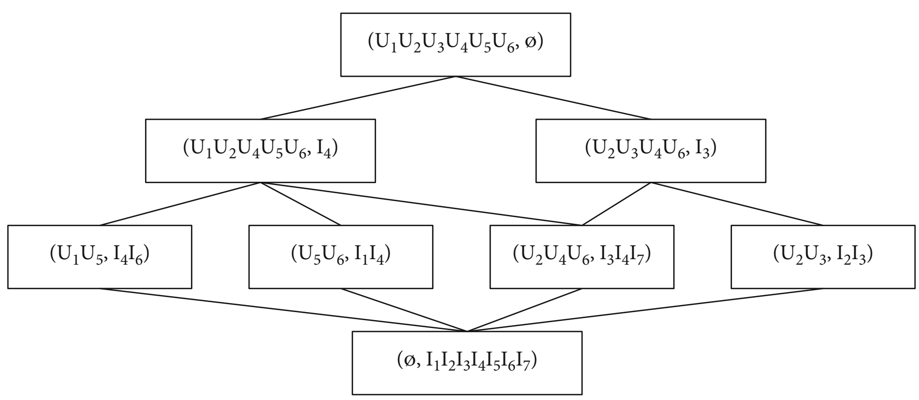

Figure 3: Final user interest concept lattice.

of the data set, the calculation formula of sparsity S is

$$S = \frac{R}{\mathrm{UI}}. \tag{11}$$

Generally, neighborhood-based collaborative filtering algorithms use similarity to find users similar to recommended users or items similar to candidate items. The similarity between items is also calculated using the scores provided by users. However, if there are few or no common scoring items in the given scoring data, these methods become inapplicable.

2.2.2. Noise Issues. Noise in recommendation system refers to the data that will affect the score prediction in the data set. Noise in recommendation system data set can be divided into malicious noise and nonmalicious noise (natural noise), both of which are very important and will have adverse effects on recommendation performance.

Malicious noise refers to the behavior that some biased data are intentionally added to the system, which is intentionally introduced by external agents and intentionally deviates the output of the system in a specific way, which has a great impact on the recommendation performance. Foreign agents will maliciously attack the recommendation system in order to have significant advantage in the recommendation system. Because many recommendation systems run in the business environment, some people will use the recommendation system to seize the advantage in the business competition. For example, if authors hope to promote their work by exporting artificially high reviews for their publications through the recommendation system, and at the same time reduce the recommendations for other similar works, they will find some people to improve the false scores, resulting in biased recommendation results.

Natural noise is the output data of users' real evaluation, which is produced by users' activity errors. This kind of noise is related to the method of collecting or inferring users' preferences in recommendation system. Because all human activities are error-prone, and the user's preference output is usually a heavy process, some errors will naturally appear in the data. In the data set noise of recommendation system, this paper mainly studies the natural noise.

2.2.3. Cold Start Problems. In the recommendation system, the cold start user problem refers to the fact that the system

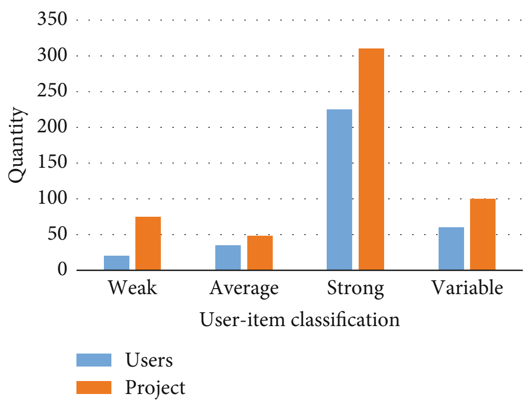

Figure 4: Sample data set user-item classification.

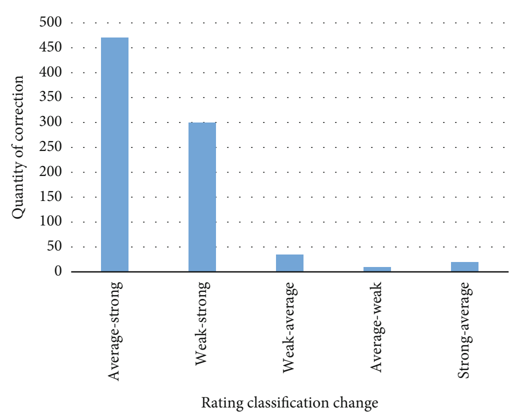

Figure 5: Number of revised ratings in sample data set.

