Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 4655030, 15 pages
https://doi.org/10.1155/2022/4655030

Research Article

Deep Collaborative Filtering: A Recommendation Method for Crowdfunding Project Based on the Integration of Deep Neural Network and Collaborative Filtering

Pei Yin, Jing Wang, Jun Zhao, Huan Wang, and Hongcheng Gan

Business School, University of Shanghai for Science and Technology, 516 Jungong Road, Shanghai 200093, China

Correspondence should be addressed to Pei Yin; pyin@usst.edu.cn

Received 13 April 2022; Revised 1 June 2022; Accepted 29 June 2022; Published 21 July 2022

Academic Editor: Wei Liu

In real recommendation systems, implicit feedback data is more common and easier to obtain, and recommendation algorithms based on such data will be more applicable. However, implicit feedback data cannot directly express user preferences. Meanwhile, data sparsity caused by massive data is still an urgent problem to be solved in recommendation system. In response to this phenomenon, this paper proposes a deep collaborative filtering algorithm. In the perspective of implicit feedback, this method uses the advantages of convolutional neural network for effective learning of the nonlinear interaction of users and items and the characteristics of collaborative filtering algorithm for modeling the linear interaction of users and items and combines the two methods for recommendation. Finally, the baseline method is set up and the comparative experiment and parameter adjustment is carried out. The experimental results show that the proposed algorithm has significantly improved the recommendation accuracy on public dataset (Yahoo! Movie). The parameter adjustment results show that, under the condition of uniformly collecting negative feedback data and setting a certain number of convolution layers, the sparser the data is, the better the recommendation performs. As a result, this paper has made some progress in solving the problem of data sparsity and enriching the research of recommendation system.

1. Introduction

Crowdfunding is a new financing method that emerged with the rise of the Internet [1]. So far, Kickstarter1, the world's largest crowdfunding platform, has successfully raised $4,384,962,222 for 165,564 projects and the famous European crowdfunding platform Ulule2 has more than 2.6 million users and has successfully funded 28,294 projects, with a total financing amount of €143,381,350. However, Kickstarter's funding success rate is only 37%, and Ulule's is only 65%. Project creativity, project advertising exposure, and project duration may affect the success rate of project financing [2]. However, the biggest reason is that there are so many projects on the platform that it is difficult for investors to browse the projects that they are really interested in. Therefore, crowdfunding project recommendation, matching investors' preferences, is conducive to improving the platform's financing success rate.

The survey shows that the sparsity of user behavior on most crowdfunding platforms is higher than 99%. Too sparse data makes common collaborative filtering algorithms invalid, such as user-based collaborative filtering. Due to the lack of a large amount of data, similarity between users cannot be calculated effectively. In addition, the public datasets commonly used in recommender system research are user ratings, such as Movie-lens, Netflix Prize, etc. These datasets can directly judge the level of interest of users according the score, so as to determine positive and negative feedback data. However, the user investment behavior data of crowdfunding websites does not have a clear score, which belongs to the implicit feedback data.

For the sake of reducing the impact of sparse data on recommendation, the matrix decomposition technology represented by Singular Value Decomposition (SVD) decomposed the user's rating of the item into the eigenvector matrix of the user and the item and utilized the potential relationship between the user and the item to obtain the predicted value, so as to achieve the dimension reduction of high-dimensional sparse matrix. Then, after dimensionality reduction [3], matrix decomposition uses low-dimensional space to estimate high-dimensional user-item interaction, affecting the accuracy of recommendation.

For the past few years, the deep learning technology developed by artificial neural network has achieved remarkable results in the fields of natural language understanding, speech recognition [4], and image processing. Deep neural network in deep learning has brought new ideas to the research of recommendation system. Therefore, by combining deep learning technology, it can make up for the simple linear combination of traditional collaborative filtering when learning complex user-item interactions.

Therefore, this paper proposes a deep collaborative filtering algorithm based on matrix factorization (CNNMF). The algorithm uses the Kronecker product to calculate the relationship between users and items to construct a relationship graph and then uses a convolutional neural network to perform nonlinear learning on the relationship graph. Finally, through comparative experiments, the influence of linear and nonlinear relationship on recommendation accuracy is investigated. The experimental data in this paper are extremely sparse implicit feedback data with little feature information.

The contributions of this paper are summarized as follows:

(1) Take Internet financial projects as the object, by designing recommendation algorithms, quantifying investors' preferences, and realizing accurate project recommendation, thereby improving the success rate of project financing.

(2) Define a new model called "deep collaborative filtering algorithm based on matrix factorization." The algorithm calculates the relationship between users and items by Kronecker product, constructs a relationship graph, and then uses the advantage of convolutional neural network to effectively learn local features to perform nonlinear learning on the relationship graph.

(3) The experimental data is extremely sparse implicit feedback data with little feature information. Combined with the characteristics of crowdfunding websites and recommendation algorithms, we conduct in-depth research to solve the problem of sparse implicit feedback data, which has certain practical significance and enriches the research on recommendation systems of crowdfunding platforms.

The rest of the paper is organized as follows. The second part of this paper will introduce the research status of collaborative filtering recommendation in detail and fully understand the main content, existing problems, and current research trends of collaborative filtering recommendation. In the third part, combined with the characteristics of collaborative filtering algorithm to model user-item interaction information and the advantages of convolutional neural network to extract features, the implementation process of deep collaborative filtering algorithm based on matrix decomposition is described in detail. In the fourth and fifth parts, experiments and analysis are carried out. First of all, this paper is a research on the data of crowdfunding websites, so the data of crowdfunding websites are collected, processed, and analyzed. Then, the evaluation method and experimental design are proposed. Furthermore, the experiment is carried out with the user-item implicit feedback data of crowdfunding websites as the object, and the influences of the number of hidden factors in the hidden layer, the number of convolution layers, and different negative feedback data collection methods on the recommendation effect are compared. And compare and analyze the performance of the algorithm proposed in this paper with the baseline method. Finally, a summary and outlook are given, and the research work and experimental results of this paper are summarized. On this basis, the shortcomings of the work are discussed, and the future research direction of the algorithm combining deep neural network and collaborative filtering is prospected.

2. Related Works

2.1. Recommendation in Crowdfunding Platforms. Recently, researchers have begun to focus on designing recommender systems for crowdfunding platforms, with the aim of increasing the success rate of projects getting fully funded. The research of the researchers mainly focuses on two aspects. On the one hand, the recommendation algorithm is constructed based on the mathematical model. For example, Vineeth et al. [5] proposed a probabilistic recommendation model called CrowdRec, which recommends projects to investors by combining the ongoing state of the project, the individual preferences of individual members, and the collective preferences of groups. Song et al. [6] proposed a recommender system based on a structural econometric model to match returning donors with fundraising activities on charitable crowdfunding platforms. Maximize the utility of altruism (from the welfare of others) and egoism (from personal motivation). Zhang and Zhang [7] proposed a personalized crowdfunding platform recommendation system, which is based on a multiobjective evolutionary algorithm. Consider the profit and variety of recommendations while capturing investor preferences. However, it is not very effective for crowdfunding platforms with a lot of data, on the other hand, building a crowdfunding platform personalized recommendation algorithm based on machine learning. For example, Benin [8] compared the application of various machine learning algorithms in crowdfunding platforms, such as gradient boosting tree, Bayesian belief nets collaborative filtering, latent semantic collaborative filtering etc. Wang and Chen [9] proposed a bipartite graph-based collaborative filtering model by combining collaborative filtering and personal rank. The model divides the nodes into user nodes and market activity nodes, calculates the global similarity by personal rank, and finally generates a recommendation list for any node through the collaborative filtering algorithm.

Although some scholars have found that collaborative filtering alleviates the problem of data sparseness, it is far from deep learning. Deep learning technology can extract more complex and abstract features from historical user-item interaction data and has a strong ability to represent large-scale data, but few scholars have applied it to crowdfunding platform recommendation.

2.2. Recommender Systems. Collaborative filtering algorithms have been widely used in recommender systems. Hu et al. [10] divided collaborative filtering recommendation algorithms into two main categories: one is memory-based collaborative filtering recommendation, and the other is model-based collaborative filtering recommendation. For the past few years, with the rapid rise of artificial intelligence research, deep learning technology is gradually applied to collaborative filtering recommendation. Since it mainly constructs models to learn user preferences, it is also a model-based collaborative filtering recommendation.

