Hindawi
Complexity
Volume 2021, Article ID 6635202, 11 pages
https://doi.org/10.1155/2021/6635202

Research Article

Research on Hybrid Collaborative Filtering Recommendation Algorithm Based on the Time Effect and Sentiment Analysis

Xibin Wang,[1,2] **Zhenyu Dai**,[3] **Hui Li**,[3] **and Jianfeng Yang**[1,2]

[1]*School of Data Science, Guizhou Institute of Technology, Guiyang 550003, Guizhou, China*
[2]*Special Key Laboratory of Artificial Intelligence and Intelligent Control of Guizhou Province, Guiyang 550003, Guizhou, China*
[3]*College of Computer Science & Technology, Guizhou University, Guiyang 550025, Guizhou, China*

Correspondence should be addressed to Hui Li; cse.huili@gzu.edu.cn

Received 14 December 2020; Revised 19 February 2021; Accepted 22 February 2021; Published 8 March 2021

Academic Editor: Wei Wang

In this study, we focus on the problem of information expiration when using the traditional collaborative filtering algorithm and propose a new collaborative filtering algorithm by integrating the time factor (ITWCF). This algorithm considers information influence attenuation over time, introduces an information retention period based on the information half-value period, and proposes a time-weighted function, which is applied to the nearest neighbor selection and score prediction to assign different time weights to the scores. In addition, to further improve the quality of the nearest neighbor selection and alleviate the problem of data sparsity, a method of calculating users' sentiment tendency by analysis of user review features is proposed to mine users' attitudes about the reviewed items, which expands the score matrix. The time factor and sentiment tendency are then integrated into the K-means clustering algorithm to select the nearest neighbor. A hybrid collaborative filtering model (TWCHR) based on the improved K-means clustering algorithm is then proposed, by combining item-based and user-based collaborative filtering. Finally, the experimental results show that the proposed algorithm can address the time effect and sentiment analysis in recommendations and improve the predictive performance of the model.

1. Introduction

E-commerce has gradually developed into social commerce with the rapid development of the Internet. Users can publish through and obtain information from an increasing number of channels. Recommendation technology, based on the notion of collaborative filtering, helps users better utilize information [1]. Both practically and theoretically, collaborative filtering recommendation algorithms based on users or items have achieved good results. However, problems such as data sparsity, cold start, and information expiration still occur [2, 3]. Researchers have thus proposed various improvements to achieve higher quality predictions and recommendations.

The most common method for addressing the problem of data sparsity is to use dimension reduction techniques to compress the original data [4]. In terms of the cold start problem, Guo et al. [5] noted that, in early stages, neighbors can be found according to user or item features. To address the problem of information expiration, the nonlinear forgetting function was introduced, which considers the loss of information influence over time, and a series of recommendation algorithms incorporating this function have been proposed [3, 6–9].

Most e-commerce websites feature online reviews representing users' specific feedback on products, but the issue of data sparsity is still a major challenge. Researchers have found that mining the users' sentiment tendencies from review information will improve user preference models and recommendation accuracy [10–12]. Thus, the mining of reviews to establish users' interest preferences, combined with user scores to improve the traditional collaborative filtering algorithm, has recently become a topic of great interest. Ganu et al. [13] proposed a multilabel text classifier based on the support vector machine (SVM) to classify reviews and to generate text scores and recommendations based on them. However, this method requires the topic category and sentiment classification of 3400 sentences to be manually annotated. McAuley and Leskovec [14] proposed the hidden factors as topics (HFT) model by combining the latent-factor model with the document topic generation model (Latent Dirichlet allocation, LDA). The LDA method is used to obtain the product review topic distribution, which is then combined with the latent-factor model to establish the relationship between the review topic and the score. This method regards the review topic distribution as consistent with the potential score dimension and thus establishes the transformation. Dehkordi et al. [15] added user reviews and users with similar preferences as implicit feedback into the collaborative filtering algorithm to improve the accuracy of the recommendation results.

The abovementioned improved algorithms are to some extent able to solve the problems faced by traditional collaborative filtering algorithms, but the recommendation quality could still be further improved by effectively solving the problems of information expiration and sparsity. In this study, we address information expiration by first introducing an improved time-weighted collaborative filtering algorithm (ITWCF), which assumes that although the influence of information is nonlinearly attenuated with the passage of time, it will not change significantly within a specific period. The time window in which the information remains unchanged is integrated into the attenuation function. By applying the information half-value period [7] and the proposed concept of a period of information retention, an improved time-weighted function is generated in this study and introduced into the traditional similarity calculation to improve its accuracy, with the aim of achieving better recommendation results. In addition, to make better use of user review information and alleviate the problem of data sparsity, a sentiment tendency calculation method based on review features' analysis is proposed and applied to clustering analysis, which improves the quality of the nearest neighbor selection. Thus, a hybrid recommendation model is proposed by combining item-based and user-based collaborative filtering. The experimental results show that the proposed method can effectively account for the time factor and user sentiment tendency, thus improving the performance of the recommendation model.

2. Related Work

2.1. User-Based Collaborative Filtering. In user-based collaborative filtering, it is assumed that the target user will like items that are similar to his or her interests and preferences. This similarity is calculated based on the score set of users' items, and those that are more similar to the target user make a greater contribution to the predicted score. Currently, the most common similarity calculation methods include cosine similarity, the Pearson correlation coefficient, and the adjusted cosine [16].

(1) Cosine similarity regards each user's historical score as an n-dimensional vector, where the vectors u and v represent the historical scores of users u and v, respectively. Here, the ith element of the vector is the user's score for the ith item, and an unrated item is represented by 0. The cosine similarity of users u and v can be expressed by the cosine of the angle between the two vectors, that is,

$$\operatorname{sim}(u,v)=\cos(u,v)=\frac{\vec{u}\cdot\vec{v}}{\|\vec{u}\|\cdot\|\vec{v}\|}=\frac{\sum_{i\in I_{uv}} r_{ui}\cdot r_{vi}}{\sqrt{\sum_{i\in I_u} r_{ui}^2}\cdot\sqrt{\sum_{i\in I_v} r_{vi}^2}}, \tag{1}$$

where r_{ui} is the score of user u for item i and I_{uv} is the set of items rated by users u and v.