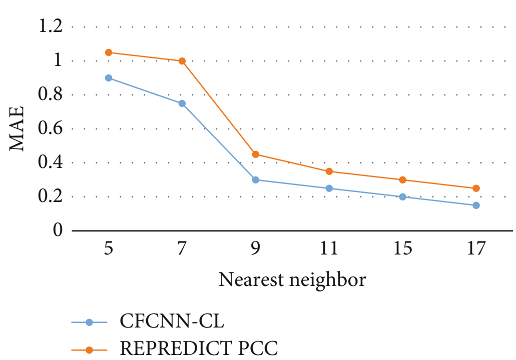

Figure 6: MAE comparison of different nearest neighbors.

cannot recommend related items for the user when the user is a new user, because there is a lack of item scoring history information to help determine the user's interest. Similarly, an item can only be recommended after a large number of users have rated it. For an item that has never been evaluated by users, the system usually cannot make high-quality suggestions. This problem is called cold start project problem.

The cold start problem is caused by the lack of user data and project scoring history. Cold start problems can be mitigated by adding information about user items, and valuable data can be provided to determine users' interest in items by identifying trust relationships between users and the influence of one user on another, which is very useful for making suggestions to users more accurately and objectively.

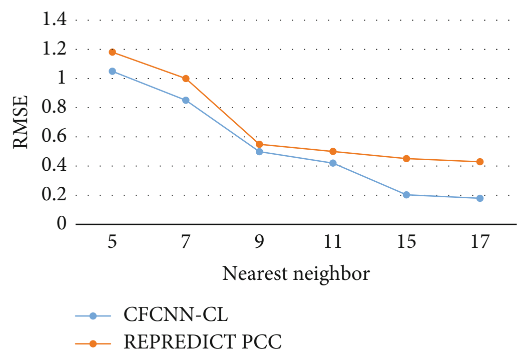

Figure 7: RMSE comparison of different nearest neighbors.

2.2.4. Scalability Issues. As the number of users and projects gradually increases, scalability problems arise. The recommendation system not only needs to deal with the interaction between the original users and projects but also needs to respond to the interaction information between new users and projects. Therefore, the recommendation system needs to deal with a large amount of data, which requires powerful computing power to execute and quick response to the needs of online users. In the recommendation system, the scalability of the system also needs to be considered. A recommendation system with good scalability can quickly deal with the needs of a large number of users and recommend accurate items.

3. Recommendation Score Prediction Algorithm Based on User Interest Concept Lattice

3.1. Problem Description and Analysis. Recommendation system mainly depends on the information left by users after browsing. Among this information, the explicit feedback information between users and items is very important. Among them, the user's rating data is the most commonly used explicit feedback information. The higher the user's rating on an item, the more the user likes and interests the item.

In the recommendation system, collaborative filtering algorithm can achieve good results when there are more score data. However, because some users have no habit of scoring after using the project, they cannot give the system a clear feedback on their love for the project, and the scoring data in most system databases will become very few, which leads to the recommendation system cannot recommend satisfactory projects to the target users well. Among them, the "nearest neighbor" selection is based on the assumption that if two users have similar scores for common items, they can be regarded as having similar preferences, and the services received by one user may be recommended to another user. In the implementation of the algorithm, the most commonly used similarity calculation method is Pearson's Correlation (PC) coefficient, and the similarity between the

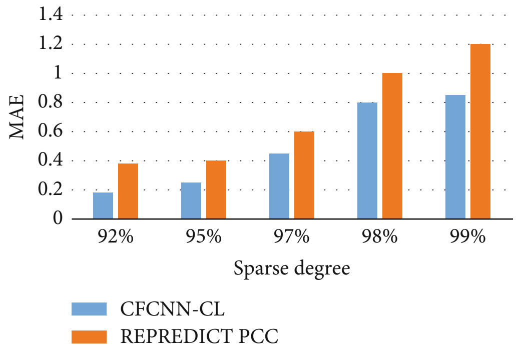

Figure 8: MAE comparison with different sparsity.