2.2.1. Memory-Based Collaborative Filtering. Memory-based collaborative filtering recommendations usually load data into memory for operations and generate recommendations based on similarity. These include user-based CF and item-based CF collaborative recommendation. For example, Zhan and Hong [11] proposed the ternary interaction between consumers, items, and producers. By using the collaborative filtering recommendation based on adversarial learning, disturbances were added to the parameters of the ternary model to solve the binary interaction problem and improve the accuracy. Zhu et al. [12] proposed an algorithm that fuses time weighting factors and item attributes based on user rating preferences. The algorithm considers the user's preference for items is time-sensitive and the impact of item attributes on similarity to improve the calculation of item similarity. However, it is difficult for both user-based and item-based algorithms to solve the problem of data sparsity effectively, so many scholars have conducted improvement research on this problem in recent years. For example, Hong and Yu [13] proposed a collaborative filtering recommendation algorithm based on correlation coefficient. This algorithm is founded on item-based collaborative filtering algorithm, fills unrated items in accordance with the correlation coefficient, and introduces semantic similarity to improve the calculation of item similarity to deal with data sparsity. Logesh et al. [14] proposed a collaborative filtering recommendation system based on bio-inspired clustering ensemble, which finds the closest neighborhood for users by clustering method and generates similarity matrix for recommendation.

2.2.2. Model-Based Collaborative Filtering. Model-based collaborative filtering is to build a model in the light of the user's historical ratings to predict scores. Such methods usually employ dimensionality reduction to extract latent features of users and items. For example, Wang et al. [15] proposed an algorithm that combines the Singular Value Decomposition (SVD) technology with the trust model, which reduces the dimension of a high-dimensional sparse matrix and then improves the prediction accuracy by introducing a trust factor. However, SVD is too slow when decomposing data with dimensions above 1000, and it has certain limitations in the calculation of up to tens of millions of dimensions in real systems.

In recommender system research, implicit feedback data is more common than explicit feedback data (such as ratings), including user clicks, purchases, and searches, etc., which has attracted more and more scholars' attention. For example, Hu et al. [16] model the implicit feedback data based on matrix factorization. They believe that explicit feedback represents the user's preference for items, while implicit feedback represents the confidence of the preference, so that the user behavior data is decomposed into preference and confidence. And then they use the objective function of matrix decomposition to solve it; Koren and Bell [17] proposed a matrix factorization model supporting implicit feedback (SVD++), which is an improved algorithm on the basis of SVD. The introduction of implicit feedback information increases the prediction accuracy. Bi-yi et al. [18] proposed an EIFCF algorithm combined with explicit and implicit feedback, which makes the most of the implicit feedback data to reflect the user‘s hidden preference and the explicit feedback to reflect the user's preference. Besides, weighted matrix factorization is used to overcome the problem of lack of negative samples in implicit feedback data, thereby alleviating the impact of data sparsity.

2.2.3. Deep Learning-Based Collaborative Filtering. With the continuous research and application of deep learning technology, collaborative filtering based on deep learning has developed into a hot research trend, which not only improves the accuracy of recommendation and broadens application scenarios, but also enriches model-based collaborative filtering recommendations.

Deep learning can use deep neural networks to learn the raw data, so as to find the more abstract feature representation relationships between users and items in a finer-grained manner. For instance, Yi et al. [19] proposed a deep matrix factorization model embedded with implicit feedback. This model constructs a deep network pool to extract latent factors from the input of user and item information and uses them to predict user ratings. Wu et al. [20] proposed a collaborative denoising autoencoder model (CDAE). By adding noise to the user's rating vector, the robustness of this model is improved, and the low-dimensional user implicit vector is obtained. And then it is used to predict the missing score.

Convolutional neural network has application in recommender systems to better address data sparsity issues. At present, convolutional neural networks are primarily planed on modeling auxiliary information of users or items (such as item descriptions, reviews, etc.), while matrix factorization techniques are still utilized to model user-item interactions (such as user-item ratings). For example, Kim et al. [21] proposed a convolutional matrix factorization model that associates convolutional neural network with matrix factorization. In this model, the convolutional neural network is used to extract the features in the context information and the auxiliary information of the item is used to address the sparsity issues and enhance the accuracy of rating prediction. Zheng et al. [22] constructed a convolutional neural network to learn features from using user and item reviews. And then user-item interaction information is obtained through matrix decomposition to predict ratings. Finally, the impact of data sparsity on recommendation results is reduced. Zhang et al. [23] proposed a collaborative knowledge base embedding learning model (CKE), which mines implicit feedback information through the knowledge base and obtains the implicit vectors of users and items to predict ratings.

2.3. Literature Review. Existing research has proposed many solutions to the problem of data sparseness. However, there is still room for improvement.

First: Memory-based collaborative filtering mainly reduces the sparsity of data by fusing features such as user and item attributes or clustering users. However, it is oriented towards explicit feedback data and collected auxiliary information of users and items, which is highly dependent on data and not scalable.

Second: Model-based collaborative filtering conducts research on implicit feedback information, and matrix decomposition technology also alleviates the problem caused by data sparsity to some extent. However, matrix factorization mainly uses a simple and fixed inner product for linear modeling, which is not conducive to estimating the complex characteristic relationship between users and items.

Third: Collaborative filtering based on deep learning handles the problem of sparse data through the learning of auxiliary information and extracts features to obtain deeper relationships between users and items. However, although these methods use auxiliary information to replace the construction of linear models, the sparsity problem of user-item interaction data in collaborative filtering recommendation still remains unsolved.

Therefore, in view of the above shortcomings, a recommendation algorithm based on deep learning and collaborative filtering is proposed in this paper. For the implicit feedback data set of crowdfunding projects, this model exploits the advantages of convolutional neural networks to learn local features effectively and the characteristics of collaborative filtering algorithm to model user-item interaction information to reduce the impact of user-item interaction information sparsity on recommendation accuracy.

3. Our Proposed CNNMF Method

3.1. Deep Collaborative Filtering Model. The deep collaborative filtering model in this paper combines matrix decomposition with deep neural network. The basic idea is shown in Figure 1. This model uses investors' investment behavior data and negative feedback data to perform the linear learning of matrix decomposition and nonlinear learning of deep neural network, respectively, and finally outputs the recommendation list.

The collaborative filtering recommendation algorithm firstly uses the user‘s preference for the item to form the user-item interaction matrix and models this interaction matrix to learn the interaction between users and items. Then the predictive value or recommendation list is obtained.

The model of deep collaborative filtering is shown in Figure 2. Each layer is implemented as follows.

Input layer. The input layer includes user ID and country, as well as item ID and category, and uses one-hot encoding to convert them into binary sparse vectors represented by v_d, v_c, v_d, v_g.

This paper analyzes the collection of negative feedback data in the implicit feedback data of crowdfunding websites. In the label data, the user behavior data with project investment behavior is marked as 1 in the implicit feedback matrix, and the uninvested project is marked as 0, as shown in

$$z_{uv} = \begin{cases} 1, & \text{invested}, \\ 0, & \text{uninvested}. \end{cases} \tag{1}$$

Since the implicit feedback data does not have the characteristics of negative feedback, 0 here does not represent the user's negative feedback intention. Therefore, in the experimental part, the negative feedback data will be uniformly collected and non-uniformly collected, and different negative feedback collections will have different effects on the model recommendation results.

Embedding layer. The embedding layer reduces the dimension of the sparse vector obtained by the input layer to obtain the feature matrix of the user, p_d, p_c, and the feature matrix of the item, q_d, q_g. As shown in formulas (2)–(5), the embedding matrix $p_{m\times k}$, $p_{t\times k}$, $p_{n\times k}$, and $p_{h\times k}$ is K-dimensional weight matrix. K represents the number of hidden factors set by the embedding layer, that is, the dimension of users and items after dimension reduction. The dense vector obtained by the linear transformation of the embedding layer not only represents the corresponding relationship of a single user or item, but also represents the internal relationship of all users or all items. During the training of the model, the embedding matrix updates the weights according to the user-to-user, item-to-item relationship.