(2) Pearson similarity describes the degree of consistency between two users' rating trends on several items in user-based collaborative filtering. The calculation method is as follows:

$$\operatorname{sim}(u,v)=\frac{\sum_{i\in I_{uv}}\left(r_{ui}-\bar{r_u}\right)\cdot\left(r_{vi}-\bar{r_v}\right)}{\sqrt{\sum_{i\in I_{uv}}\left(r_{ui}-\overline{R_u}\right)^2\cdot\sqrt{\sum_{i\in I_{uv}}\left(r_{vi}-\overline{r_v}\right)^2}}}, \tag{2}$$

where $\overline{r_u}$ is the average score of user u on the items.

(3) In modified cosine similarity, all users have different rating preferences. To correct the deviation of different users' rating scales, the user's average score is subtracted. The modified cosine similarity calculation method is as follows:

$$\operatorname{sim}(u,v)=\frac{\sum_{i\in I_{uv}}\left(r_{ui}-\bar{r_u}\right)\cdot\left(r_{vi}-\bar{r_v}\right)}{\sqrt{\sum_{i\in I_{u}}\left(r_{ui}-\overline{r_u}\right)^2\cdot\sqrt{\sum_{i\in I_{v}}\left(r_{vi}-\overline{r_v}\right)^2}}}. \tag{3}$$

(4) Prediction score and recommendation calculates the similarity of users, where the nearest neighbor set $N_{(u)}$ of target user u is obtained according to the Top-N algorithm, and then, the prediction score of user u for the unrated item i is

$$P\left(r_{ui}\right)=\overline{r_u}+\frac{\sum_{v\in N_{(u)}}\operatorname{sim}(u,v)\left(r_{v,i}-\bar{r_v}\right)}{\sum_{v\in N_{(u)}}\operatorname{sim}(u,v)}. \tag{4}$$

All unrated items of target user u are predicted based on the above method, and the Top-N items with the highest predicted scores are selected and recommended to target user u.

2.2. Item-Based Collaborative Filtering. The unrated item I_i of user u can be taken as an example. First, the similarity between the target item I_i and other items in set I is calculated. The k items with the highest similarity are then removed to form the nearest neighbor set $N_{(i)}=\{I_1, I_2, \ldots, I_k\}$ of item I_i. Similarity can be calculated using various methods, and the most basic of which are cosine similarity, Pearson correlation similarity, and modified cosine similarity [16].

(1) The cosine similarity method is similar to the angle between two score vectors. The smaller the angle, the higher the similarity (see formula (5)). If the score in the matrix r is null, it takes the value of 0:

$$\operatorname{sim}(i, j) = \cos(i, j) = \frac{\vec{i} \cdot \vec{j}}{\| \vec{i} \| \cdot \| \vec{j} \|} = \frac{\sum_{u \in U_{ij}} r_{ui} \cdot r_{uj}}{\sqrt{\sum_{u \in U_i} r_{ui}^2} \cdot \sqrt{\sum_{u \in U_j} r_{uj}^2}}, \tag{5}$$

where r is the vector space, $\vec{i}$ and $\vec{j}$ are the score vectors of all users on items i and j, respectively, and U_{ij} is the set of users who have rated items i and j.

(2) Pearson similarity is used to measure the linear correlation between two score vectors and is calculated as follows:

$$\operatorname{sim}(i, j) = \frac{\sum_{u \in U_{ij}} \left(r_{ui} - \bar{r}_i\right) \cdot \left(r_{uj} - \bar{r}_j\right)}{\sqrt{\sum_{u \in U_{ij}} \left(r_{ui} - \overline{R_i}\right)^2 \cdot \sqrt{\sum_{u \in U_{ij}} \left(r_{uj} - \overline{r_j}\right)^2}}}, \tag{6}$$

where $\overline{r_i}$ and $\overline{r_j}$ are the average scores of all users for items i and j, respectively.

(3) Modified cosine similarity addresses a major shortcoming of using the basic cosine method to measure similarity. Specifically, the basic cosine method ignores the differences among users in their understanding of the rating criteria. To address this, the modified cosine similarity is calculated after subtracting the average score of the corresponding user from each score. The details are as follows:

$$\operatorname{sim}(i, j) = \frac{\sum_{u \in U_{ij}} \left(r_{ui} - \bar{r}_u\right) \cdot \left(r_{uj} - \bar{r}_u\right)}{\sqrt{\sum_{u \in U_{ij}} \left(r_{ui} - \overline{r_u}\right)^2 \cdot \sqrt{\sum_{u \in U_{ij}} \left(r_{uj} - \overline{r_u}\right)^2}}}. \tag{7}$$

(4) In prediction score and recommendation, after obtaining the nearest neighbor set $N_{(i)}$ of item I_i, the score of user u on I_i can be predicted based on the score of target user u for the items in $N_{(i)}$:

$$P\left(r_{ui}\right) = \bar{r}_i + \frac{\sum_{j \in N_{(i)}} \operatorname{sim}(i, j)\left(r_{uj} - \bar{r}_j\right)}{\sum_{j \in N_{(i)}} |\operatorname{sim}(i, j)|}. \tag{8}$$

All unrated items of target user u are predicted according to the above method, and the Top-N items with the highest predicted scores are selected and recommended to target user u.

3. Collaborative Filtering Based on Time Effect

3.1. Time Effect Analysis of Item Score. The traditional collaborative filtering recommendation algorithm does not consider changes over time. In reality, however, the contribution of item scores to recommendations does vary over time, so the time effect should be considered in the recommendation [17]. In general, users are more interested in the most recently selected item than those selected earlier. However, when calculating the neighbor set, the traditional algorithm treats the item scores of different time periods equally, which means that the neighbor set of the target user may not include the nearest neighbor in a true sense, thus reducing the recommendation accuracy.

Calculating the set of neighbors based on user item scores in the same or a similar time period is more accurate. An example illustrating this is presented in Table 1, in which the score records of 4 users (*User*, denoted by u) correspond to 3 time periods for 5 items.