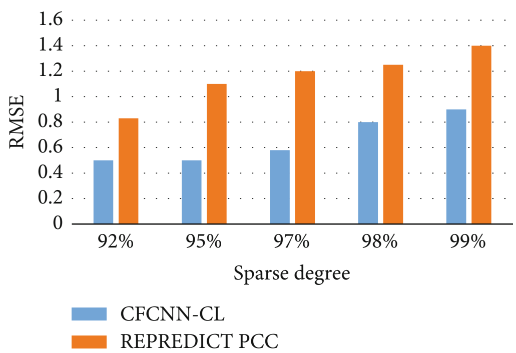

Figure 9: Comparison of RMSE with different sparsity.

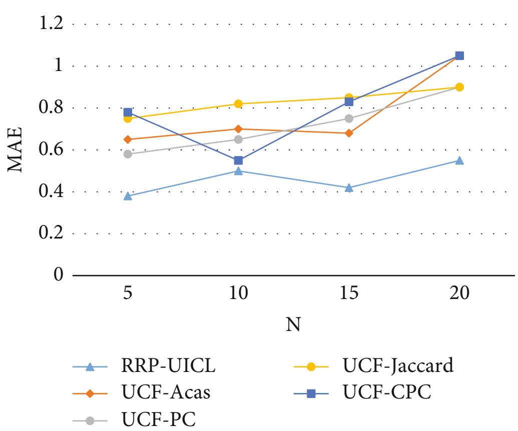

Figure 10: Comparison of MAE values of different methods in data set-1.

target user u and the neighbor user v is calculated by formula (12).

$$\operatorname{sim}(u, v)=\frac{\sum_{i \in I_u \cap I_v}\left(r_{u,i}-\overline{r_u}\right)\left(r_{v,i}-\overline{r_v}\right)}{\sqrt{\sum_{i \in I_u \cap I_v}\left(r_{u,i}-\overline{r_u}\right)^2} \sqrt{\sum_{i \in I_u \cap I_v}\left(r_{v,i}-\overline{r_v}\right)^2}}. \quad (12)$$

It can be seen from the expression of calculating similarity that the calculation of similarity mainly depends on $I_u \cap I_v$, that is, the common item set of target user u and neighbor user v. However, in the actual user-item scoring data set, the scoring data is very few, and the common item set that can be provided will be very few, which will affect the calculation of similarity. After the similarity degree is calculated, the

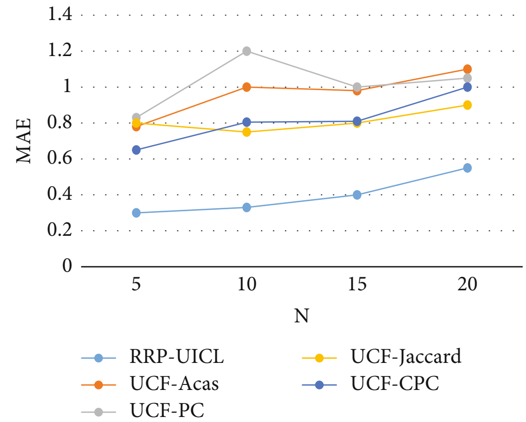

Figure 11: Data set-2 comparison of MAE values for different methods.

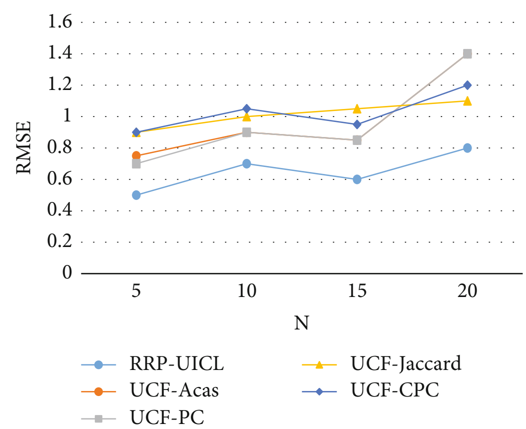

Figure 12: Comparison of RMSE values of different methods in data set-1.

