Hindawi
Complexity
Volume 2020, Article ID 8834908, 12 pages
https://doi.org/10.1155/2020/8834908

Research Article

Personalized News Recommendation and Simulation Based on Improved Collaborative Filtering Algorithm

Kunni Han [1,2]

[1] *School of Journalism and Communication, Qingdao University, Qingdao 266071, China*
[2] *Institute for Research on Portuguese-Speaking Countries, City University of Macau, Macau SAR 999078, Macao, China*

Correspondence should be addressed to Kunni Han; hkn0532@qdu.edu.cn

Received 11 September 2020; Revised 10 October 2020; Accepted 15 October 2020; Published 23 October 2020

Academic Editor: Wei Wang

Faced with massive amounts of online news, it is often difficult for the public to quickly locate the news they are interested in. The personalized recommendation technology can dig out the user's interest points according to the user's behavior habits, thereby recommending the news that may be of interest to the user. In this paper, improvements are made to the data preprocessing stage and the nearest neighbor collection stage of the collaborative filtering algorithm. In the data preprocessing stage, the user-item rating matrix is filled to alleviate its sparsity. The label factor and time factor are introduced to make the constructed user preference model have a better expression effect. In the stage of finding the nearest neighbor set, the collaborative filtering algorithm is combined with the dichotomous K-means algorithm, the user cluster matching the target user is selected as the search range of the nearest neighbor set, and the similarity measurement formula is improved. In order to verify the effectiveness of the algorithm proposed in this paper, this paper selects a simulated data set to test the performance of the proposed algorithm in terms of the average absolute error of recommendation, recommendation accuracy, and recall rate and compares it with the user-based collaborative filtering recommendation algorithm. In the simulation data set, the algorithm in this paper is superior to the traditional algorithm in most users. The algorithm in this paper decomposes the sparse matrix to reduce the impact of data sparsity on the traditional recommendation algorithm, thereby improving the recommendation accuracy and recall rate of the recommendation algorithm and reducing the recommendation error.

1. Introduction

With the deepening of informatization and rapid changes in network technology, the era of information explosion is coming one after another, and the ways for users to obtain information are becoming more abundant [1, 2]. It is true that the rapid development of information technology allows users to query information that they consider valuable from a richer information resource [3]. However, the massive amount of information presented at the same time makes it difficult for users to discover what they are interested in. We have entered the era of information overload from the era of information scarcity. At this time, how to accurately and efficiently filter out the content that users are really interested in from the dazzling information has become more and more important [4].

The content-based recommendation algorithm first models real-time news, then builds a user interest model based on the browsing information of a specific user, and finally recommends news events that are similar to the target user's interest model but are not included in the browsing history [5, 6]. It can be seen that constructing a real interest model and calculating the similarity between the models are the key points of this algorithm. Relevant scholars apply collaborative filtering to the Tapestry mail system [7]. The system reorders the new mail received by analyzing the historical behavior habits of users' mail reading, so as to improve the efficiency of users' mail reading [8]. In the Tapestry system, users can determine the type of e-mail based on their own interests and can decide whether to read this e-mail based on the label of the e-mail [9]. The system cannot actively recommend according to the user's interest preferences, which limits the system to only smaller users. It also requires a high degree of cooperation from users. Although Tapestry has many technical flaws, its more important role is to show us a new recommendation idea [10, 11]. Subsequently, the University of Minnesota built and launched a movie recommendation website. In this system, users rate the movies they have watched, and then the system recommends movies with similar ratings to users based on the ratings, which is more convenient to use [12]. In addition, a laboratory developed the music recommendation system Ringo, which requires users to compare the ratings of musicians, calculate the similarity between users based on the results of the ratings, and cluster users with higher similarities together [13]. Web Watcher is one of the more popular early personalized recommendation systems. At the beginning, it requires users to feed back their personal interests and form user attribute characteristics. Then, combined with the user's browsing history, it provides users with recommended links based on the benefit links of the current user's greatest interest [14]. Personal Web Watcher is an improved system of Web Watcher. It no longer requires users to describe their interests but builds an interest model for users based on the web pages they have visited. The Amazon.com book recommendation website system adopts collaborative filtering technology, which can analyze all users' purchases of books in a timely and accurate manner and then recommend books that have been purchased by other users who have purchased the same book to users [15, 16]. The purchase history, products of interest, and other information including browsing products, topics of interest, demographic characteristics, and other information are combined together, and finally a list of books that users may buy (like) is displayed to users [17]. The potential and real demand also reflects the value of recommendation algorithms in commercial applications [18]. The research done by these researchers on recommendation technology and recommendation system applications is of great significance to the development of personalized recommendation systems [19, 20].

In this paper, the dichotomous K-means clustering algorithm has been improved and combined with the collaborative filtering algorithm. The label factor and time factor are introduced, the similarity measurement formula is improved, and an improved collaborative filtering recommendation algorithm is obtained. This paper conducts an experimental test on the algorithm proposed in this paper. In the experiment, a simulated data set and a real data set were selected to test the performance of the algorithm and compared with the traditional user-based collaborative filtering recommendation algorithm. On the simulated data set, this paper tests the performance of the algorithm in this paper and the traditional algorithm under different neighbor sizes and different similarity measures. Experiments show that the algorithm proposed in this article is better than the traditional algorithm. On the real data set, this paper first tested the pros and cons of the three similarity schemes and then selected a similarity scheme to test the algorithm in this paper and the traditional algorithm under different neighbors.

2. Related Theories and Technologies

2.1. Personalized Recommendation System. Personalized recommendation system is a very effective solution to solve information overload. It recommends information of interest to users based on their behavioral data and interest preferences. The recommendation system can calculate the similarity by studying the user's interests and preferences and finally help users find their information needs. A good recommendation system should not only provide users with personalized services but also establish a close relationship with users, so that users can rely on them. The personalized recommendation system has now been widely used in many fields, including e-commerce, video, news, email, etc. At the same time, the research enthusiasm in academia is also very high, and it has gradually formed an independent subject. The recommendation system has three important modules: user modeling, recommendation objects, and recommendation algorithms. The traditional recommendation system model process is shown in Figure 1.

The content-based recommendation method originated in the field of information acquisition, and it is an important research content in the field of information retrieval. The algorithm first extracts the content characteristics of the recommended objects and then matches the user interest preferences in the user model and finally recommends the objects with higher matching degrees to the user. For example, in a news recommendation system, the system first analyzes the commonality of the user's previous reading of news, finds his interest preferences, and then recommends other news similar to his interest. The key part of the recommendation strategy is to calculate the similarity between the content feature of the recommended object and the interest feature in the target user model.