Assume that $User_1$ is the target user, there are only three neighbor users, and it is necessary to predict $User_1$'s score on $Item_3$. The time difference between t_1 and t_2 is relatively small, but the difference from t_4 is relatively large. According to the traditional similarity calculation method, the nearest neighbors of $User_1$ are $sim\ (u_1, u_2) > sim(u_1, u_3) > sim\ (u_1, u_4)$. If the time effect is considered, then the weight of the recommended contribution should be increased for the more recent time periods. In this case, the nearest neighbors of $User_1$ are $sim\ (u_1, u_2) > sim(u_1, u_4) > sim\ (u_1, u_3)$. Thus, the traditional method is unable to appropriately judge the nearest neighbor of $User_1$.

3.2. Improved Time-Weighted Function. In reference [3], the nonlinear exponential forgetting function is used to describe the attenuation degree of information, and the time-weighted function Tvalue (t) is proposed, which reflects the different contributions of the scores at different times toward the recommendation. To describe the process of information from release to decay and finally its disappearance, the concept of the information half-value period is proposed in [7].

The definition of this information half-value period T_s is the time it takes for the information released to halve its influence, that is, after time T_s, the influence of the information is halved. Thus, it can be described as follows:

$$\text{Tvalue}\left(T_s\right) = 0.5 \times \text{Tvalue}(0). \tag{9}$$

From the above formula, after time T_s, the time-weighted function becomes 0.5, that is, the reference value of the user's score becomes half of the original. We then define the attenuation factor γ as follows:

$$\gamma = \frac{(\ln 0.5)}{T_s}. \tag{10}$$

The time-weighted function T value (t) can thus be calculated as follows:

$$\text{Tvalue}(t) = e^{\gamma \cdot t}, \tag{11}$$

where $t = t_{\text{now}} - t_{ui}$, t_{ui} is the rating time of the item i by user u, and the value of Tvalue (t) is the time-weighted value, that is, the degree of attenuation of the information. The value of

Table 1: User-item scores in different time periods.

Item	*Item*$_1$	*Item*$_2$	*Item*$_3$	*Item*$_4$	*Item*$_5$
User$_1$ (t_1)	5	4	?	3	4
User$_2$ (t_4)	4	3	4	3	4
User$_3$ (t_3)	3	4	3	4	3
User$_4$ (t_1)	2	4	5	2	2

this function is kept in (0, 1], and it decreases with the increase of time $t_{u,i}$, which indicates that the user's recent rating records have more predictive value.

The influence of information generally shows a nonlinear decline, but within a certain period, it does not change significantly. Thus, we introduce the concept of an information retention period.

The definition of information retention period T^p is the time period in which the influence of information remains constant.

Introducing the information retention period gives an improved time-weighted function F value (t):

$$\text{Fvalue}(t) = e^{\gamma\cdot T^p\cdot\lceil t/T^p\rceil}, \tag{12}$$

where $\gamma = (\ln 0.5)/T_s$, $t = t_{\text{now}} - t_{ui}$, and t_{ui} is the rating time of the item i by user u.

Adding the concept of the information retention period to the improved time-weighted function is equivalent to introducing a time window, in which the information remains basically unchanged, into the original weighted function. This leads to a gradient of exponential attenuation of the information, which is more in line with reality.

3.3. Improved Similarity Calculation Methods. In traditional cosine similarity calculations, the improved time-weighted function is introduced to assign a time weight to each score.

(1) The improved calculation method of item-based similarity measurement is as follows:

$$\text{Tsim}(i, j) = \frac{\sum_{u\in U_{ij}} r_{ui}\cdot\text{Fvalue}(\Delta t_{ui})\cdot r_{uj}\cdot\text{Fvalue}(\Delta t_{uj})}{\sqrt{\left(\sum_{u\in U_{ij}} r_{ui}\cdot\text{Fvalue}(\Delta t_{ui})\right)^2}\times\sqrt{\left(\sum_{u\in U_{ij}} r_{uj}\cdot\text{Fvalue}(\Delta t_{uj})\right)^2}}, \tag{13}$$

where *Tsim* (i, j) is the similarity of items i and j, r_{ui} and r_{uj} are user u's scores on items i and j, respectively, Fvalue ($\Delta t_{u,i}$) is the time-weighted function, and $\Delta t_{ui} = t_{now}$-t_{ui} is the interval between the rating time of item i and the current time.

(2) The improved user-based similarity measurement calculation method is as follows:

$$\text{Tsim}(u, v) = \frac{\sum_{i\in I_{uv}} r_{ui}\cdot\text{Fvalue}(\Delta t_{ui})\cdot r_{vi}\cdot\text{Fvalue}(\Delta t_{vi})}{\sqrt{\sum_{i\in I_u}(r_{ui}\cdot\text{Fvalue}(\Delta t_{ui}))^2}\cdot\sqrt{\sum_{i\in I_v}(r_{vi}\cdot\text{Fvalue}(\Delta t_{vi}))^2}}, \tag{14}$$

where *Tsim* (u, v) is the similarity between users u and v.

(3) The improved user-based score prediction is as follows:

$$P_{\text{user}}(r_{ui}) = \bar{r_u} + \frac{\sum_{v\in N_{(u)}}\text{Tsim}(u, v)\cdot\text{Fvalue}(\Delta t_{vi})(r_{vi} - \bar{r_v})}{\sum_{v\in N_{(u)}}|\text{Tsim}(u, v)|\cdot F(\Delta t_{vi})}, \tag{15}$$

where P (r_{ui}) is user u's predicted score for item i, $N_{(u)}$ is the set of nearest neighbors of user u, and $\overline{r_u}$ and $\overline{r_v}$ are the average scores of users u and v in the entire item set.

(4) The improved item-based score prediction is as follows:

$$P_{\text{item}}(r_{ui}) = \overline{r_i} + \frac{\sum_{j\in N_{(i)}}\text{Tsim}(i, j)\cdot\text{Fvalue}(\Delta t_{uj})(r_{ui} - \bar{r_j})}{\sum_{j\in N_{(i)}}|\text{Tsim}(i, j)|\cdot\text{Fvalue}(\Delta t_{uj})}, \tag{16}$$

where $P(r_{ui})$ is user u's predicted score for item i, $N_{(i)}$ is the set of nearest neighbors of item i, and $\overline{r_i}$ and $\overline{r_j}$ are the average scores of items i and j in the entire user set.

3.4. Collaborative Filtering Algorithm Integrating the Time Effect. Based on the improvements of the similarity calculation method and the score prediction method, an improved time-weighted collaborative filtering (ITWCF) algorithm is proposed (Algorithm 1).