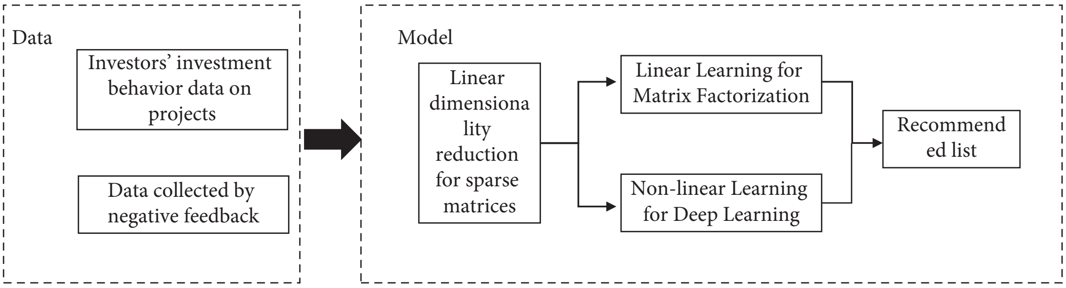
Data
Investors' investment behavior data on projects
Data collected by negative feedback
Model
Linear dimensionality reduction for sparse matrices
Linear Learning for Matrix Factorization
Non-linear Learning for Deep Learning
Recommended list

Figure 1: The basic idea of deep collaborative filtering algorithm.

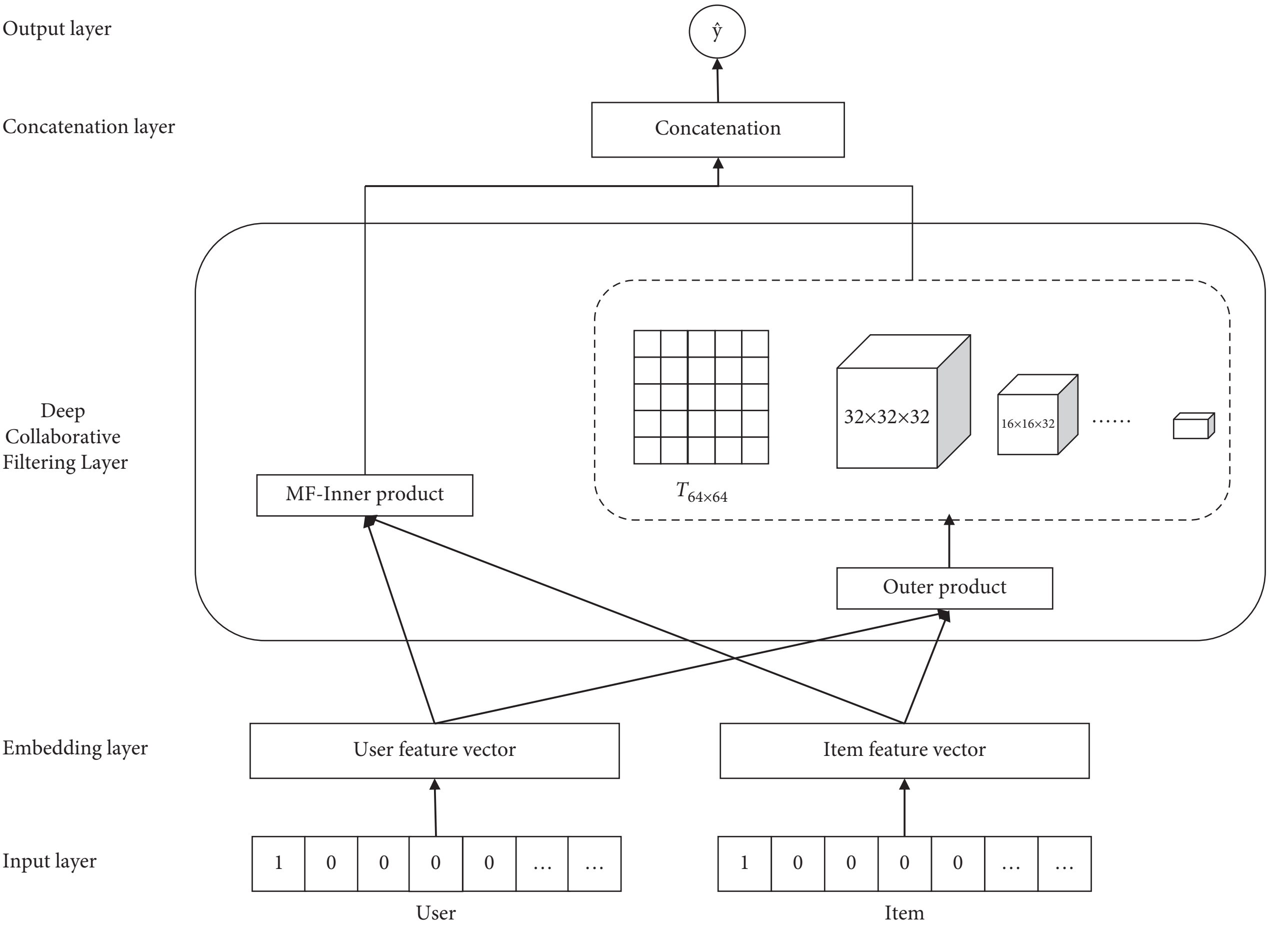

Figure 2: Deep collaborative filtering model.

$$p_d = P_{m\times k} v_d, \tag{2}$$

$$p_c = P_{t\times k} v_c, \tag{3}$$

$$q_d = P_{n\times k} v_d, \tag{4}$$

$$q_g = P_{h\times k} v_g. \tag{5}$$

After the output of the embedding layer, the user-related features are fused, and the item-related features are fused to form the user feature matrix and the item feature matrix, respectively, as shown in the following equations:

$$p_u = p_d + p_c, \tag{6}$$

$$q_i = q_d + q_g. \tag{7}$$

Deep Collaborative Filtering Layer. Perform linear learning for matrix factorization and nonlinear learning for deep neural network, respectively. The details are as follows.

Matrix Decomposition Layer. Matrix factorization linearly decomposes the Interaction matrix of users and items into two matrices multiplied together, and the obtained predicted score is the result of the interaction between the potential features of users and items. The operation of the matrix factorization layer is to perform a Hadamard product operation on the latent feature matrices of users and items, as shown in

$$f_1(p_u, q_i) = p_u \odot q_i. \tag{8}$$

Convolutional Neural Network Layer. The premise of the convolutional neural network layer is that the feature matrices of users and items perform Kronecker product operations. Kronecker product is the interaction of one element of a matrix with each element of another matrix, that is, learning its relationship with elements in other dimensions. Because there may be a correlation between each dimension of the features of users and items, the Kronecker product operation is used to generate the relationship feature matrix $T_{k\times k}$. Use ⊗ to denote the Kronecker product calculation. K is the number of hidden factors set by the embedding layer, as shown in equation (9). In the nonlinear learning of convolutional neural networks, the relationship matrix of users and items constitutes an interaction graph, and through the advantages of convolutional neural networks processing graphs, the interaction between users and items in the latent space is learned.

$$p_u \otimes q_i = \mathrm{T}_{kxk}. \tag{9}$$

In the nonlinear learning of convolutional neural networks, the relationship matrix of users and items constitutes an interaction graph, and through the advantages of convolutional neural networks processing graphs, the interaction between users and items in the latent space is learned, as shown in Figure 3.

The output of the last layer is the latent feature vector of users and items. Therefore, the overall calculation of the convolutional neural network layers is shown in

$$f_2(p_u, q_i) = \Phi_x(\ldots \Phi_2(\Phi_1(p_u \otimes q_i))). \tag{10}$$

The deep collaborative filtering layer combines the advantages of matrix decomposition and deep neural network to train recommendation models. Mapping the relationship between user and item to potential space by linear matrix factorization to obtain f_1, simultaneously, the deep neural network is leveraged to learn the hidden layer characteristics of users and items to obtain f_2. Such a combination of linear and nonlinear learning methods enables more precise matching between users and items.

Concatenation Layer. Concatenate the latent feature vectors f_1 and f_2 learned by matrix factorization and deep neural network.

Output Layer. The predicted value $\hat{y}$ is obtained by linear calculation, as shown in

$$\hat{y} = w\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} + b, \tag{11}$$

where m is the number of users, n is the number of items, f_1 is the inner product function of matrix decomposition, f_2 is the nonlinear function of deep learning, and Φ_x is the nonlinear learning of the x layer of deep neural network.

3.2. Deep Collaborative Filtering Algorithm.

3.2. Deep Collaborative Filtering Algorithm. Firstly, in the embedding layer, the algorithm uses matrix decomposition to obtain the feature dense vectors p_u and q_i of users and items. Matrix factorization can reduce the dimensionality of high-dimensional sparse data and predict ratings by calculating the inner product of the eigenvectors of users and items. However, the user's historical rating data is sparse, resulting in a large number of vacancies in the rating matrix. And matrix decomposition uses a simple linear combination to learn user-item interactions, which makes it difficult to fill the vacancies. Therefore, this paper introduces a convolutional neural network to calculate the potential nonlinear function of the user or item and then learns the potential relationship between users and items they interact with, so as to supplement the missing data.