Content-based recommendation systems are widely used to recommend objects with specific text characteristics. Their operation object is text content, and the corresponding weight is mainly given according to the frequency of the key words in the text, and finally the relationship between the texts is calculated for the final prediction and recommendation. Content-based recommendation systems usually use the TF-IDF method in information filtering. TF-IDF is mainly used to measure the importance of a keyword to the entire text content. It has two measurement standards: term frequency (TF) and reverse document frequency IDF (inverse document frequency). TF represents the ratio of the number of files containing keywords to the total number of files. When the IDF value is larger, the adjective word appears in multiple files, so the word cannot be used as a keyword to distinguish files. When the IDF value is smaller, it means that the word only appears in one or a small number of files.

The disadvantages of content-based recommendation are as follows:

① The wide application of content-based recommendation is restricted by the problem of feature extraction ability of recommended objects. This is because there is no effective feature extraction

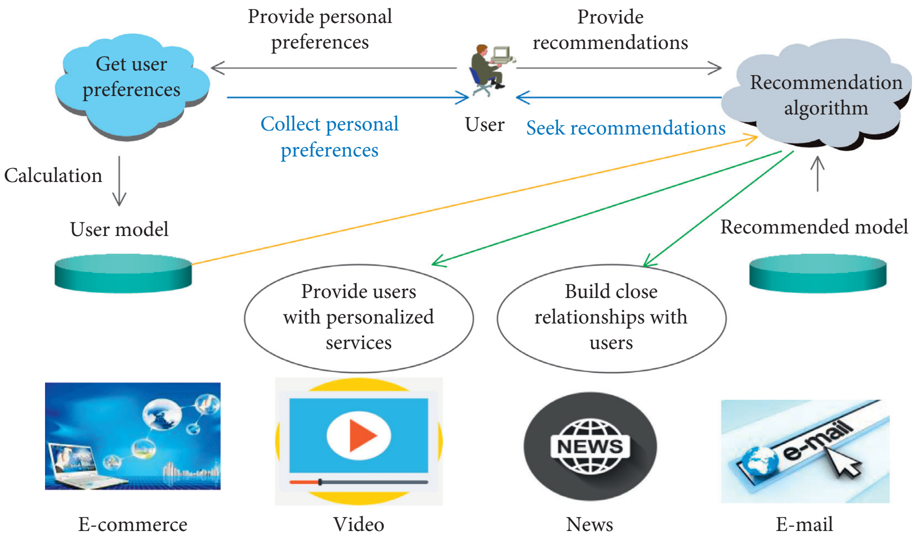

Figure 1: General model of the recommendation system.

method for multimedia resources. Even for text resources, feature extraction can only reflect part of the content of the multimedia resources.

② In the case of new users, there is a cold start problem. When a new user joins, it is difficult for the system to obtain the user's interest and thus cannot match the content characteristics of the recommended object. As a result, satisfactory information cannot be recommended to the user.

③ It is difficult to recommend new content. Recommendation can only be made when the content feature of the recommended object matches the user's interest feature. Therefore, the user is limited to obtaining recommendation results similar to the previous ones, and it is difficult to obtain new interest information that the user has not discovered or easily ignored.

④ The classification method of recommended target content requires a large amount of data.

⑤ Another problem faced by content-based recommendation systems is the incompatibility between user models described in different languages and recommended object models.

2.2. Collaborative Filtering Recommendation. The basic idea of collaborative filtering recommendation comes from daily life and is inspired by the purchase of commodities and the selection of interest news. If a friend with similar interests buys a certain product, then the probability of buying this product is also high. When a user likes a certain type of product, when he sees a product similar to this type of product and when other users have a high evaluation of this product, the purchase probability will also be high. The process of user-based collaborative recommendation is shown in Figure 2.

Item-based collaborative recommendation is based on the assumption that if users have very similar ratings for some recommended objects, then current users' ratings for these items are also very similar. Similar to if many users trust a certain brand, it is relatively easy for other users to choose the brand.

The basic idea of project-based collaborative filtering is as follows:

First, it needs to find the nearest neighbors of the target object; second, you predict its score on the target recommendation object based on the current user's score on the nearest neighbor, because the current user's score on the nearest neighbor object is relatively similar to the score on the target object; third, it is necessary to select the top N target objects with high prediction scores to recommend to the current user.

2.3. Process of Collaborative Filtering Recommendation. Collaborative filtering recommendation is to generate a recommendation list for target users based on the preferences of similar neighbors. It first searches several neighbors of the target user, then predicts the target user's score for the item based on the user's score in the nearest neighbors, and finally generates a recommendation list. From the introduction in the previous chapters, we know that, in order to use collaborative filtering for personalized recommendation, the following three conditions must be met: first, users do not exist independently, and there is a certain relationship between users; second, the rating matrix can show some users' interest preferences and potential preferences; third, users can predict and score the items based on users in similar neighbor sets. The traditional collaborative filtering recommendation process first constructs a user-item rating matrix, then calculates user similarity according to the rating matrix, and finally predicts the recommendation.

The collaborative recommendation system must first obtain the user's previous consumption, evaluation, browsing information, and item attribute information and perform data preprocessing on these data to obtain a matrix $R_{m\times n}$ of user-item ratings. m represents the number of users,

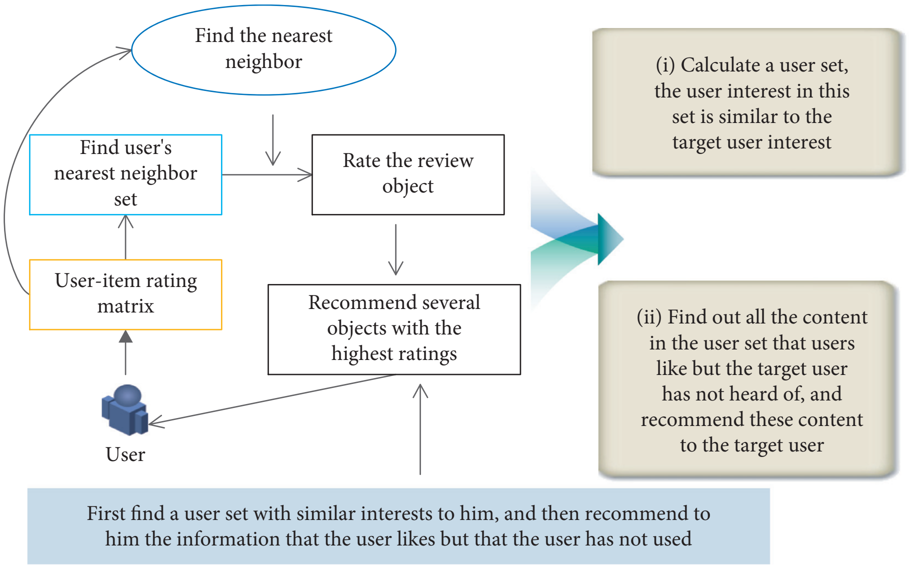
Find the nearest neighbor
Find user's nearest neighbor set
Rate the review object
User-item rating matrix
Recommend several objects with the highest ratings
User
(i) Calculate a user set, the user interest in this set is similar to the target user interest
(ii) Find out all the content in the user set that users like but the target user has not heard of, and recommend these content to the target user
First find a user set with similar interests to him, and then recommend to him the information that the user likes but that the user has not used

Figure 2: User-based collaborative filtering recommendation process.

n represents the number of items, and R_{ij} represents the rating value of user i for item j. Usually, the rating value ranges from 1 to 5, where 1 means dislike and 5 means very much. If the item is not rated, then the score value is 0. The rows in the rating matrix represent the user's rating vector, and the columns represent the news rating vector. If there is no scoring setting on the front end of the system, but user log information can be obtained, these logs contain user browsing, evaluation, publishing, and other behavior information; then these behavior types can be marked with different weights; for example, browsing behavior is marked as 1 point, the behavior is marked as 2 points, and the publishing behavior is marked as 2 points, to achieve the user's final score on the news.