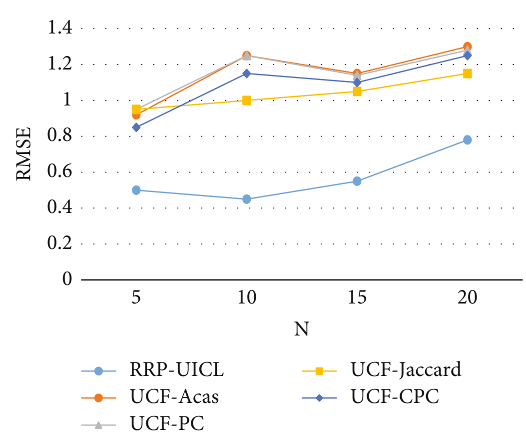

Figure 13: Data set-2 comparison of RMSE values for different methods.

Table 4: Data set-1 comparison of RMSE values of different methods before and after noise correction.

Method	N	RRP-UICL	UCF-ACos	UCF-PC	UCF-CPC	UCF-jaccard
Noise data set-1	5	0.5496	0.7835	0.7156	0.9378	0.9156
	10	0.7689	0.9845	0.9478	1.0856	1.0707
	15	0.6914	0.9989	1.0002	1.0818	1.1403
	20	0.8756	1.4956	1.4956	1.2945	1.2315
CFCNN-CL correction Noise data set-1	5	0.5471	0.7756	0.7056	0.9316	0.9118
	10	0.7051	0.9646	0.9289	1.0256	1.0239
	15	0.6845	0.9695	0.9565	1.0789	1.0956
	20	0.8535	1.2813	1.4209	1.2656	1.1489

Table 5: Data set-2 comparison of RMSE values of different methods before and after noise correction.

Method	N	RRP-UICL	UCF-ACos	UCF-PC	UCF-CPC	UCF-jaccard
Noise data set-2	5	0.4530	0.9456	1.0045	0.8812	0.9817
	10	0.4756	1.3617	1.3727	1.2233	1.0589
	15	0.5515	1.2902	1.2986	1.1945	1.0666
	20	0.8956	1.4795	1.4789	1.4569	1.3303
CFCNN-CL correction noise data set-2	5	0.4389	0.8964	1.0012	0.8759	0.9002
	10	0.4739	1.1807	1.2554	1.1854	1.0424
	15	0.4995	1.1088	1.1422	1.0867	0.9653
	20	0.8035	1.3256	1.2597	1.4068	1.3197

score value of the target user for the unscored items can be calculated, and the score value can be predicted by using Equation (13).

$$\mathrm{R}_{\mathrm{u,i}} = \overline{r_u} + \frac{\sum_{v\in N_u} sim(u,v)\left(r_{v,i} - \overline{r_v}\right)}{\sum_{v\in N_u} sim(u,v)}. \tag{13}$$

It can be seen from the expression of the predicted score value that the predicted score value mainly depends on the neighbor user set N_u and the similarity sim (u, v) of the target user u. The small neighbor user set and the large similarity error will affect the prediction of the score value and lead to the decrease of the recommendation accuracy.

For example, Table 1 is a simple user-item scoring matrix. It can be seen from the table that in the whole scoring matrix, only users U_2 and U_3 have a common scoring item I_2, and there are no common scoring items of U_1 and U_2, U_1 and U_3. At this time, when the similarity is calculated using the correlation-based method, since there is no common scoring item of U_1 and U_2, U_1 and U_3, the similarity of U_1 and U_2, U_1 and U_3 cannot be calculated. On the other hand, except when recommending item I_2 to user U_1, the target user has two neighbor users U_2 and U_3, and in other cases, there is only one neighbor user, and the collection of neighbor users is very small, which will affect the prediction of score value and lead to the degradation of recommendation quality of recommendation system.

3.2. Overview of Algorithm Model. In this paper, the data structure of user interest concept lattice is introduced, and a recommendation score prediction algorithm based on user interest concept lattice is proposed. The main steps of RRP-UICL algorithm proposed are shown in Figure 1.

As can be seen from Figure 1, the proposed method includes two main stages: one is the "nearest neighbor" module, and the other is the recommendation module.

In the recommendation module, when recommending to the target user, the similarity between the target user and the "nearest neighbor" should be calculated first. In this paper, different methods are used to calculate the similarity between the target user and the direct "nearest neighbor" and the indirect "nearest neighbor", and then, based on the similarity, the weighted average method is used to predict the scoring value of the target user for the unscored items. The algorithm will be described in detail below.