Secondly, the algorithm learns the relationship of each dimension of the feature vector through the outer product and obtains the relationship feature matrix $T_{k\times k}$ of the users and the items.

Finally, the relationship feature matrix $T_{k\times k}$ is input into the convolutional neural network, local connectivity is extracted in the convolutional layer of each layer, and the convolutional kernel is used to convolve the relational matrix to extract the potential interaction of users and items.

The activation function of each layer is ReLU, which iterates, and the input of the next layer is the output of the upper layer. The convolutional kernel of each layer is a 2 × 2 matrix, and each convolution layer generates K channels $\Phi_1^1, \Phi_1^2, \ldots, \Phi_1^l$; that is, the number of channels is compared to the dimension of the embedding layer with the same number ($l = K$). The algorithm learns K features in the convolutional kernel, and the first convolution of the lth convolution kernel obtains $c_{1,1}^l$; Φ_1^l represents the feature matrix extracted by the lth convolutional kernel of the first layer after performing $s/2^l$ convolution operations, as shown in equations (12)–(14).

$$c_1^l = \mathrm{Re}LU\left(\sum_{i=0}^{1}\sum_{j=0}^{1} t_{i,j}\cdot w_{i.j} + b\right), \tag{12}$$

$$\Phi_1^l = \begin{bmatrix} c_{1,1}^l & \cdots & c_{1,\frac{s}{2^l}}^l \\ \vdots & \ddots & \vdots \\ c_{\frac{s}{2^l},1}^l & \cdots & c_{\frac{s}{2^l},\frac{s}{2^l}}^l \end{bmatrix}, \tag{13}$$

$$\Phi_1 = \left[\Phi_1^1, \Phi_1^2, \ldots, \Phi_1^l\right]. \tag{14}$$

The nonlinear activation functions include sigmoid, tanh, or ReLU. In this paper, the ReLU function is selected, which is

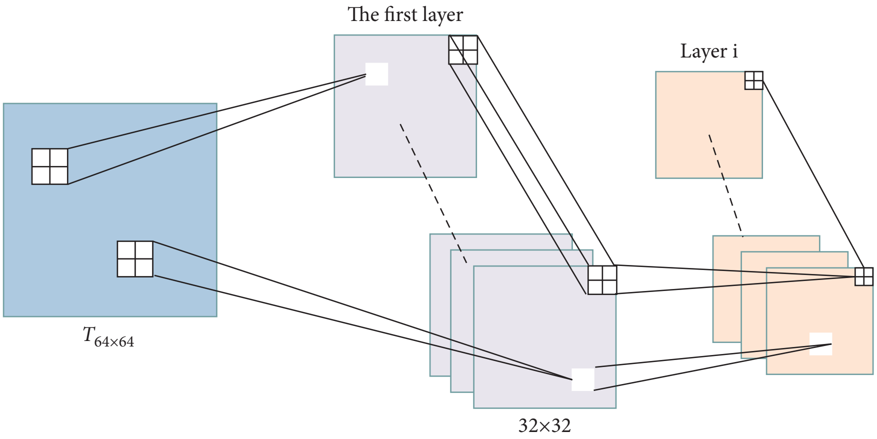

Figure 3: Structural diagram of convolutional neural network.

similar to the activation response of brain neurons to information. The ReLU function only activates the accepted information, while other information is filtered, so it is more suitable for sparse dataset calculation. Both the sigmoid and tanh functions have the problem of gradient disappearance, and overfitting will occur during the training process. The final results of the experiment also show that the performance of the ReLU function is better than the other two.

3.3. Deep Collaborative Filtering Model Training. Crowdfunding data in this paper are implicit feedback data with values of 1 or 0.1 meaning that the user has invested in the project; that is, the user is interested in the project. 0 may indicate that the user is not interested in the project or the user is unaware of the project's existence. Explicit feedback data (such as ratings) directly reflect the user's preference for an item, while the implicit feedback data cannot indicate the user's preference, but can only measure the confidence of the user's preference. Therefore, the final output of this model is not a prediction of ratings, but a personalized list of recommendations for the target user.

The ranking of items in the recommendation list is directly related to the user's satisfaction with the recommendation results. Therefore, the model adopts the Bayesian Personalized Ranking loss function [24] to learn the ranking of positive and negative feedback samples in pairs, as shown in equation (16).

In rating prediction, the goal of pointwise learning is to minimize the mean squared error between the predicted value and the label. When paired learning samples rank in the recommendation list, the positive feedback sample i should be ranked higher than the negative feedback j. And make the difference between the two predictions as large as possible. $\widehat{y}_{ui}$ is the predicted value of the positive feedback, and $\widehat{y}_{uj}$ is the predicted value of negative feedback, as

$$\widehat{y}_{uij} = \widehat{y}_{ui} - \widehat{y}_{uj}, \tag{15}$$

$$L = \sum_{(u,i,j)\in D} \ln \sigma\left(\widehat{y}_{uij}\right) - \lambda_{\Theta}\|\Theta\|^2, \tag{16}$$

where λ_{Θ} is the regularization parameter. D means $D{:} = \{(u,i,j)|i \in y_{ui} \wedge j \notin y_{ui}\}$; that is, user u interacts with item i.

In model training, Adaptive Moment Estimation (Adam) [25] is used. In stochastic gradient descent (SGD), the learning rate is kept constant during the training process. It uses a single learning rate to correct the weights, which takes a long time to train. However, Adam computes different adaptive learning rates for different parameters, which dynamically adjusts the learning rate of each parameter by using the first and second moment estimates of the gradient. Experimental results also show that Adam converges faster than SGD on our model.

4. Performance Analysis

4.1. Experimental Design. This paper takes the investment of users in projects on Ulule, the largest crowdfunding platform in Europe, as the research object. And then collect data and design experiments to measure the recommendation performance of this algorithm. The experimental process is divided into three parts: model, training, and experiment, as shown in Figure 4.

> Model part: We analyze the user behavior data of the crowdfunding website, learn user and item feature information through the deep collaborative filtering layer to generate a feature matrix, and then output it through the fully connected layer.
>
> Training part: We train the deep collaborative filtering model on the training set, output the predicted score of the item, and generate the recommendation list. Then, the model iterates, updates the weights and parameters iteratively based on the Adam training process, and saves the best recommendation list after 100 times, along with the weights and parameters at this time.
>
> Experimental part: We conduct experiments to verify the effects of data with different sparsity, the number of layers of convolutional neural networks, and implicit feedback data collection methods on recommendation performance and select the baseline method and design