Cosine similarity regards the user rating matrix as a vector in the n-dimensional item space. If the user does not rate the item, the rating value is set to 0. The cosine similarity method is usually applied to text objects. The angle value of the vector is used to indicate the similarity. When calculating the similarity between users, the data in the rating matrix is regarded as a vector, and the cosine value is the similarity value between two users and is proportional to the user similarity value. The formula for calculating the cosine similarity is as follows:

$$\bar{r}_{a,j} = \sum_{i\in s} \mathrm{sim}\left(R_{i,j}, a_i\right). \qquad (1)$$

Among them, $R_{i,j}$ and a_i, respectively, represent the specific score value of the corresponding position in the matrix. Cosine similarity sets the value of the user's unrated items to 0, which effectively improves the calculation efficiency. But in fact, the user's preference for unrated items cannot be all zero. Therefore, in the case of sparse data, the cosine similarity cannot correctly calculate the similarity between users or items. Another problem with cosine similarity is that the user's rating scale is not considered. The modified cosine similarity solves this problem.

2.4. Collaborative Filtering Bottleneck Problems and Solutions.

The continuous expansion of the recommendation system has led to a sharp increase in the number of users and products, which has brought many thorny problems to collaborative filtering, such as data sparseness, cold start, and scalability issues. These three bottlenecks and their solutions are described in detail in the following.

The recommendation system is getting bigger and bigger, and the number of users and items is increasing, which makes the possibility of news overlapping among users become smaller and smaller, which leads to the problem of data sparsity. Generally, the ratio of the number of ratings between users and products to the number of possible ratings is used to measure the data sparsity of the system. After calculation, the data sparsity of Movielens is 4.5%, and the data sparsity of Netflix is 1.2%. In fact, these are already relatively dense data sets, and the data sparsity of Delicious is 0.046%.

The cold start problem is often encountered in recommendation systems. It is mainly divided into project cold start and user cold start. The item cold start problem means that when a new item is first added to the recommendation system, no user has rated it. The collaborative filtering recommendation algorithm analyzes and recommends the item based on the user's historical rating information. Therefore, if no user has ever rated it, then it will be difficult to get recommendations. User cold start mainly refers to that, in the early stage of the recommendation system, there is no user-item rating information in the system, and the collaborative filtering recommendation algorithm cannot be used to complete the recommendation. The working schematic diagram of the recommendation system based on collaborative filtering is shown in Figure 3.

The scale of the recommendation system is getting bigger and bigger, and the number of users and items is also increasing. As all users and items need to find neighbors, the amount of similarity calculation is also rising sharply. There

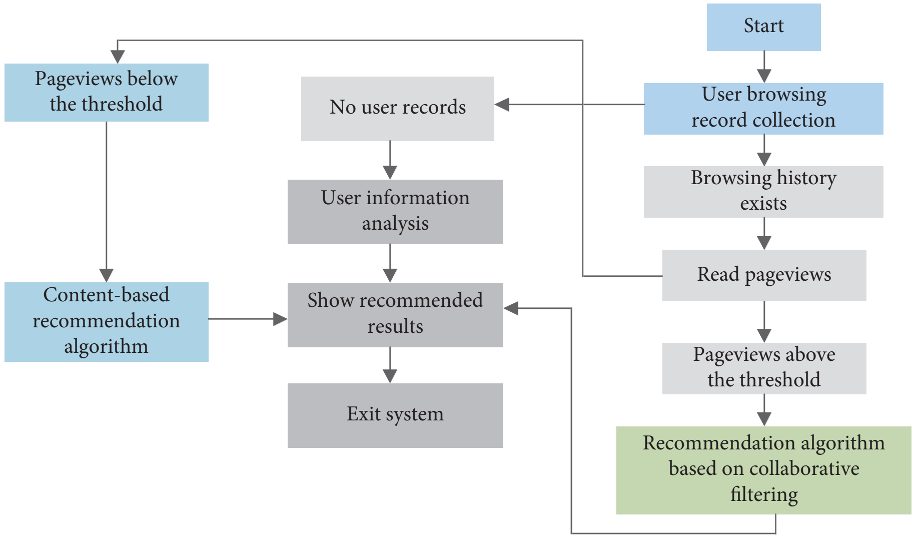
Start
User browsing record collection
No user records
Pageviews below the threshold
Browsing history exists
User information analysis
Read pageviews
Content-based recommendation algorithm
Show recommended results
Pageviews above the threshold
Exit system
Recommendation algorithm based on collaborative filtering

Figure 3: Schematic diagram of the recommendation system based on collaborative filtering.

are also many problems in calculating the most similar set of users or items on such a huge data set and recommending any target user in real time, such as how to simultaneously ensure the scalability of the recommendation algorithm and quickly and accurately calculate the predicted value and improve the real-time performance of the system. Therefore, it is also a research focus to improve the scalability and real-time performance of the system without affecting the recommendation accuracy. The dimensionality reduction technology can be used to solve the scalability problem, and the size of the recommender system can be reduced by dimensionality reduction, thereby improving the scalability of the system. In addition, classification and clustering techniques are often used to solve this problem. Classification techniques mainly use classifiers to train data sets. Once a good classifier is created, it can be used to classify new items.

3. Improved Collaborative Filtering Algorithm

3.1. Fill in Scoring Data. Aiming at the shortcomings in the collaborative filtering algorithm, this article uses the following ways to improve: first, in the sample processing stage, the similarity between the item and the target user's scored item can be used to predict the target user's score, and then the user item-rating matrix is filled, which can alleviate the sparseness of the user-item rating matrix and improve the recommendation quality of the algorithm; the second is to combine the item tag and rating time with the user rating by introducing tag factors and time factors. The performance measurement formula is improved so that the set of nearest neighbors of the target user can better simulate user preferences; the third is to combine the collaborative filtering algorithm with the dichotomous K-means clustering algorithm. Before using the collaborative filtering algorithm, you use the dichotomous K-means clustering algorithm. The means clustering algorithm determines the user cluster to which the target user belongs, and then uses it as the search range of the nearest neighbor set, so as to avoid the collaborative filtering algorithm from searching on the entire user set, and the binary K-means algorithm can maintain a good performance in large data sets, so it will be well displayed in the recommendation system. Since the improved collaborative filtering algorithm only considers the clustered user clusters when determining the nearest neighbor set, when the client recommendation system architecture is adopted, the user cluster information can be arranged on the local client, making the collaborative recommendation strategy easier.