3.3. Constructing User Interest Concept Lattice. The binary matrix must be represented by a list of items of interest to each user, and in the rating matrix, the value of the item with the higher rating is set to <1>, and the values of all other items are set to <0>. In the reference scale, the items of score 4 and 5 are the items that users are interested in, and their values are set as <L>, while the values of other items are set as <0 >.

After the scoring matrix is converted into a binary matrix, if a user marks an item as L, it means that the user has the attribute of an item. The binary matrix can be regarded as a user interest formal background $K = (U, I, R)$, where U is the set of all users, which is equivalent to the set of objects, I is the set of all items, which can be regarded as the attribute set, and R is a relationship between U and 1. After obtaining the formal background of user interest, the concept lattice structure model is established according to the binary relationship between objects and attributes (users and items) in the formal background of user interest K, and the concept lattice of K is represented by L. After constructing the concept lattice, the recommendation algorithm can be analyzed based on the concept lattice theory, and the concept lattice theory can be applied to the recommendation algorithm.

Table 2 shows the scoring matrix of 6 users for 7 items, in this scoring matrix, according to the conversion principle of binary matrix, the items with scores of 4 and 5 are defined as items of interest to users, their values are set to <1 >, and other values are set to <0 >. The results are shown in Table 3, and the binary matrix can be regarded as a background of user interest form $K = (U, I, R)$, where user set $U = \{U_1, U_2, U_3, U_4, U_5, U_6\}$ and item set $I = \{I_1, I_2, I_3, I_4, I_5, I_6, I_7\}$. The user interest concept lattice constructed based on the user interest formal background in Table 3 is shown in Figure 2.

Because the traversal time is very complex, it needs to speed up the recommendation. It is necessary to delete some redundant L of user interest. This paper defines two conditions to delete formal concepts:

For a formal concept Z in the user interest concept lattice L_k,

(1) Z can be deleted if $\exists Z \in L_K$ is such that $|Ext(Z)| = 1$

(2) Z can be deleted if $\forall Z \in L_K$ is so that $Int(Z) \in Int(Z')$

According to the deletion condition of redundant formal concepts, the user interest concept lattice in Figure 2 is deleted, and the obtained end user interest concept lattice L_k is shown in Figure 3.

3.4. Partitioning "Nearest Neighbor." In this stage, the existing methods are mainly used to find the most similar users, and the "nearest neighbors" are divided by the most similar users.

The immediate "nearest neighbor" N_u^d of the target user u is represented as follows:

$$N_u^d = \{x | x \in N_u \text{ 且 } x \in MN_u\}. \tag{14}$$

Similarly, the indirect "nearest neighbor" N_u^{id} of the target user u is represented as follows:

$$N_u^{id} = \{x | x \in N_u \text{ 且 } x \notin N_u \cap MN_u\}. \tag{15}$$

Among the other "nearest neighbors" users, these users are just similar to the target users but do not show great interest in the recommended items. This paper classifies these users as indirect "nearest neighbors". For example, if a "nearest neighbor" Nu of a target user u is $\{U_1, U_2, U_3, U_4, U_5, U_6\}$, and the most similar user MNu is $\{U_3, U_4, U_6\}$. According to the above definition, the direct "nearest neighbor" N_u^d is $\{U_2, U_4, U_6\}$, and the indirect "nearest neighbor" N_u^{id} is $\{U_1, U_5\}$.

3.5. User Interest Forecast. According to the obtained direct "nearest neighbor" and indirect "nearest neighbor" of the target user, items can be recommended to the target user. There are two main methods of project recommendation: prediction method and list method. In the prediction method, in the list method, all items of interest to the "nearest neighbor" user are recommended to the target user.

3.5.1. Calculation of Correlation Coefficient between Users. For indirect "nearest neighbor" users, this paper uses Equation (16) to calculate the similarity between indirect "nearest neighbor" users and target users, and the similarity calculation formula between user U and user V is defined [16]:

$$\text{sim}(u, v) = \frac{\max(1, |I_u \cap I_v|) \sum_{i \in I_u} \sum_{j \in I_v} (r_{u,i}/r_{v,j})}{|I_u| \cdot |I_v| \cdot |I_u \cup I_v|}. \tag{16}$$

Weighted average forecast unscored items.