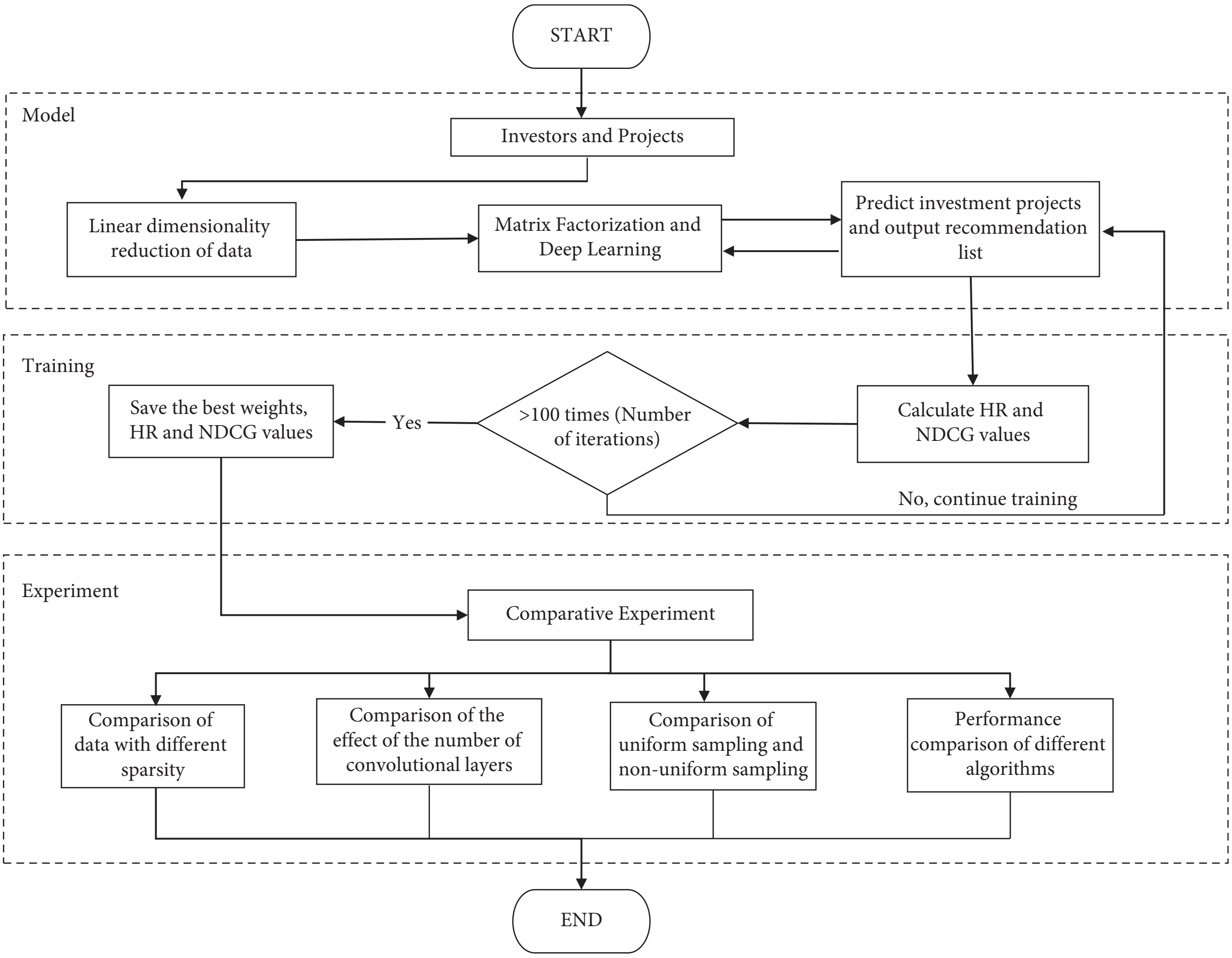
START
Model
Investors and Projects
Linear dimensionality reduction of data
Matrix Factorization and Deep Learning
Predict investment projects and output recommendation list
Training
Save the best weights, HR and NDCG values
Yes
>100 times (Number of iterations)
Calculate HR and NDCG values
No, continue training
Experiment
Comparative Experiment
Comparison of data with different sparsity
Comparison of the effect of the number of convolutional layers
Comparison of uniform sampling and non-uniform sampling
Performance comparison of different algorithms
END

Figure 4: Experimental procedure.

comparative experiments to verify the superiority of the proposed algorithm.

4.2. Experimental Data Collection. The homepage of the crowdfunding platform Ulule displays the number of users, the number of successful projects, and the success rate of the platform in real time. The website is open to everyone, and all data can be crawled through their API. Although the crowdfunding projects that have expired cannot be crawled, the website has collected statistics, including the number of successful financing projects each year, the financing success rate, the number of projects being funded, and the number of website user registrations, etc., as shown in Table 1.

Table 1: Ulule website related data.

Ulule website related data	Year			
	2019	2018	2017	2016
Number of financings	705,350	640,523	577,985	520,777
Number of project startups	6,042	8,049	7,452	7,045
Number of successful crowdfunding	4,398	5,152	4,856	4,804
Financing success rate	73%	64%	65%	68%

In this paper, web crawler technology is used to obtain experimental data. In terms of projects, it obtains key information such as project codes and classification labels; in terms of users, key information such as the codes of users and the codes of each invested project is crawled. The collected data includes the following: (i) In terms of projects, crawl the list of all successful financing and in-progress projects, where the content of the project list includes the project home page, whether the financing has been completed, the total amount of financing, the country of the publisher, the project code, and the classification label of the project. (ii) According to all the invested projects, crawl the projects that the user has invested in, including projects that are still in progress, where the content of the user list includes the user code, user homepage, user country, language, and time zone. Iterate in this way until the dataset has a certain size.

4.3. Experimental Data Description. This article crawled data from the Ulule crowdfunding website in April 2019. The data obtained is in Jason's data format. After cleaning the data, a total of 363,608 users' project investment behavior data were obtained. After statistics, there were 274,292 users and 41,894 projects where users participated in financing. Among them, 27,241 projects were successfully financed, with a success rate of 65%. As shown in Figure 5, most users invest less than 5 times, among which users who invest once are the most (this crawling does not include users with 0 investment), which is about 82.5%. Therefore, the data sparsity of the computing website is 99.98%.

The data sparsity influences the recommendation accuracy, which is one of the main research issues in this paper. Therefore, datasets with different sparsity are collected by us to observe the performance of the proposed model. And the data is processed according to the number of times users invest in crowdfunding projects. According to the degree of sparseness [26], it is divided into low-sparse data (9 or more investment times per capita), moderately sparse data (5 times or more per capita investment), and extremely sparse data (3 times or more per capita investment), as shown in Table 2. After statistics, the more investment times, the fewer users, the fewer projects, and the lower data sparsity.

4.4. Training Set and Test Set.

4.4. Training Set and Test Set. In this paper, the leave-one-out method is used to divide the training set and test set. The data set of n samples is divided into the training set of $n - 1$ samples and the test set of 1 sample. Firstly, 99 projects are randomly matched from the projects that the user has not invested in, and then the last invested project is taken out of the projects that the user has invested in to form a test set of 100 uninvested projects. Excluding one item extracted in the test set, the remnant items invested by users are the training set.

4.5. Collection of Negative Feedback Data. For the collection of negative feedback data, this paper conducts two experiments of uniform sampling and nonuniform sampling.

(i) Uniform sampling: Uniform sampling is to collect all uninvested projects of users as negative feedback data in each iteration or flexibly controls the ratio of positive feedback to negative feedback according to the experimental situation.

(ii) Nonuniform sampling: Nonuniform sampling is mainly to set a standard for the selection of items, such as high popularity, many exposure opportunities, etc., because users have a high probability of finding these items in a great volume of items. Therefore, if users do not invest, it is more likely that they are not interested in it.

We perform uniform and nonuniform sampling on extremely sparse datasets, respectively. Using the uniform sampling method, 4 negative feedbacks are randomly collected for each user, and the obtained negative feedbacks are evenly distributed among the 230 items, as shown in Figure 6.

Nonuniform sampling collects items according to their popularity, and the negative feedback obtained is concentrated on 7 items, while other items only get a small amount of collection, as shown in Figure 7.

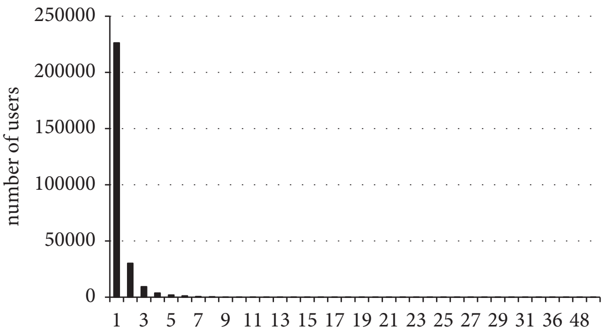

Figure 5: Statistics on the number of user investment projects.

In the collection of negative feedback in the data set, the fluctuation of randomly obtained negative feedback samples is small, and the degree of data dispersion is small, while the degree of dispersion of samples obtained by nonuniform collection is large.

4.6. Metrics. We employ two evaluation methods to measure the performance of the recommendation algorithm.

(i) Hit ratio (*HR*): in top-*K* recommendation, *HR* is a commonly used metric to measure recall. It calculates the ratio of fundraising items in the top-*K* recommendation list that belong to the test set, as shown in equation

$$\mathrm{HR} = \frac{n}{N}, \tag{17}$$

where n represents the number of items in the test set in the recommendation list, and N is the total number of items in the test set.