With the increase of commodity items, the operation effect of traditional user-based collaborative filtering recommendation algorithm in the recommendation system is affected. This is because the number of users in the recommendation system is much smaller than the number of commodity items, resulting in a severely sparse user-item evaluation matrix. At this time, if the similarity between users is directly calculated, the recommendation quality of the recommendation system will be affected due to the reduction of common scoring items among users.

In order to solve this problem, this research first needs to fill the user-item rating matrix to alleviate its sparsity. The specific operation is to predict the items that user a and user b have not scored in Ia and b, and reduce the sparsity of the user-item score matrix by expanding the common item score set between users. The principle of filling is to use the similarity between the items that have been evaluated by the target user and the items that have not been evaluated by the target user to predict the score of the un-evaluated items. For example, user A has scored four items $I1$, $I2$, $I3$, and $I4$. If you want to know the predicted score of user A on item $I5$, you must first calculate the similarity between item $I5$ and items $I1$, $I2$, $I3$, and $I4$. Then, you can predict that user A may rate item $I5$:

$$P_{u,i} = \frac{\sum_{j\in \mathrm{NBS}_i} R_{u,j} \cdot S_{i,j}}{\sum_{j\in \mathrm{NBS}_i} \left| S_{i,j} \right|}. \tag{2}$$

Among them, NBS_i is the set of items with the highest similarity between user u's rated items and item i, $S_{i,j}$ is the similarity between item i and item j, $R_{u,j}$ is user u's rating of item j, and $P_{u,i}$ is user u's predictive score for item i.

3.2. Forecast Time Scale Value. When the traditional collaborative filtering algorithm makes recommendations, it only considers the user's rating of the item, ignoring that user preferences will change over time, which will also affect the recommendation effect of the recommendation system. To this end, this study introduces a time factor, taking into account the changes in user preferences over time to improve the recommendation quality of the recommendation system.

By filling the user-item rating matrix, the sparsity of the rating matrix can be significantly reduced. However, it is not clear whether the filled data is true and effective, and since the scoring of the filled data is generated by prediction, it is impossible to know the time when the scoring occurs. For this reason, it is necessary to predict the occurrence time of the filling data by setting a time window and judge whether the filling information is true and effective.

Based on the basic assumption that the interest preferences of the target users will not change in a short time, this paper makes time predictions on the fill score data of the target users. First, we determine the time span T of the target user u's score according to the scored set I_u of the target user u and then divide the time span T into i time windows of equal length according to the time span T. The specific formula is as follows:

$$T = \text{sum}\left(T_{u,i}\right). \tag{3}$$

Among them, $1 \le i \le n$, where n is a natural number greater than 0, so that the item set I_u, i in each time window $T_{u,i}$ can represent the preference of the user u in this time span. Then, by calculating the similarity between the item to be predicted and the item set $I_{u,i}$ in each time window $T_{u,i}$, the main idea is as follows: the item j that needs to be predicted is more similar to the item in the time window $T_{u,i}$, indicating that the score of item j is more likely to occur in this time window. And we use the following formula to calculate the overall similarity between project j and the set of projects in the time window. The specific formula is as follows:

$$S_{I,j} = \frac{\sum_{\text{item} \in I} S_{j,\text{item}}}{\text{Size}(I)}. \tag{4}$$

Among them, $\text{Size}(I)$ represents the number of items in the set $I_{u,i}$. The above formula shows that the project j that needs to predict the time scale value belongs to which time window is determined based on the average similarity between project j and the set of items in each time window.

In order to judge whether the filled scoring data is true and valid, we need to set a threshold δ to measure the overall similarity between the predicted item and the time window. We sort the similarity between the obtained prediction items and the time window to obtain a set of data $\text{Max}(S)$ with the largest similarity, and then judge the size of $\text{Max}(S)$ and the threshold δ. If $\text{Max}(S) \ge \delta$, it means that the scoring data will occur in this time window, and we assign the middle scale value of the time window to the scoring data; if $\text{Max}(S) \le \delta$, it means that the scoring data cannot be in this time window. When the time window occurs, the data needs to be eliminated. The setting of the threshold δ needs to change according to different recommendation systems, and the size of the threshold δ in different recommendation systems may also be different. It is necessary to determine the size of the threshold δ through repeated experiments. If the threshold δ is set too large, the predicted score data will be largely eliminated, and the user-item score matrix still has serious sparsity problems, which will affect the recommendation accuracy of the recommendation system; if the threshold δ is set too small, it will cause a large number of inaccurate prediction scores to be filled into the user-item score matrix, which will affect the recommendation quality of the recommendation system.

3.3. Performing Bipartite K-Means Clustering. In order to better simulate user preferences, the improved collaborative filtering algorithm no longer uses the user-item score matrix as the search range of the target user's nearest neighbor set but uses the method of combining tag factors with time factors and user scores to build users. The construction process of the user-tag scoring matrix is given in the following.

By filling in the scoring data and predicting the time scale value, we can already get a closer user-item scoring matrix. Considering the introduction of label factors in the collaborative filtering algorithm, it is necessary to obtain the user's preference score for each label. At the same time, the user's preference is not constant; it will change according to the change of time, so it is necessary to use the following formula to obtain the time factor of each score to represent the weight of the tag score. The specific formula is as follows:

$$f(t) = e^{-|t-1|}. \tag{5}$$

Among them, t represents the standardized time difference, and the standardized method is

$$t = \frac{\left(t_{\max} - t_l\right)}{\left(t_{\max} - t_{\min}\right)}. \tag{6}$$

Among them, $t_{\max}$ and $t_{\min}$ represent the user's most recent and earliest scoring time, respectively, and t_1 represents the time that needs to be standardized. Since the user's preferences will change over time, the closer to the most recent scoring time, the more representative the score is. For example, a user scored in January and June of this year. At this time, it is necessary to predict the user's score on a certain item in July. At this time, the user's preferences must be more similar to those in June. Standardizing the time can make the t_1 closer to the most recent rating time closer to 0 after standardization, and the time factor $f(t)$ is also closer to 1, which indicates that the rating is more representative of user preferences; after the time t_1 is standardized, the closer t is to 1, the closer the time factor $f(t)$ is to 0, which indicates that the score cannot fit the user preference well.

We use the following formula to get the user's scoring preference for each label:

$$E_{i,l} = \frac{\sum_{j=0}^{n-1} f\left(t_j\right) \cdot R_{i,j}}{n}. \tag{7}$$

Among them, $R_{i,j}$ represents the score of user i on the jth item that matches the label l, f(t_j) represents the time factor of the jth item that matches, n represents the number of items that match the label l, and $E_{\mathrm{i},\,1}$ represents the weighted average score of user i for label l.