The prediction of score value is the last important step in the recommendation algorithm. Use the weight to obtain the final prediction score of each item. The steps of the prediction method are as follows:

First, you need to calculate the average score of recommended items. For recommended item I, use Equation (17) to calculate the average score.

$$\overline{r_i} = \frac{\sum_{v \in N_u^d \cup N_u^{id}} r_{v,i}}{|N_u^d \bigcup N_u^{id}|}. \tag{17}$$

In the scoring matrix of Figure 2, it is necessary to recommend the item I_4 to the target user U_3. It can be obtained that the "nearest neighbor" Nu of the target user U_3 is $\{U_1, U_2, U_3, U_4, U_5, U_6\}$, the most similar user MNu is $\{U_2, U_4, U_6\}$, the direct "nearest neighbor" N_u^d is $\{U_2, U_4, U_6\}$, and the indirect "nearest neighbor" N_u^{id} is $\{U_1, U_5\}$. According to the scores of the direct "nearest neighbor" and the indirect "nearest neighbor", the average score of the recommended item I_4 is calculated as $\overline{r_i} = (5 + 5 + 4 + 5 + 5)/5 = 4.8$.

Then, the score of the recommended item is predicted, and the score R_u, i of the target user u for the item i is predicted using formula (18)

$$R_{u,i} = \frac{\sum_{v \in N_u^d} r_{v,i} \left(a - |r_{v,i} - \overline{r_i}| - 1\right)^2 + \sum_{v \in N_u^{id}} r_{v,i} \text{sim}(u, v) \left(a - |r_{v,i} - \overline{r_i}| - 1\right)^2}{\sum_{v \in N_u^d} \left(a - |r_{v,i} - \overline{r_i}| - 1\right)^2 + \sum_{v \in N_u^{id}} \text{sim}(u, v) \left(a - |r_{v,i} - \overline{r_i}| - 1\right)^2}. \tag{18}$$

4. Experimental Design and Result Analysis

4.1. Experimental Design. In the experiment part, firstly, we validate the effectiveness of CFCNN-CL algorithm to solve the natural noise in the recommendation system, and then, we validate the effectiveness of RRP-UICL algorithm to solve the data sparse problem in the recommendation system. Finally, we combine CFCNN-CL algorithm and RRP-UICL algorithm to recommend, and validate the effectiveness through experiments. Three parts of the experiment are using the recommendation algorithm to evaluate the average absolute error and mean root error for comparative analysis.

4.2. Experimental Data Set. In this paper, the data set MOVIELENS 100 K is used to verify the effectiveness of the algorithm. MOVIELENS data set is one of the most commonly used data sets to evaluate the effectiveness of the recommendation algorithm. A score of 4 means that the user likes the movie, and a score of 5 means that the user likes the movie very much. In the whole data set, each user evaluates at least 20 scores after watching the movie.

4.3. Performance Evaluation Indicators. Average absolute error and mean root error are used to evaluate the accuracy of our method. Under normal circumstances, the smaller the MAE, the higher the prediction accuracy, and the calculation formula is

$$\mathrm{MAE}=\frac{\sum_{i=1}^{n}\left|r_i-p_i\right|}{n}. \tag{19}$$

The RMSE is calculated by dividing the sum of squares of the difference between the actual score value and the predicted score value by the score set in the test set. The calculation formula is

$$\mathrm{RMSE}=\sqrt{\frac{\sum_{i=1}^{n}\left(r_i-p_i\right)^2}{n}}. \tag{20}$$

4.4. Method Analysis

4.4.1. Performance Analysis of CFCNN-CL Algorithm. According to the algorithm, the users and items of the sample data set are classified, as shown in Figure 4.

See Figure 5 for the number of natural noise correction in sample data set by collaborative filtering method based on concept lattice.

For verifying the effectiveness of CFCNN-CL algorithm, CFCNN-CL algorithm and existing PCC reprediction methods are tested on data sets, and the effects of nearest neighbor and sparsity on MAE and RMSE values are compared.