4. Sentiment Analysis of Review Information

The sentiment tendency analysis conducted in this study is aimed at expanding the score matrix, so quantitative analysis results are required. In our analysis, using neutral sentiment as the reference, we assess the sentiment deviation tendency (deviation intensity) of reviews, which establishes the polarity intensity and enables the results to be quantified. A score matrix is finally constructed according to the calculated sentiment values, which can enable score prediction.

4.1. Review Data Preprocessing. A review sentence is generally composed of subjective and objective clauses. Objective clauses are not useful for analyzing sentiment tendency, so they must be deleted. A method based on [18] is used to analyze the type of clause and retain the subjective clauses. The word segmentation tool ICTCLAS is used to segment and label each review sentence. To make the analysis more effective, any information that is inconsistent in terms of the review topic is also manually labeled in advance, and a UTF-8 stop words' table is used to remove the stop words.

4.2. Feature Extraction and Sentiment Analysis of Review Information. Assume that the set of reviews is Review_D, and all feature words in the reviews are $F = \{f_1, f_2, \ldots, f_n\}$.

Step 1. Utilize the Chinese word segmentation tool ICTCLAS to output all adjectives and adverbs: $F_{AA} = \{f_1, f_2, \ldots, f_{AA}\}$.

Step 2. Adopt the IKAnalyzer to segment each review, and calculate the corpus frequency-inverse document frequency (CF-IDF) value w of all feature words. The calculation method is shown as formula (17), and feature words $F_{\text{CF-IDF}} = \{f_1, f_2, \ldots, f_{\text{CF-IDF}}\}$ are selected:

$$\text{CF} - \text{IDF}_i = \text{CF}_i \times \text{IDF}_i = \frac{f_i}{\sum_{i=1}^{n} f_i} \times \log \frac{|\text{Review}_D|}{\left|\{j | t_i \in d_j\}\right|}, \tag{17}$$

where f_i is the word frequency of the ith word in the entire corpus, $|\text{Review}_\text{D}|$ is the number of review texts in corpus Review_D, and $|\{j|t_i \in d_j\}|$ is the number of review texts containing the ith word in the corpus.

Step 3. Obtain the sentiment words in the reviews, including adjectives, adverbs, verbs, and nouns:

$$F = F_{AA} \cup F_{\text{CF-IDF}} = \{f_1, f_2, \ldots, f_n\}. \tag{18}$$

Step 4. Merge the features. Different words are often used to describe the same feature f in reviews, and thus, if the features are not merged, major deviations may occur in the analysis. We use the sentiment lexicon based on HowNet [19] and the point mutual information (PMI) method [20] to determine the semantic similarity between sentiment feature words (the calculation formula is (19)). When similarity reaches the set threshold, the features are merged:

$$\text{Sim}(f_i, f_j) = \log_2 \left(\frac{P(f_i, f_j)}{P(f_i) P(f_j)} \right), \tag{19}$$

where $\text{Sim}(f_i, f_j)$ is the similarity of the features f_i and f_j, $P(f_i, f_j)$ is the probability that the features f_i and f_j appear together, $P(f_i)$ is the probability that the feature f_i is included in the review, and $P(f_j)$ is the probability that the feature f_j is included.

Step 5. Calculate the sentiment tendency of the feature words. For the feature word f_i, based on formula (19), the details are as follows:

$$\text{Tendency}(f_i) = \sum_{\text{Pword} \in \text{PosWords}} \text{Sim}(f_i, \text{Pword}) - \sum_{\text{Nword} \in \text{NegWords}} \text{Sim}(f_i, \text{Nword}), \tag{20}$$

where *PosWords* and *NegWords* are the sets of HowNet positive and negative sentiment words, respectively.

If $\text{Tendency}(f_i) > 0$, the feature f_i is a positive sentiment word and is denoted as positive once.

If $\text{Tendency}(f_i) < 0$, the feature f_i is a negative sentiment word and is denoted as negative once.

If $\text{Tendency}(f_i) = 0$, the feature f_i is a neutral sentiment word.

Input. Target user u; information half-value period T_s; information retention period T^p.
Output. A list of n items recommended for target user u.
Step 1. Set the items that target user u has not rated as the target item set I_{ua}.
Step 2. Use formula (12) to calculate the time-weighted values of scores in r to obtain the score-weighted matrix.
Step 3. Use formula (13) to calculate the similarity between target item $i \in I_{ua}$ and other items, and select the most similar k items to form the nearest neighbor set of target item i. Use formula (14) to calculate the similarity between target user u and other users, and based on this similarity, the most similar k users are selected to form the nearest neighbor set of target user u.
Step 4. Use formulas (15) and (16) to predict the scores $P_{user}\ (r_{ui})$ and $P_{item}\ (r_{ui})$ of target user u for item i, respectively.
Step 5. Repeat *Steps 3* and *4* for all items in I_{ua} to predict the scores of all unrated items. Then, calculate the weighted scores. Finally, recommend the top n items with the highest predicted $P\ (r_{ui})$ values for target user u.

Algorithm 1: The ITWCF algorithm.

Input. User rating matrix r; set i of n items; m users U; sentiment tendency value *Pol* of users toward items; the number of clusters K; and, the similarity threshold η.
Output. Clustering number K and K cluster centers.
Step 1. Select the input data. Extract set I of n items and set U of m users from r.
Step 2. Calculate the time-weighted scores, and integrate the sentiment tendency. Set the half-value period T_s and the retention period T^p, and use formula (12) to calculate the time-weighted value F value (Δt_{ui}) of each score r_{ui} to form a time-weighted score, as shown in the following: $r'_{ui} = r_{ui} \times \text{Fvalue}\ (\Delta t_{ui}) \times \text{Pol}^{\cdot}$
Step 3. Randomly select cluster centers. Randomly select the time-weighted scores of K items as the initial cluster centers, denoted as $C = \{C_u^1, C_u^2, \ldots, C_u^k\}$. Here, each cluster center $C_u^j \in C$ corresponds to a cluster, denoted as C^j.
Step 4. Calculate the similarity between the item and the cluster center. Use formula (22) to calculate the similarity between each item $i \in I$ and the cluster center $C_u^j \in C$, and the first s items that are greater than the similarity threshold η are put into cluster C^p corresponding to the most similar cluster center C_u^p.$\text{sim}\,(i, C_u^j) = (\sum_{u\in U_i} r'_{u,i} \times r_{uC_u^j} / \sqrt{(\sum_{u\in U_i} r'_{ui})^2} \times \sqrt{(\sum_{u\in U_i} r_{uC_u^j})^2}\,)^{\cdot}$
Step 5. Update the cluster centers. Calculate the new center for each cluster, that is, update the cluster center vector after adding new items in one iteration.
Step 6. The algorithm is terminated, and the result is output. Repeat *Step 4* and *Step 5* until the cluster centers no longer change and convergence is reached. The clustering number K and cluster centers are thus obtained.