The traditional bipartite K-means clustering algorithm can reduce the nearest neighbor search space, but it has the following shortcomings: first, the algorithm uses the Euclidean distance to calculate the distance from the point to the centroid, but the score in the user-label score matrix generally takes a value; if Euclidean distance is used as a measurement index, the difference in results will not be obvious; second, the algorithm uses Sum of Squared Error (SSE) as an index to measure the clustering effect, and it is hoped that the overall error will become smaller and smaller in each division. But due to the small value of the user-rating matrix, the SSE gap is not large, which will affect the final clustering effect.

Aiming at the traditional dichotomous K-means clustering algorithm that uses Euclidean distance as a measurement index, this paper has made improvements and uses Pearson's correlation coefficient as a measurement index. After adopting the Pearson correlation coefficient as the criterion of clustering, it is necessary to modify the measurement index of the clustering effect. After each division, the overall similarity to the cluster center is improved. That is, the following formula is used as a measure of the clustering effect:

$$D = \sum_{i=0}^{K-1} \sum_{j=0}^{N_j-1} d\left(M_i, ul_j\right). \tag{8}$$

Among them, M_i represents the cluster center of the i-th cluster, $d\,(M_i, ul_j)$ represents the similarity between user j and the cluster center to which it belongs, and D represents the overall similarity of the sample divided at the moment.

After using the dichotomous K-means algorithm to reach the user cluster that matches the target user, you can directly find the target user's nearest neighbor set in the user cluster, instead of searching in the entire user-item rating matrix, reducing the nearest neighbor search space.

The flow chart of the improved collaborative filtering algorithm is shown in Figure 4.

4. Simulation Experiment and Analysis

4.1. Similarity Method Selection Experiment. Since this article uses three similarity measurement methods, and for large-scale data, only one better measurement method needs to be selected. Therefore, first we test on large-scale data to see which similarity index is better.

The criterion for evaluating the similarity index is to see which similarity can better distinguish the difference between users. If the similarity distribution between users obtained by a certain similarity index is relatively scattered, the similarity is considered, and the indicators are relatively good.

If the similarity values between users calculated by a certain similarity index are very close or even a lot equal, it will be difficult to choose when selecting the user's similar neighbors, but if a certain similarity index is used to calculate the user's similarity between the two, the obtained similarity value is very different; it is easy to select the user's similar neighbors according to the similarity.

In order to test the similarity index, this article first randomly selects a user, might as well select the first user, and then calculates the similarity between this user and all users under three different similarity measurement methods.

Figures 5–7, respectively, show the distribution of similarity among users under the three similarity measurement methods. The ordinate in the figure represents the similarity between users, the abscissa represents the users, and each point represents the similarity between the first user and the user on the abscissa corresponding to the point.

It can be clearly seen from the above experiment that the similarity distribution between users calculated under the revised cosine and Jaccard indicators is too dense, which is not conducive to the selection to distinguish the differences between users, and the similarity index based on cosine can distinguish better. So in subsequent experiments, this article selects the cosine indicator to calculate the similarity between users.

4.2. Traditional UBCF Algorithm Test Experiment. The experiment is divided into two categories: simulated data and real data. The performance of the algorithm is verified through the recommended average absolute error, the recommended accuracy rate, and the recommended recall rate, and the results of the traditional algorithm are compared and analyzed. The effectiveness of the improved algorithm in this article is shown.

The simulated data is divided into two subcategories: traditional algorithm experiment and improved algorithm experiment in this paper. Figure 8 records the average absolute error of recommendation obtained when the traditional user-based collaborative filtering recommendation algorithm UBCF is used to recommend two users on the simulated data. The abscissa in the figure represents the number of nearest neighbors KNN, and the ordinate represents the recommended average absolute error MAE. Each point in the figure represents the error calculated according to the corresponding similarity index under the corresponding number of neighbors.

Among them, the first data represents the average absolute error calculated according to the cosine similarity when the number of neighbors KNN is 2, the second data represents the error obtained when the KNN is 3, and so on.

It can be seen from the experimental results that, on the simulated data set, the error obtained based on the modified cosine similarity index is relatively large. The main reason for this result is that the modified cosine similarity is

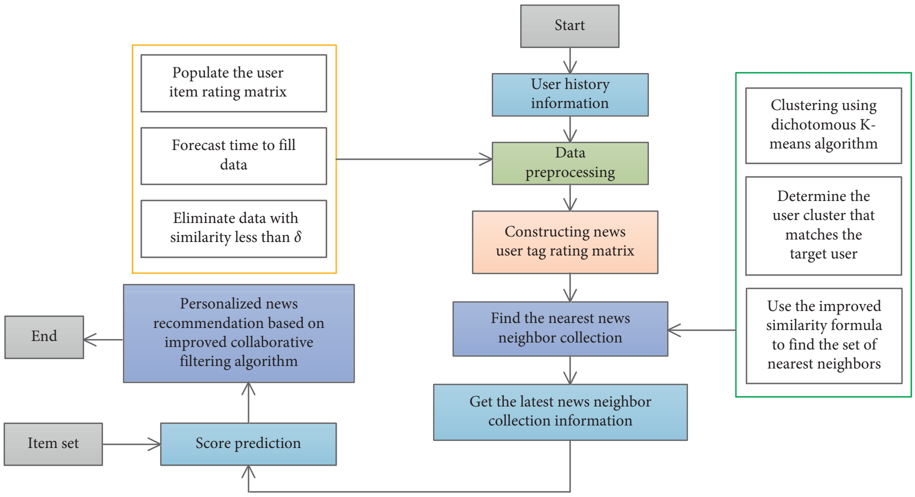
Start
User history information
Data preprocessing
Constructing news user tag rating matrix
Find the nearest news neighbor collection
Get the latest news neighbor collection information
Populate the user item rating matrix
Forecast time to fill data
Eliminate data with similarity less than δ
Clustering using dichotomous K-means algorithm
Determine the user cluster that matches the target user
Use the improved similarity formula to find the set of nearest neighbors
Personalized news recommendation based on improved collaborative filtering algorithm
End
Item set
Score prediction

Figure 4: Improved collaborative filtering algorithm flow chart.

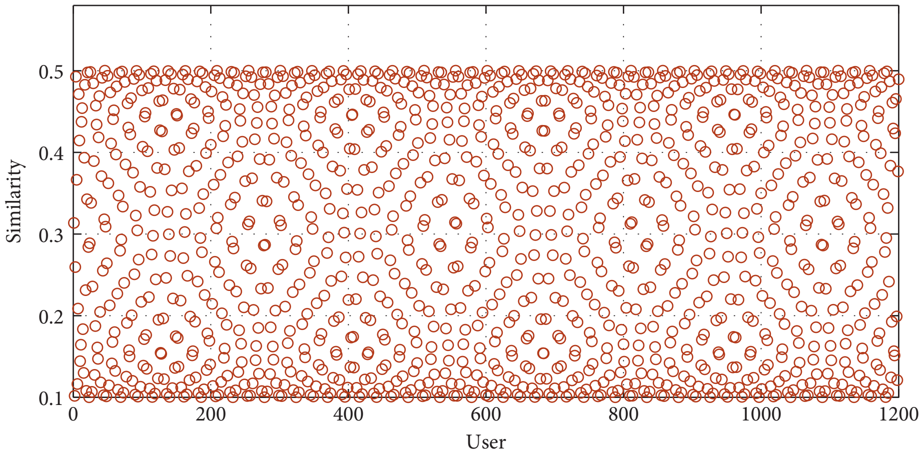

Figure 5: The distribution of similarity between users under the cosine similarity index.