(1) The Influence of Nearest Neighbor. With the change of the nearest neighbor number, the changes of MAE and RMSE values in the corresponding two methods are shown in Figures 6 and 7, respectively. By analyzing Figures 6 and 7, it can be concluded that the MAE and RMSE of CFCNN-CL algorithm and PCC reprediction method decrease with the increase of nearest neighbors, and the increase of nearest neighbors improves the accuracy of the two methods. However, the MAE and RMSE values of the proposed method are lower than those of PCC reprediction method, so the noise correction method proposed in this paper has better performance and better prediction accuracy than PCC reprediction method.

(2) The Effect of Sparsity. For verifying the effectiveness in sparse scenarios, the available ratings in sample data sets are randomly changed to zero to form five data sets with different sparseness, which are 92%, 95%, 97%, 98%, and 99%, respectively. Then, the CFCNN-CL algorithm and the existing PCC reprediction method are tested on five data sets with different sparseness. Finally, the MAE and RMSE values of different methods are compared, respectively.

Figures 8 and 9 show the experimental results of MAE and RMSE value changes in different sparse scenarios with the proposed method and PCC reprediction method. The natural noise correction method proposed in this paper is superior to PCC reprediction method.

4.4.2. Performance Analysis of RRP-UICL Algorithm. When the recommended items $N = 5$, 10, 15, and 20, for data set-1 and data set-2 with different sparsity, the experimental results of MAE of the five methods varying with the recommended items are shown in Figures 10 and 11.

As can be seen from Figures 10 and 11, of the five methods, the MAE values of the RRP-UICL method in both data sets are smaller than those of the other four methods. It can be seen from the analysis chart that in sparse scenes, this method has better prediction accuracy than the commonly used collaborative filtering methods.

Similarly, under different recommended items $N = 5$, 10, 15 and 20, the experimental results of RMSE values of five methods in data set-l and data set-2. RRP-UICL method has better prediction accuracy than commonly used collaborative filtering methods in Figures 12 and 13.

4.4.3. Recommended Performance Analysis of Fusion CFCNN-CL and RRP-UICL. In this section, CFCNN-CL algorithm to solve natural noise and RRP-UICL algorithm to solve data sparsity are recommended, and data set-1 and data set-2 with different sparsity in the previous section are used for experimental analysis. At the beginning of the experiment, CFCNN-CL algorithm is used to correct the natural noise in data set-1 and data set-2. Then, the recommended score prediction algorithm RRP-UICL and the four commonly used methods UCF-ACos, UCF-PC, UCF-CPC, and UCF-Jaccard are tested on data set-1 and data set-2 with corrected natural noise, respectively. Finally, the experimental results are compared with those on data set-1 and data set-2 without corrected natural noise, and the results are analyzed.

When the recommended item $N = 5$, 10, 15, and 20, the experimental results of RMSE values of the five methods varying with the recommended items are shown in Table 4 for the data set-1 without correcting the natural noise and the data set-1 with correcting the natural noise, and the experimental results of RMSE values of the five methods varying with the recommended items are shown in Table 5 for the data set-2 without correcting the natural noise and the data set-2 with correcting the natural noise.

The recommendation combining CFCNN-CL and RRP-UICL also has the smallest RMSE value and the highest recommendation accuracy in data set-1 and data set-2 in Table 4 and Table 5. For the comparison before and after noise correction, the RMSE value of the five methods after noise correction is lower than that before noise correction, and the performance has been improved accordingly. However, UCF-ACos, UCF-PC, UCF-CPC, and UCF-Jaccard are all affected by data sparsity to varying degrees, so the recommendation performance will decrease in sparse data, and with the increase of sparsity, the recommendation performance will become worse. The recommendation method based on CFCNN-CL and RRP-UICL avoids the influence of natural noise and data sparsity, and the recommendation accuracy is kept in good condition.

5. Conclusion

With the continuous development of recommendation system and the increasing demand of people, the performance and accuracy of recommendation algorithm are required to be higher and higher. Firstly, this paper analyzes the development status of recommendation system and concept lattice and explains the research background and significance of this paper. After that, the related theories of recommendation system and concept lattice are introduced, which provides theoretical support for the following methods.

The existing recommendation algorithms cannot recommend accurately due to the influence of sparse data, this paper proposes a recommendation rating prediction algorithm based on user interest concept lattice, considering the different influence degree of the "nearest neighbor" users of the target users in the rating prediction process. The prediction method proposed in this paper not only solves the problem of sparse data in recommendation system but also has high performance and prediction accuracy.