(ii) Normalized discount cumulative gain (NDCG): NDCG is an evaluation index used to measure the quality of the *TOP-K* ranking. It not only reflects whether the predicted *TOP-K* results are really relevant, but also reflects the relative ranking of the *TOP-K* results. Putting an item that users are not interested in at the top of the recommendation list will have a higher error rate, as shown in equations (18) and (19)

$$\mathrm{DCG} = \sum_{i=1}^{10} \frac{2^{rel_i} - 1}{\log_2 (i + 1)}, \tag{18}$$

$$\mathrm{NDCG} = \frac{\mathrm{DCG}}{\mathrm{IDCG}}, \tag{19}$$

where rel_i represents the correlation between the item and the user at position i in the recommendation list. Equation (18) shows that DCG combines recommendation accuracy with ranking position. In order to have comparability between different recommendation lists, normalization processing is performed, as shown equation (19). The value of DCG is between (0, IDCG] and the value of NDCG is between (0, 1].

Table 2: Data sets with different sparsity.

Dataset	Investments per capita	Sparsity (%)	Number of investors	Number of items	Total number of investments
Low-sparse data	9 times or more	94.36	639	220	7922
Moderately sparse data	5 times or more	96.81	2986	229	21847
Extremely sparse data	3 times or more	97.95	8598	230	40542

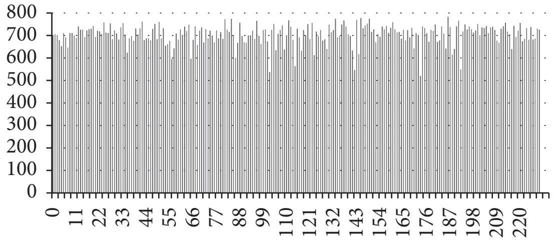

Figure 6: The distribution of negative feedback obtained by balanced collection method.

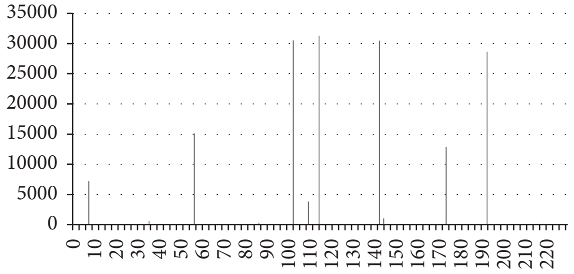

Figure 7: The distribution of negative feedback obtained by imbalanced collection method.

5. Experimental Results

5.1. Performance of Data with Different Sparsity on Recommendation Results. For the purpose of eliminating the impact of the number of convolutional layers on the experimental considerations result, the number of convolutional layers is set to 4, 5, and 6 layers for experiments, respectively. The obtained results are shown in Tables 3 and 4. The deep collaborative filtering model achieves the best recommendation effect on extremely sparse datasets when taking different convolutional layers. Under the circumstances of extremely sparse data and a large amount of data, the majority of memory-based collaborative filtering algorithms will be limited in the recommendation performance. This is because the similarity between users (items) will be difficult to calculate accurately as the data is sparse and the dimension increases, which will affect the recommendation effect. However, deep learning methods can break through this limitation and achieve better recommendation results on extremely sparse datasets. In addition, on the crowdfunding platform Ulule, the data sparsity is much larger than the extremely sparse dataset selected in this paper. Therefore, the proposed model is suitable for the Ulule platform with large and sparse data.

Table 3: The maximum value of HR of recommendation on datasets of different sparsity.

Layer	Dataset		
	Low-sparse	Moderately sparse	Extremely sparse
4	0.43348	0.47857	0.53920
5	0.46478	0.50569	0.54268
6	0.45696	0.49062	0.53512

Table 4: The maximum value of NDCG of recommendation on datasets of different sparsity.

Layer	Dataset		
	Low-sparse	Moderately sparse	Extremely sparse
4	0.23946	0.27741	0.31798
5	0.26223	0.29240	0.33299
6	0.26287	0.29350	0.32186

5.2. Performance of Convolutional Layers of Convolutional Neural Network on Recommendation Results. The number of the convolution layers of convolutional neural network will have a noticeable impact on the extraction of user and project features, which in turn affects the recommendation effect. Therefore, the effects of different convolutional layers on the recommendation results are compared through experiments.

Since the previous experimental results have proved that the proposed model can produce the best recommendation effect on extremely sparse datasets, this experiment will select extremely sparse datasets to construe the effect of the number of convolution layers. Moreover, similar algorithm is applied for the optimization of the parameter tuning process during the convolutional neural network training [27].

Figure 8 shows that as the number of convolutional layers increases, the HR value and the initial value of NDCG also increase, and when the convolutional layers increase to 6, the HR and NDCG values begin to decrease. It can be seen that the number of convolutional layers of the convolutional neural network has a significant impact on the recommendation results. And when the number of convolutional layers increases to a critical point, the optimal recommendation effect will be achieved. Continuing to increase the number of layers will reduce the accuracy of feature extraction of convolutional neural networks.

5.3. Performance of Negative Feedback Data Sampling Methods on Recommendation Results. The extremely sparse data set is also used as the experimental data. For each user,

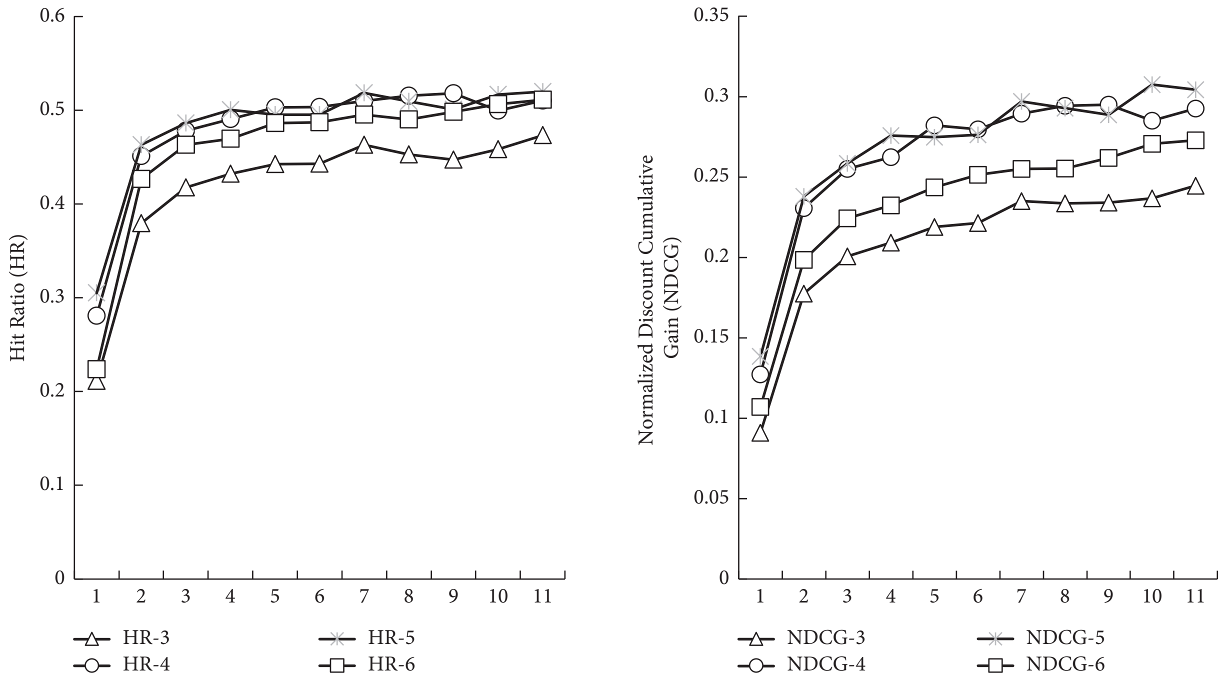

Figure 8: The HR and NDCG of recommendation based on the model with different convolution layers on extremely sparse dataset.

1~4 negative feedbacks are collected, respectively. The effects of the negative feedback data collection methods of uniform sampling and nonuniform sampling on recommendation are compared, and the results are shown in Tables 5 and 6.

Tables 5 and 6 show that the recommended effect of the uniform sampling method is better, and the best recommendation effect can be obtained when the number of collections is 1.

Figure 9 shows the iterative process of 4 negative feedbacks for uniform and nonuniform sampling [28]. The abscissa represents the interval of the number of iterations, and the ordinate represents the average value of HR and NDCG corresponding to each interval. In the iterative process, the convergence speed of the two methods is close, but the effect of uniform acquisition is obviously superior to that of nonuniform acquisition. Additionally, the peak values of HR are all in the third interval; that is, the best results are obtained when the iteration is about 20 times, indicating that increasing the number of experiments will limit the results. Different parameter settings have different convergence speeds of iterations.