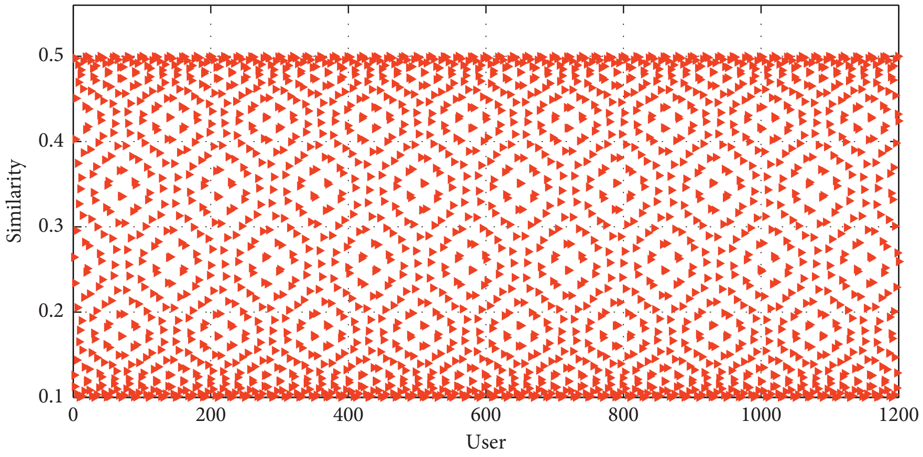

Figure 6: The distribution of similarity among users under the modified cosine index.

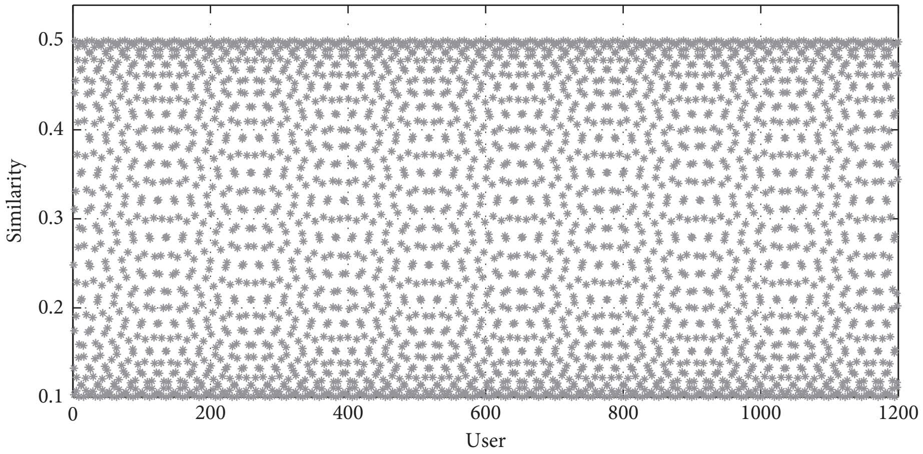

Figure 7: Distribution of similarity among users under the Jaccard similarity index.

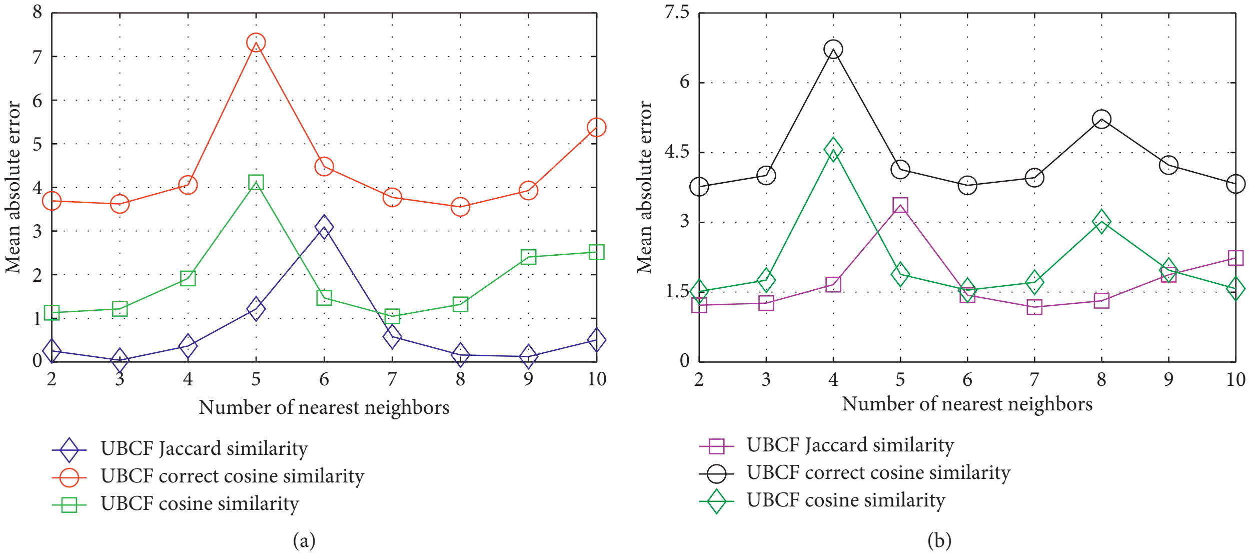

Figure 8: Experimental results of MAE based on the traditional UBCF algorithm under different similarity indicators. (a) Test result of the first user. (b) Test result of the second user.

specifically designed for hot news in the recommendation system, and its purpose is to reduce the impact of hot news on recommendation. The error obtained using the Jaccard similarity measurement index is very close to the result obtained using the cosine similarity measurement method, but the result obtained based on the Jaccard index is better overall. Based on comprehensive consideration, it is found that the traditional algorithm works better when K is 4.

4.3. Algorithm Test Experiment of This Paper. Figure 9 shows the recommended average absolute error results of the recommended collaborative filtering recommendation algorithm based on non-negative matrix factorization for two users on the simulated data set. The experiment gives the statistical results of the average absolute error obtained by the algorithm in this paper under different number of neighbors KNN and different similarity calculation methods. The values of KNN are 2, 3, 4, 5, 6, 7, 8, 9, and 10.