Algorithm 2: A K-means clustering algorithm based on time effect and sentiment analysis.

Step 6. Calculate the sentiment tendency value of the review sentence. Extract the feature words of each review sentence and the corresponding number of favorable reviews, and then, calculate the sentiment tendency value of the whole review sentence. The calculation method is shown in the following formula:

$$\text{Pol} = \frac{\sum_{i=1}^{N} \text{Tendency}\left(f_i\right)}{N}, \tag{21}$$

where N is the total number of features in the review sentence and $\text{Tendency}\left(f_i\right)$ is the sentiment tendency of the feature f_i.

5. Clustering Algorithm Based on Time Effect and Sentiment Analysis

As mentioned, the influence of information will decay over time. In this study, the improved time-weighted function F value (t) is applied to the item clustering. In addition, because the sentiment attitude of the user review item is a direct expression of the user's behavior, making full use of the sentiment tendencies of users can lead to improved adaptation to their personalized needs. Thus, the ITWCF algorithm can be optimized by clustering analysis, and therefore, an item clustering algorithm combining sentiment analysis and time-weighted function is proposed (Algorithm 2).

6. Hybrid Collaborative Filtering Model

To ensure that the recommendation algorithm has the features of item-based and user-based collaborative filtering, a hybrid collaborative filtering model based on both is proposed, which effectively improves the recommendation accuracy after clustering by the K-means algorithm.

6.1. Hybrid Model Construction

Step 1. Use n-fold cross-validation to predict and generate the training data for *Step 2*. R is defined as the user's original score matrix, and the whole original score set is randomly divided into n equal parts. The sth part is denoted as R_s (training set), and $\overline{R_s}$ (test set) is used to denote other rating data except R_s in the score matrix. P_{user} is the user-based score prediction function, and P_{item} the item-based score prediction function. Utilize formula (22) to construct the training data of *Step 2*:

Input. User score matrix r; set I of n items; m users U; clustering number K; and, similarity threshold η.
Output. A list of n items recommended for target user u.
Step 1. Substitute the half-value period T_s and retention period T^p in formula (12) to obtain the time-weighted values of each item's score.
Step 2. Utilize Algorithm 2 to cluster the score matrix and obtain the K clusters, namely, the nearest neighbor set.
Step 3. Use the hybrid collaborative filtering model to calculate score $P\,(r_{ui})$.
Step 4. Predict the scores of all items in the set I, sort them according to the score level, and then recommend the top n items to target user u.

Algorithm 3: TWCHR algorithm.

$$P_{\text{next}}(r_{ui}) = \{P^s_{\text{user}}(r_{ui}), P^s_{\text{item}}(r_{ui}), (r_{ui}) | r_{ui} \in R_s, s = 1, 2, \ldots, n\}, \tag{22}$$

where $P_{\text{next}}\,(r_{ui})$ is the predicted value, $P^s_{\text{user}}\,(r_{ui})$ is the predicted value of r_{ui} based on the user's prediction score function, and $P^s_{\text{item}}\,(r_{ui})$ is the predicted value of r_{ui} based on the score prediction function of the item.

Step 2. Conduct a weighted fusion of item-based and user-based prediction functions based on the training data generated in *Step 1*, and formula (23) is the fused prediction model:

$$P(r_{ui}) = \beta_1 \times P_{\text{user}}(r_{ui}) + \beta_2 \times P_{\text{item}}(r_{ui}), \tag{23}$$

where $P\,(r_{ui})$ is the predicted score after weighted fusion and β_1 and β_2 are the weights of item-based and user-based predicted values, respectively.

6.2. Hybrid Model Solution. The above problem is transformed into a quadratic optimization problem with constraints, referred to as an objective function, and the details are as follows:

$$\min_{\beta} \frac{1}{2} \sum_{j=1}^{|\text{next}|} \left(\beta^T x_k - r_k\right)^2, \tag{24}$$
$$\text{s.t.} \quad \beta_1 \geq 0, \beta_2 \geq 0,$$

where β is the parameter of the model, $\beta = (\beta_1, \beta_2)^T$, $|\text{next}|$ is the size of the training set, (x_k, r_k) is the kth training sample, and $x_k = (P_{\text{user}}(r_k), P_{\text{item}}(r_k))^T$.

Use the Lagrangian multiplier and the KT condition to solve the optimization problem. Set $J(\beta) = (1/2) \sum_{j=1}^{|\text{next}|} (\beta^T x_k - r_k)^2$, and the derivative of β can be obtained:

$$\begin{cases} \beta_1 = \dfrac{gp - fh}{g^2 - ef} \\ \\ \beta_2 = \dfrac{gh - eg}{g^2 - ef} \end{cases}, \quad \text{s.t.}\ \beta_1, \beta_2 \geq 0, \tag{25}$$

where $e = \sum_j P^2_{\text{user}}(r_j)$, $f = \sum_j P^2_{\text{item}}(r_j)$, $g = \sum_j P_{\text{user}}(r_j) P_{\text{item}}(r_j)$, $h = \sum_j r_j P_{\text{user}}(r_j)$, and $p = \sum_j r_j P_{\text{item}}(r_j)$.