The reason for the poor experimental results of nonuniformly sampled negative feedback data is probably that, in the data set constructed in this paper, users are not sensitive to the popularity of the project, and the negative feedback data set is highly discrete. The negative feedback sample collection based on the popularity of the items only focuses on the 7 most popular items, and the negative feedback repeatedly selects these 7 items, resulting in extremely poor results.

Table 5: The HR of recommendation with different negative feedback collection methods on extremely sparse dataset.

Number of negative feedback	HR uniform	HR nonuniform
1	0.50524	0.10329
2	0.52366	0.19405
3	0.53311	0.24883
4	0.54268	0.28936

Table 6: The NDCG of recommendation with different negative feedback collection methods on extremely sparse dataset.

Number of negative feedback	NDCG uniform	NDCG nonuniform
1	0.25393	0.05372
2	0.27859	0.0856
3	0.30628	0.09459
4	0.33299	0.11475

5.4. Comparative Experiments. This paper selects the five algorithms as baseline method and a deep collaborative filtering algorithm to design comparative experiments. The six baseline methods include classic recommendation algorithms and cutting-edge algorithms, so as to comprehensively examine the performance and effects of our algorithms.

Most Popular. The recommendation list of this method is sorted according to the popularity of the items, which is a nonpersonalized recommendation. All users get the same recommendation result. This algorithm is often used to compare recommendation results.

eALS. He et al. [29] proposed a matrix factorization model based on implicit feedback. The model weights the unrated parts according to the popularity of the item. Simultaneously, for the purpose of improving the

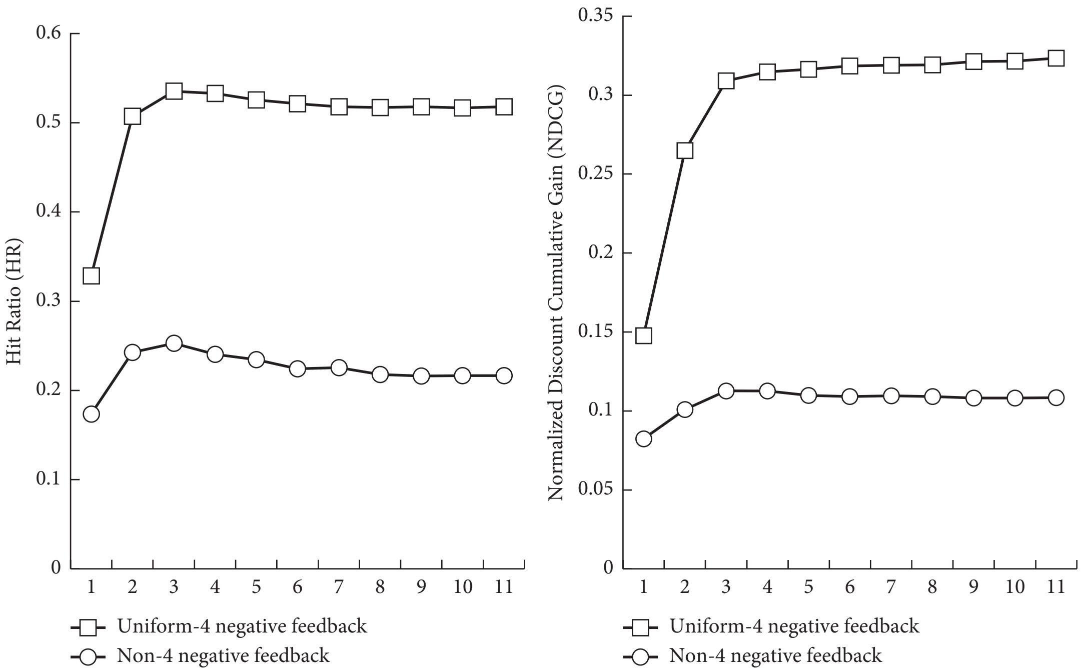

Figure 9: Uniform and nonuniform sampling of 4 negative feedback HR and NDCG values.

computational efficiency of model parameters, an eALS learning algorithm is designed, and an online update strategy suitable for dynamic data is established on this basis to capture users' short-term preferences. Moreover, the online scene enables the implicit feedback-based matrix factorization to be applied in large-scale practical environments.

BPR. A personalized ranking algorithm based on Bayesian theory has been proposed by Rendle et al. [24]. The algorithm sorts and optimizes the personalized recommendation list generated by the matrix decomposition or nearest neighbor method according to the implicit feedback data, so as to improve the user's satisfaction with the recommendation list. Using the partial order relationship between the purchased and unpurchased products, the maximum a posteriori estimate is derived by Bayesian analysis, and then the model is trained. Finally, the optimized item ranking is generated.

CM-RIMDCF. A two-stage deep collaborative filtering model has been proposed by Fu et al. [30]. In the first stage, users and items are learned separately through local and global models to obtain feature information representing user-user and item-item, and this is input into the second stage. Then, the second stage uses neural networks to learn information about user and item interactions, resulting in predictive scores.

Multicriteria DCF. Nassar et al. [31] proposed a multicriteria deep collaborative filtering model. The model divides user preferences into overall ratings and multicriteria ratings to quantify users' interest in different features of items. On this basis, a two-stage deep neural network model is designed. The first stage first predicts the multicriteria score and then feeds it into the next stage of the deep neural network to predict the overall score.

Deep Collaborative Filtering Algorithm (FMMLP). Inspired by the effective combination of deep learning technology and recommendation system in recent years, we combine the advantages of factorization machine second-order interaction with multilayer perceptron to fully learn the information of each dimension in user behavior. In this way, the recommendation effect of the model can be improved.

Comparative experiments use public datasets. Yahoo! Movies user rating dataset: Users rate movies on a scale of A+ to F. Since this dataset is an explicit feedback dataset, it needs to be preprocessed as implicit feedback data. The behavior of "rating" (regardless of the size of the score) is regarded as 1, indicating that there is an interactive behavior and 0 otherwise. In this dataset, users rated more than 10 movies, and each movie has at least one user rating. Therefore, with a total of 7,642 users and 11,915 movies, the rating (number of interactions) is 211,231 and the sparsity is 99.77%.

The recommendation results of our algorithm and the other method are shown in Table 7. According to the experimental results, both HR and NDCG values of CNNMF are significantly better than the other six methods. Therefore, the CNNMF algorithm can competently enhance the performance of recommendation.

Further analysis of the reasons is as follows:

(i) In the extremely sparse dataset (40,542 total investment times), the investment times of the top 10 most popular items accounted for 37.97% of the

Table 7: The comparison between the proposed algorithm and the baselines.

Recommendation algorithm	HR	NDCG
Most popular	0.492	0.388
eALS	0.563	0.444
BPR	0.520	0.410
CM-RIMDCF	0.422	0.327
MulticriteriaDCF	0.830	0.655
FMMLP	0.496	0.403
CNNMF	0.886	0.699

total investment times, but the top 10 items accounted for only 4.35% of the total number of items. It can be seen that the top 10 popular items have already received more attention from users, and it cannot provide users with personalized items. Therefore, the recommendation effect of the most popular algorithm is also the worst.

(ii) BPR and Eals algorithms propose different optimization methods for implicit feedback data but still build models based on traditional matrix factorization. Their processing of sparse data has certain limitations, which affects both model training and recommendation results.

(iii) The CM-RIMDCF algorithm learns the interaction of users and items based on the correlation between users and items. However, the local and global representation model proposed by this method, which uses explicit scoring for learning, is not suitable for implicit feedback. This affects the predictions of the second-stage neural network. In addition, the neural network in this method has only one fully connected hidden layer, which has certain limitations in learning the relationship between users and items.

(iv) The multicriteria scoring proposed by multicriteria DCF can explain the reasons for user preferences. However, in practical application scenarios, this type of data is limited, most of which are single overall scores. In addition, this method only uses deep neural networks to calculate multicriteria scores and overall scores, which is still insufficient in the fusion of deep neural networks and collaborative filtering.

(v) The effect of the algorithm FMMLP in this paper is also poor, including two reasons: One is that the Yahoo! Movies dataset has no feature information, so the factorization machine layer does not play a role in the algorithm. The second is that the algorithm is a serial structure. The second-order linear learning of the factorization machine is the input of the nonlinear learning. The feature learning of the factorization machine affects the nonlinear learning of the multilayer perceptron. The sparser the data, the greater the impact. So the algorithm performs worse on sparser data. The experimental results also show that it is the worst among the six personalized recommendation algorithms in this paper.