It can be seen from the comparative experiments in Figures 8 and 9 that the algorithm in this paper is significantly better than the traditional algorithm in the user's recommendation situation, because the average absolute error obtained by the algorithm in this paper is much smaller. The proposed algorithm based on the modified cosine similarity index has poor recommendation error results, and the results are extremely unstable, while the results obtained by using the latter two indicators have less fluctuation. From the above experiment, it can be seen that the MAE based on the Jaccard similarity measurement method is smaller. On the simulated data set, the proposed algorithm is better than the traditional UBCF algorithm.

Figures 10 and 11 show the recommendation accuracy and recall rate obtained when the algorithm in this paper

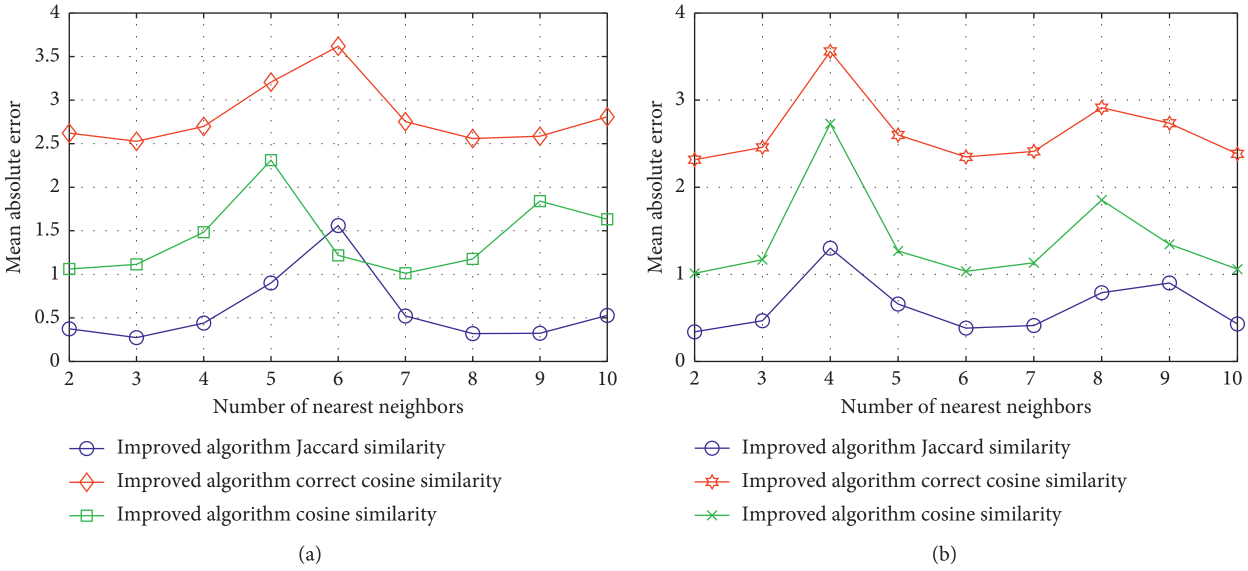

Figure 9: MAE experimental results of the NMFCF algorithm in this paper based on different similarity indicators. (a) Test result of the first user. (b) Test test result of the second user.

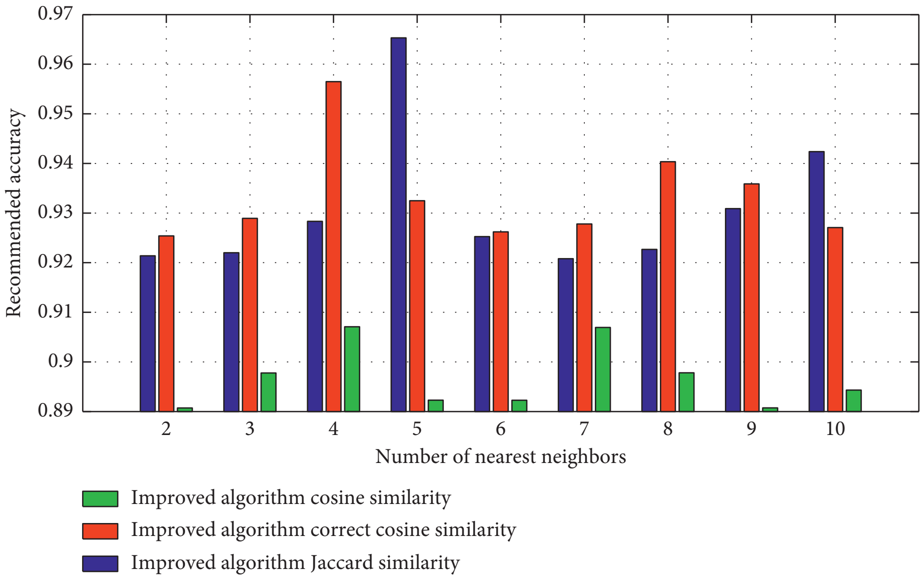

Figure 10: The accuracy of the algorithm in this paper's recommendation to the first user.

generates recommendations to users. The results in Figure 10 show that, on the simulated data set, the results obtained by the cosine similarity index are very poor. This is mainly because the cosine similarity is designed to reduce the impact of popular news on recommendations, but it does not exist in this simulated data set. Due to the relatively small number of users and news on the simulated data set, the cosine similarity index may not be suitable for such data. The algorithm in this paper uses Jaccard and modified cosine similarity index to obtain the average absolute error that is not very sensitive to the number of neighbors KNN, and the results obtained by the two are relatively similar. The experiments in Figures 10 and 11 clearly show that the algorithm proposed in this paper can obtain high recommendation accuracy and recall rates.

The above experiments show that the collaborative filtering recommendation algorithm based on nonnegative matrix factorization proposed in this paper can obtain higher recommendation accuracy and recall rate and lower average absolute error, which shows that the proposed algorithm is better than the traditional user-based collaborative filtering algorithm. On the simulated data set, the

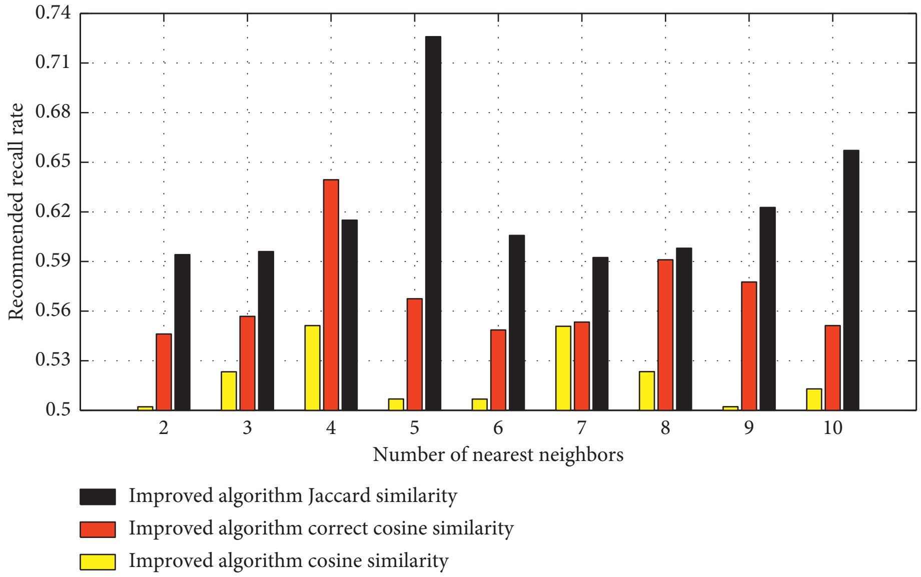

Figure 11: The recall rate of the algorithm in this paper for the recommendation of the first user.

algorithm in this paper and the traditional algorithm have better experimental results under the condition of Jaccard similarity.