In the experimental part, the experimental settings are introduced firstly, and then, the effectiveness of CFCNN-CL algorithm and RRP-UICL algorithm and the fusion of CFCNN-CL and RRP-UICL recommendation are verified by using sample data sets. In the experimental results of CFCNN-CL algorithm, under the influence of nearest neighbor or sparsity, the noise correction method proposed in this paper has better performance and better prediction accuracy than PCC reprediction method. In the experimental results of RRP-UICL algorithm, in sparse scenarios, the proposed method has better performance and prediction accuracy than four commonly used methods: modified cosine similarity measure, Pearson's correlation measure, constrained Pearson's correlation measure, and Jaccard measure. In the final experimental results, under the influence of the number and sparsity of recommended items, the MAE value and RMSE value of CFCNN-CL and RRP-UICL recommendation method are the smallest, and the recommendation accuracy is the highest.

Data Availability

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

Conflicts of Interest

The author declares no conflicts of interest regarding this work.

References

[1] Y. Chen, "Mining of instant messaging data in the Internet of Things based on support vector machine," *Computer Communications*, vol. 154, pp. 278–287, 2020.

[2] A. Suresh and M. J. Carmel Mary Belinda, "Online product recommendation system using gated recurrent unit with Broyden Fletcher Goldfarb Shanno algorithm," *Evolutionary Intelligence*, 2021.

[3] Z. Ali, I. Ullah, A. Khan, A. Ullah Jan, and K. Muhammad, "An overview and evaluation of citation recommendation models," *Scientometrics*, vol. 126, no. 5, pp. 4083–4119, 2021.

[4] F. He and P. Wei, "Research on comprehensive point of interest (POI) recommendation based on spark," *Cluster Computing*, vol. 22, no. S4, pp. 9049–9057, 2019.

[5] A. Da'U, N. Salim, and R. Idris, "An adaptive deep learning method for item recommendation system," *Knowledge-Based Systems*, vol. 213, no. 8, article 106681, 2021.

[6] R. Gerbaudo, R. Gaspar, and R. G. Lins, "Novel online video model for learning information technology based on micro learning and multimedia micro content," *Education and Information Technologies*, vol. 26, no. 5, pp. 5637–5665, 2021.

[7] X. Han, Z. Wang, and H. J. Xu, "Time-weighted collaborative filtering algorithm based on improved mini batch K-means clustering," *Advances in Science and Technology*, vol. 105, pp. 309–317, 2021.

[8] Y. Gao and L. Ran, "Collaborative filtering recommendation algorithm for heterogeneous data mining in the Internet of Things," *IEEE Access*, vol. 7, pp. 123583–123591, 2019.

[9] Y. Wu, S. Zhao, and R. Guo, "A novel community answer matching approach based on phrase fusion heterogeneous information network," *Information Processing & Management*, vol. 58, no. 1, article 102408, 2021.

[10] B. Kaya, "A hotel recommendation system based on customer location: a link prediction approach," *Multimedia Tools and Applications*, vol. 79, no. 3-4, pp. 1745–1758, 2020.

[11] H. Zhang and F. Ye, *A personalized recommendation algorithm for user-preference similarity through the semantic analysis*, vol. 2, Springer International Publishing, 2016.

[12] X. Ye and D. Liu, "An interpretable sequential three-way recommendation based on collaborative topic regression," *Expert Systems with Applications*, vol. 168, no. 2, article 114454, 2021.

[13] Y. T. Song and S. Wu, "Slope one recommendation algorithm based on user clustering and scoring preferences," *Procedia Computer Science*, vol. 166, pp. 539–545, 2020.

[14] Y. Chen, "Research on personalized recommendation algorithm based on user preference in mobile e-commerce," *Information Systems and e-Business Management*, vol. 18, no. 4, pp. 837–850, 2020.

[15] Q. Zhou, F. Liao, C. Chen, and L. Ge, "Job recommendation algorithm for graduates based on personalized preference," *CCF Transactions on Pervasive Computing and Interaction*, vol. 1, no. 4, pp. 260–274, 2019.

[16] C. Zou, D. Zhang, J. Wan, M. M. Hassan, and J. Lloret, "Using concept lattice for personalized recommendation system design," *IEEE Systems Journal*, vol. 11, no. 1, pp. 305–314, 2017.