Therefore, the deep collaborative filtering model learns the potential relationship between user and item feature information more accurately by better integrating deep neural network and collaborative filtering, so as to obtain better recommendation effect.

6. Conclusion

In view of the extremely sparse data of crowdfunding platform Ulule, and the investment behavior of platform users with implicit feedback data without negative feedback, this paper integrates deep neural network and collaborative filtering recommendation algorithms to generate personalized crowdfunding project recommendations list for investors. In this way, the matching degree between investors and projects is improved, thereby improving the overall financing success rate of the crowdfunding platform.

For the recommendation problem of crowdfunding platform, the proposed algorithm has been enhanced in the following aspects:

(i) Extract the feature data of users and items, as well as the actual interaction information, and train the model.

(ii) Combine matrix factorization and convolutional neural network to learn feature information, respectively, to extract the potential relationship between user and item feature information. The complementary advantages of the two methods are used to improve the accuracy of feature extraction.

(iii) Sampling and analyzing the implicit feedback data of the crowdfunding platform through uniform and nonuniform negative feedback sample sampling methods.

For the purpose of verifying the effectiveness of the algorithm, five baseline methods are selected for comparative experiments. The final experimental results show that the recommendation effect of the deep collaborative filtering model on the dataset is significantly improved compared with the baseline model. At the same time, it is proved that integrating deep neural network to learn the nonlinear interaction between investors and projects can effectively address the issue of data sparsity. And for the problem of larger and sparser data, the best recommendation performance can be obtained by adjusting the number of convolutional layers of the convolutional neural network. In addition, for the collection of negative feedback data, the characteristics of the project need to be considered. Although nonuniform collection has a better interpretation effect, it is not necessarily suitable for the training of this model.

This study not only helps to enhance the financing success rate of crowdfunding platforms, but also enriches the research on recommender systems. In future research, the following aspects are worth exploring in depth:

(i) Consider contextual information, including industry developments, macroeconomics, and timing effects. Taking these factors into consideration will help to predict user preference.

(ii) Experiments can be carried out on the user and project data of different Internet financial platforms to further explore the generality of the method proposed in this paper.

(iii) Add auxiliary information as experimental data, such as user comments, item description information and item type, etc., to enrich the feature information of users and items.

(iv) Further explore the application of other deep learning methods in recommender systems.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest with any financial organizations regarding the material reported in this manuscript.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 71771177 and 71871143). This financial support is gratefully acknowledged by the authors.

References

[1] A. Agrawal, C. Catalini, and A. Goldfarb, "Some simple economics of crowdfunding," *Innovation Policy and the Economy*, vol. 14, no. 1, pp. 63–97, 2014.

[2] M. H. Por, S. B. Yang, and T. Kim, "Successful crowdfunding: the effects of founder and project factors," in *Proceedings of the 18th Annual International Conference on Electronic Commerce: e-Commerce in Smart connected World*, pp. 1–7, Suwon, Republic of Korea, 2016.

[3] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl, "Application of dimensionality reduction in recommender system-a case study," in *Proceedings of the KDD Workshop on Web Mining for e-Commerce: Challenges and Opportunities*, Boston, MA, USA, 2000.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[5] R. Vineeth, C. L. Wang, and K. R. Chandan, "Probabilistic group recommendation model for crowdfunding domains," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pp. 257–266, San Francisco, CA, USA, 2016.

[6] Y. Song, Z. Li, and N. Sahoo, "Matching returning donors to projects on philanthropic crowdfunding platforms," *Management Science*, vol. 68, no. 1, pp. 355–375, 2021.

[7] L. Zhang, X. Zhang, F. Cheng, X. Y. Sun, and H. K. Zhao, "Personalized recommendation for crowdfunding platform: a multi-objective approach," *in Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand*, pp. 3316–3324, 2019.

[8] A. C. Benin, "A comparison of recommender systems for crowdfunding projects," *Universidade Federal do Rio Grande do Sul*, Porto Alegre, Brazil, 2018.

[9] H. Wang and S. Chen, "A bipartite graph-based recommender for crowdfunding with sparse data," *Banking and Finance*, IntechOpen, London, UK, 2020.

[10] Y. Hu, Q. Peng, X. Hu, and R. Yang, "Time aware and data sparsity tolerant web service recommendation based on improved collaborative filtering," *IEEE Transactions on Services Computing*, vol. 8, no. 5, pp. 782–794, 2015.

[11] W. Zhan, Z. Hong, L. Fang, Z. Wu, and Y. Lyu, "Collaborative filtering recommendation algorithm based on adversarial learning," *Computer Science*, vol. 48, no. 7, pp. 172–177, 2021.

[12] L. Zhu, Q. Hu, L. Zhao, and J. Yang, "Collaborative filtering algorithm based on rating preference and item attributes," *Computer Science*, vol. 47, no. 04, pp. 67–73, 2020.

[13] B. Hong and M. Yu, "A collaborative filtering algorithm based on correlation coefficient," *Neural Computing & Applications*, vol. 31, no. 12, pp. 8317–8326, 2019.

[14] R. Logesh, V. Subramaniyaswamy, D. Malathi, N. Sivaramakrishnan, and V. Vijayakumar, "Enhancing recommendation stability of collaborative filtering recommender system through bio-inspired clustering ensemble method," *Neural Computing & Applications*, vol. 32, no. 10, pp. 2141–2164, 2020.

[15] J.-F. Wang, X. Li, W.-Q. Wu, and Y. Liu, "An algorithm of collaborative filtering based on SVD and trust factors," *Journal of Chinese Computer Systems*, vol. 38, no. 06, pp. 1290–1293, 2017.

[16] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proceedings of the 9th IEEE International Conference on Data Mining*, pp. 263–272, Miami, FL,USA, 2009.

[17] Y. Koren and R. Bell, "Advances in collaborative filtering, recommender systems handbook," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds., Springer, Boston, MA, pp. 77–118, 2015.

[18] C. Bi-yi, L. Huang, C.-D. Wang, and L. Jing, "Explicit and implicit feedback based collaborative filtering algorithm," *Journal of Software*, vol. 31, no. 3, pp. 794–805, 2020.

[19] B. Yi, S. Shen, H. Liu et al., "Deep matrix factorization with implicit feedback embedding for recommendation system," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 8, pp. 4591–4601, 2019.

[20] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-n recommender systems," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pp. 153–162, San Francisco, CA, USA, 2016.

[21] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in *Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 233–240, MA, USA, 2016.

[22] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 425–434, Cambridge, UK, 2017.

[23] F. Zhang, N. J. Yuan, D. Lian, X. Xing, and M. Wei-Ying, "Collaborative knowledge base embedding for recommender systems," in *Proceedings of the 22nd ACM SIGKDD International Conference*, pp. 353–362, San Francisco, CA, USA, 2016.

[24] S. Rendle, C. Freudenthaler, Z. Gantner, and S. T. Lars, “BPR: Bayesian personalized ranking from implicit feedback,” in *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pp. 452–461, Montreal, Canada, 2009.

[25] D. Kingma and J. Ba, “Adam: a method for stochastic optimization,” in *Proceedings of the 3rd International Conference for Learning Representations*, pp. 1–15, San Diego, CA, USA, 2015.

[26] Z. Chen and J. Yan, “Collaborative filtering recommendation algorithm based on sparse data preprocessing,” *Computer Technology and Development*, vol. 26, no. 7, p. 6, 2016.

[27] Y. Jiang, G. Tong, H. Yin, and N. Xiong, “A pedestrian detection method based on genetic algorithm for optimize XGBoost training parameters,” *IEEE Access*, vol. 7, pp. 118310–118321, 2019.

[28] J. Ding, Y. Quan, Q. Yao, Y. Li, and D. Jin, “Simplify and robustify negative sampling for implicit collaborative filtering,” *NeurIPS*, 2020.

[29] X. He, H. Zhang, M. Y. Kan, and C. Tat-Seng, “Fast matrix factorization for online recommendation with implicit feedback,” in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 549–558, Italy, 2016.

[30] M. Fu, H. Qu, Z. Yi, L. Lu, and Y. Liu, “A novel deep learning-based collaborative filtering model for recommendation system,” *IEEE Transactions on Cybernetics*, vol. 49, no. 3, pp. 1084–1096, 2019.

[31] N. Nassar, A. Jafar, and Y. Rahhal, “A novel deep multi-criteria collaborative filtering model for recommendation system,” *Knowledge-Based Systems*, vol. 187, Article ID 104811, 2020.