6.3. Hybrid Recommendation Model Based on Time Effect and Sentiment Analysis. We apply a clustering algorithm that integrates time effect and sentiment analysis into the ITWCF algorithm and propose a hybrid recommendation algorithm based on time-weighted and sentiment tendency clustering (TWCHR). The details are as follows (Algorithm 3):

7. Experimental Verification and Results Analysis

7.1. Datasets

7.1.1. MovieLens Dataset. The MovieLens dataset contains 100,000 rating records for 1,682 movies from 943 users, where each user has rated at least 20 movies using a score of 1–5 to represent his or her preference. This dataset is used to verify the influence of the time effect on the recommendation results. In the experiment, 5 groups of data are randomly selected from MovieLens, each of which contains 180 random users' rating information on all items, and the scores of each user in each group of data are sorted by time from most recent to longest ago. The first 70% is used as the training set and the remaining 30% as the test set, and the algorithm is verified using the cross-validation method (Algorithm 3).

7.1.2. Book Dataset. Using a crawler algorithm written in *Python*, more than 20,000 book purchase records, 20,000 scores, and 100 book-related introductions were crawled from the popular e-commerce website jd.com, including book name, book classification, book introduction, user name, user ID, price, purchase time, review information, score, and review time. This dataset is used to verify the influence of the time factor and the sentiment tendency of the review information on users' purchasing behavior.

The dataset is divided into a training set and a test set, according to the ratio of 4 : 1. The training set is used to build the recommendation model, and the test set is used to evaluate the recommendation results. The evaluation indexes include accuracy rate, recall rate, and $F1$ value.

7.2. Evaluation Indexes

7.2.1. Mean Absolute Error (MAE). This evaluates the degree of deviation between the item scores predicted by the recommendation algorithm and the actual scores given by users. The calculation formula is as follows:

$$\text{MAE} = \frac{\sum_{u,i}\left|P\left(r_{ui}\right) - r_{ui}\right|}{n}, \tag{26}$$

where $P\left(r_{ui}\right)$ is the predicted value, r_{ui} is the actual score, and n is the number of predicted items.

7.2.2. Accuracy, Recall, and F1 Values. Recommend n items for user u, denoted as $R\left(u\right)$. Let user u's favorite item set on the test set be $T\left(u\right)$, then the accuracy and recall rates are defined as follows:

$$\begin{aligned} \text{accuracy} &= \frac{\sum_u\left|R\left(u\right)\cap T\left(u\right)\right|}{\sum_u\left|T\left(u\right)\right|}, \\ \text{recall} &= \frac{\sum_u\left|R\left(u\right)\cap T\left(u\right)\right|}{\sum_u\left|R\left(u\right)\right|}. \end{aligned} \tag{27}$$

The accuracy rate and the recall rate are a pair of mutually exclusive indicators, which are usually combined. The $F1$ value is then used to measure the quality of recommendations, as follows:

$$F1 = \frac{2 \times \text{accuracy} \times \text{recall}}{\text{accuracy} + \text{recall}}. \tag{28}$$

7.3. Experimental Design and Results Analysis. Two experiments are designed to verify the effectiveness and feasibility of the method proposed in this paper. The first uses the MovieLens dataset to analyze the influence of parameters on the performance of the algorithm, including the information half-value period, information retention period, number of nearest neighbors, clustering number, and similarity threshold. Based on this, a comparative experiment is designed to compare and analyze the advantages and disadvantages of the proposed method and other methods under the same parameters. The second experiment uses the book dataset to verify the advantages of the proposed method when using the time factor and sentiment tendency.

7.3.1. Personalized Movie Recommendation Results and Analysis. (1) Analyze the Influence of Parameters on the Recommendation Effect.

(1) The influence of information half-value period T_s on the performance of the ITWCF algorithm

In this experiment, we set the information retention period $T^p = 3$ and the nearest neighbor number $cln = 25$, then observe the MAE values of the ITWCF algorithm under different half-value periods. We compare these values with those of the time-weighted collaborative filtering algorithm that does not introduce the time retention period (the TWCF algorithm), as shown in Figure 1.

Figure 1 shows that, in the case including the information retention period $T^p = 3$, the MAE of the ITWCF algorithm is the smallest, and the recommended accuracy is the highest under the information half-value period of 25. Compared with the TWCF algorithm, the ITWCF algorithm has a smaller MAE and higher recommended accuracy under the same half-value period and the same number of nearest neighbors.

(2) The influence of the information retention period T^p on the performance of the ITWCF algorithm

We set the information half-value period T_s as 15 days, 25 days, and 50 days, respectively, and the number of neighbors as $cln = 25$. Then, we observe the trend of the MAE value of the ITWCF algorithm as a function of the information retention period T^p, as shown in Figure 2.

Figure 2 shows that the smaller the value of T_s, the more sensitive the algorithm is to the changes of the value of T^p. That is, when the information retention period changes, the MAE value of the algorithm will change more significantly at smaller T_s. T^p also influences the recommendation results of the ITWCF algorithm, and the optimal values of the information retention period corresponding to different half-value periods are also different. However, the algorithm gives the best accuracy overall when T^p is 2–3 days.

(3) The influence of clustering number K and the target item similarity threshold η on the performance of the TWCHR algorithm

According to the above analysis of T_s and T^p, we set $T_s = 25$ and $T^p = 2$ and the similarity threshold to $\eta = 0.2$, 0.3, and 0.4, respectively. Then, we observe the MAE changes of the TWCHR algorithm under different clustering numbers of K, as shown in Figure 3.

Figure 3 shows that no matter how large the value of η is, when the clustering number is in the range of 6–9, the MAE of the algorithm is relatively low and the prediction accuracy is relatively high. However, when the clustering number is large, the number of items in each cluster is therefore small, and some true neighbor items will be excluded from the nearest neighbor set, leading to inaccurate recommendation results. Meanwhile, when the clustering number is too small, the MAE value will again increase because there are fewer items whose similarity to the cluster center can reach η, resulting in some items not achieving accurate prediction scores. That is, when there are too few clusters, the nearest neighbor candidate set becomes too large and some items that are nonnearest neighbors may be clustered together, which degrades the recommendation results.

(2) Comparative Analysis of Different Algorithms. This is based on the above analysis of the effects of various parameters. In this experiment, we set $T^p = 2$, $T_s = 25$, $K = 6$, and the similarity threshold between the cluster center and the target item as $\eta = 0.3$. The MAE values of the item-based collaborative filtering algorithm (ItemCF), the user-based collaborative filtering algorithm (UserCF), the ITWCF algorithm, the traditional clustering-based TWCHR (TR-TWCHR), and the TWCHR algorithm are, respectively, compared in the case of different numbers of neighbors, and the results are shown in Figure 4.