5. Conclusion

This article mainly improves the defects in the collaborative filtering algorithm from two stages. In the data preprocessing stage, one is to fill the user-item rating matrix to alleviate its sparseness; the other is to introduce label factors and time factors to make the constructed user preference model have a better expression effect. In the stage of finding the nearest neighbor set, one is to combine the collaborative filtering algorithm with the dichotomous K-means algorithm and select the user cluster matching the target user as the search range of the nearest neighbor set; the second is to improve the similarity measurement formula. This paper tests the improved algorithm on the simulated data set and the real data set and verifies the effectiveness of the algorithm proposed in this paper from three aspects: the average absolute error of recommendation, the recommendation accuracy rate, and the recommendation recall rate. Based on the comparison of user-based collaborative filtering recommendation algorithms, experiments have proved that the algorithm proposed in this paper can effectively solve the problem of data sparsity and improve the recommendation effect of the algorithm.

In subsequent research, you can try to combine other clustering methods in the clustering algorithm with the collaborative filtering algorithm, which may produce better recommendation results. On the other hand, compared with the traditional collaborative filtering algorithm, the improved collaborative filtering algorithm takes into account the label factor and the time factor, which makes the division of the nearest neighbor set more accurate, but this method is based on the user in the nearest neighbor set. It may not fit the target user's preferences accurately. In the future, the user's own data can be used to train the neural network and build the user's own preference model for each user, which can make the recommendation results more accurate.

Data Availability

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

Conflicts of Interest

The author declares that there are no conflicts of interest regarding this paper.

References

[1] A. Chakraborty, S. Ghosh, N. Ganguly, and K. P. Gummadi, "Optimizing the recency-relevance-diversity trade-offs in non-personalized news recommendations," *Information Retrieval Journal*, vol. 22, no. 5, pp. 447–475, 2019.

[2] M. He, X. Wu, and J. Zhang, "UP-TreeRec: building dynamic user profiles tree for news recommendation," *China Communications*, vol. 16, no. 4, pp. 219–233, 2019.

[3] F. Wu, C. Wu, and M. I. An, "Personalized news recommendation based on deep learning," *Nanjing Xinxi Gongcheng Daxue Xuebao*, vol. 11, no. 3, pp. 278–285, 2019.

[4] L. Zhang, P. Liu, and J. A. Gulla, "Dynamic attention-integrated neural network for session-based news recommendation," *Machine Learning*, vol. 108, no. 10, pp. 1851–1875, 2019.

[5] M. Asenova and C. Chrysoulas, "Personalized micro-service recommendation system for online news," *Procedia Computer Science*, vol. 160, pp. 610–615, 2019.

[6] B. K. Ye, Y. J. T. Tu, and T. P. Liang, "A hybrid system for personalized content recommendation," *Journal of Electronic Commerce Research*, vol. 20, no. 2, pp. 91–104, 2019.

[7] C.-Y. Lin and H.-S. Chen, "Personalized channel recommendation on live streaming platforms," *Multimedia Tools and Applications*, vol. 78, no. 2, pp. 1999–2015, 2019.

[8] L. Bai, L. Liu, and S. Song, "NCR-KG: news community recommendation with knowledge graph," *CCF Transactions on Pervasive Computing and Interaction*, vol. 1, no. 4, pp. 250–259, 2019.

[9] Z. A. Xu, Z. Fan, and W. Jiang, "Connectionist recommendation in the wild: on the utility and scrutability of neural networks for personalized course guidance," *User Modeling and User-Adapted Interaction*, vol. 29, no. 2, pp. 487–525, 2019.

[10] P. Lv, X. Meng, and Y. Zhang, "BoRe: adapting to reader consumption behavior instability for news recommendation," *ACM Transactions on Information Systems (TOIS)*, vol. 38, no. 1, pp. 1–33, 2019.

[11] B. Bodó, "Selling news to audiences-a qualitative inquiry into the emerging logics of algorithmic news personalization in european quality news media," *Digital Journalism*, vol. 7, no. 8, pp. 1054–1075, 2019.

[12] A. Darvishy, H. Ibrahim, F. Sidi, and A. Mustapha, "A customized non-exclusive clustering algorithm for news recommendation systems," *Journal of University of Babylon for Pure and Applied Sciences*, vol. 27, no. 1, pp. 368–379, 2019.

[13] S. P. Perumal, G. Sannasi, and K. Arputharaj, "An intelligent fuzzy rule-based e-learning recommendation system for dynamic user interests," *The Journal of Supercomputing*, vol. 75, no. 8, pp. 5145–5160, 2019.

[14] Y. Song, N. Sahoo, and E. Ofek, "When and how to diversify-a multicategory utility model for personalized content recommendation," *Management Science*, vol. 65, no. 8, pp. 3737–3757, 2019.

[15] Y. Tang, K. Guo, R. Zhang, T. Xu, J. Ma, and T. Chi, "ICFR: an effective incremental collaborative filtering based recommendation architecture for personalized websites," *World Wide Web*, vol. 23, no. 2, pp. 1319–1340, 2020.

[16] X. Wang, C. Gao, J. Ding, Y. Li, and D. Jin, "CMBPR: category-aided multi-channel bayesian personalized ranking for short video recommendation," *IEEE Access*, vol. 7, pp. 48209–48223, 2019.

[17] Y. Qian, "Application of collaborative filtering algorithm in mathematical expressions of user personalized information recommendation," *International Journal of Computational Intelligence Systems*, vol. 12, no. 2, pp. 1446–1453, 2019.

[18] Z. Ji, H. Pi, W. Wei, B. Xiong, M. Wozniak, and R. Damasevicius, "Recommendation based on review texts and social communities: a hybrid model," *IEEE Access*, vol. 7, pp. 40416–40427, 2019.

[19] M. Svrcek, M. Kompan, and M. Bielikova, "Towards understandable personalized recommendations: hybrid explanations," *Computer Science and Information Systems*, vol. 16, no. 1, pp. 179–203, 2019.

[20] G. Alexandridis, A. Chrysanthi, G. E. Tsekouras, and G. Caridakis, "Personalized and content adaptive cultural heritage path recommendation: an application to the Gournia and Çatalhöyük archaeological sites," *User Modeling and User-Adapted Interaction*, vol. 29, no. 1, pp. 201–238, 2019.