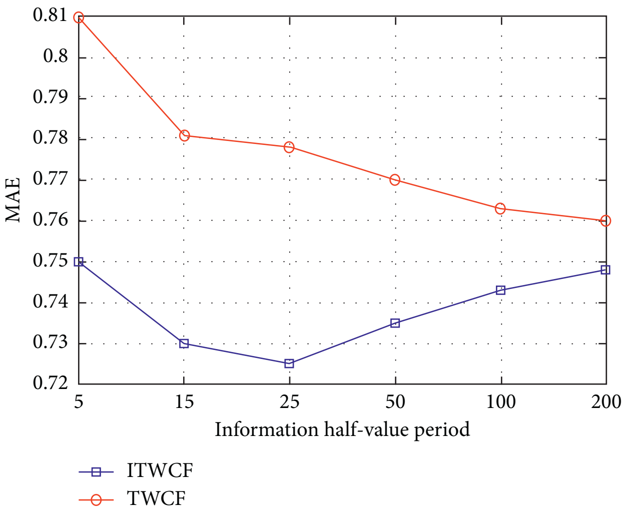

FIGURE 1: The influence of the information half-value period on the recommendation algorithm.

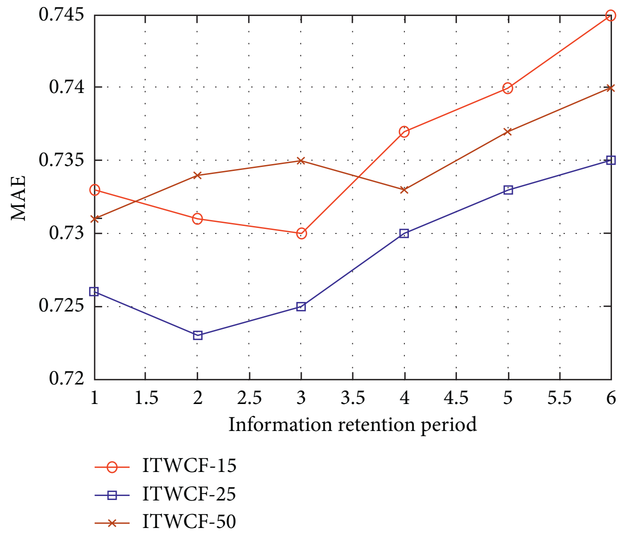

FIGURE 2: The influence of the information retention period on the recommendation algorithm.

The experimental results show that the MAEs of the ItemCF, UserCF, ITWCF, TR-TWCHR, and TWCHR algorithms all show a decreasing trend when the number of nearest neighbors increases. Thus, the selection of the nearest neighbor number is the key factor influencing the performance of the collaborative filtering algorithms. Given the same nearest neighbor number, the MAEs of the ITWCF and the TWCHR algorithms are basically equivalent, except when the number of nearest neighbors is most appropriate, in which case TWCHR outperforms ITWCF. In addition, the ITWCF, TR-TWCHR, and TWCHR algorithms, which all include the time factor, outperform ItemCF and UserCF. In conclusion, the TWCHR algorithm has the smallest MAE and is therefore superior to the others in terms of accuracy.

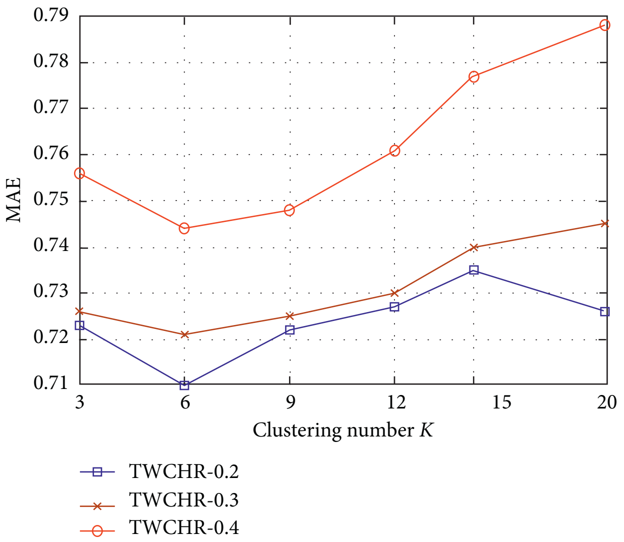

FIGURE 3: The influence of the clustering number on the recommendation algorithm.

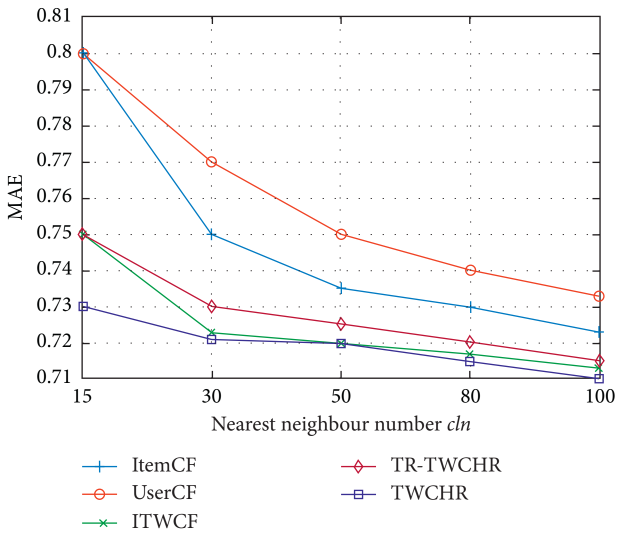

FIGURE 4: Performance comparison of five algorithms under different nearest neighbors.

Through the above experiments, we find that the time factor and the clustering number have an obvious influence on personalized movie recommendations, and the method proposed in this paper using the time-weighted factor to improve the recommendation effect is feasible.

7.3.2. Personalized Book Recommendation and Results Analysis

(1) The Influence of User Review Sentiment Analysis on Recommendation. To verify the effect of the user review sentiment analysis in personalized book recommendations, in this experiment, we compare and analyze the

Table 2: The influence of sentiment analysis on recommendation results under different clustering numbers.

Recommendation algorithm	Clustering number K	Correctly recommended quantity	Recommended quantity	Accuracy (%)	Recall (%)	$F1$ value (%)
TR-TWCHR	10	1171	1852	63.23	29.28	40.02
	20	1435	2208	64.99	35.88	46.23
	30	1526	2096	72.81	38.15	50.07
	50	1470	2137	68.79	36.75	47.91
	100	1187	1839	64.55	29.68	40.66
TWCHR	10	1292	1965	65.75	32.30	43.32
	20	1584	2324	68.16	39.60	50.09
	30	1636	2158	75.81	40.90	53.13
	50	1615	2251	71.75	40.38	51.67
	100	1355	1981	68.40	33.88	45.31

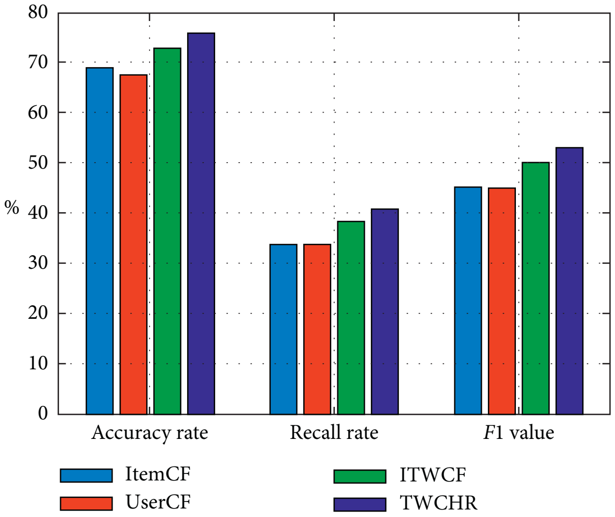

Figure 5: Performance comparison of four algorithms.

recommendation accuracy, recall rate, and $F1$ value of the TR-TWCHR and TWCHR algorithms under the different clustering numbers. The different values of cluster center K in the recommendation algorithm, based on review sentiment analysis, have different effects, so it is necessary to experiment using different K values and observe the influence of the recommendation results, which are shown in Table 2.

Table 2 shows that the recommendation algorithm based on review sentiment analysis (TWCHR) has different recommendation results under different K values. However, in general, the accuracy, recall rate, and $F1$ value of recommendations with the integrated review sentiment analysis are slightly higher than those of the recommendation algorithm without the integrated review sentiment analysis (TR-TWCHR). Thus, the performance of the recommendation model can be improved by integrating review sentiment analysis.

(2) Comparative Analysis of Different Algorithms. To verify the effectiveness of the algorithm proposed in this paper, the ItemCF, UserCF, ITWCF, and TWCHR algorithms (where $K = 30$) are compared in terms of classification accuracy rate, recall rate, and $F1$ value, and the results are shown in Figure 5.

Figure 5 shows that the TWCHR algorithm outperforms the ITWCF, which may be related to the improved method of selecting the nearest neighbors. For the TWCHR, a clustering algorithm combining the time factor with review analysis is used to select the nearest neighbors, whereas for the ITWCF, only the time factor is considered in the selection of the nearest neighbors. This indicates that the sentiment analysis of user reviews has a direct impact on the recommendation accuracy. In addition, the performance of the TWCHR algorithm is significantly better than those of ItemCF and UserCF, which may be related to both the selection method of nearest neighbors and to data sparsity. This indicates that the fusion of the time factor and sentiment analysis is very effective in improving the recommendation accuracy. In addition, the results demonstrate that the hybrid recommendation algorithm effectively combines the advantages of ItemCF and UserCF. Therefore, the algorithm proposed in this paper is reasonable and practical.

8. Conclusions

To solve the problems of information expiration and the use of review information, we first examine collaborative filtering algorithms that integrate the time factor and sentiment analysis. Second, we introduce the concept of an information retention period to improve the time-weighted function, leading to the newly proposed ITWCF algorithm. We then propose a calculation method for the sentiment tendency of review item features and a new clustering algorithm that integrates the time factor and sentiment tendency analysis to optimize the ITWCF algorithm. Third, to take advantage of item-based and user-based collaborative filtering, a hybrid recommendation model is proposed. Finally, two experiments are conducted to verify the proposed algorithm, and the results show that our algorithm can fully account for the time factor and the sentiment tendency and thus improve predictive performance.

Data Availability

The data used to support the findings of the personalized movie recommendation are available from https://grouplens.org/datasets/movielens/, and the original data of precise book review records cannot be released in order to preserve the privacy of individuals.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the Technology Foundation of Guizhou Province (Grant no. QianKeHe-JiChu[2020]1Y269), New Academic Seedling Cultivation and Exploration Innovation Project (Grant no. QianKeHe Platform Talents[2017]5789–21), Program for Innovative Talent of Guizhou Province (Grant no. QianCaiJiao[2018] 190), National Natural Science Foundation of China (Grant nos. 71901078 and 71964009), High-Level Talent Project of Guizhou Institute of Technology (Grant no. XJGC20190929), and Special Key Laboratory of Artificial Intelligence and Intelligent Control of Guizhou Province (Grant no. KY[2020]001).

References

[1] X. Du, X. He, F. Yuan, J. Tang, Z. Qin, and T.-S. Chua, “Modeling embedding dimension correlations via convolutional neural collaborative filtering,” *ACM Transactions on Information Systems*, vol. 37, no. 4, pp. 1–22, 2019.

[2] C. Feng, J. Liang, P. Song, and Z. Wang, “A fusion collaborative filtering method for sparse data in recommender systems,” *Information Sciences*, vol. 521, pp. 365–379, 2020.

[3] Y. Ding and X. Li, “Time weight collaborative filtering,” in *Proceedings of the 14th ACM International Conference on Information And Knowledge Management*, pp. 485–492, Bremen, Germany, November 2005.

[4] H. Zhu, L.-Z. Liao, and M. K. Ng, “Multi-instance dimensionality reduction via sparsity and orthogonality,” *Neural Computation*, vol. 30, no. 12, pp. 3281–3308, 2018.

[5] G. Guo, J. Zhang, and N. Yorke-Smith, “A novel recommendation model regularized with user trust and item ratings,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1607–1620, 2016.

[6] C. Xing, F. Gao, S. Zhan, and L. Zhou, “A collaborative filtering recommendation algorithm incorporated with user interest change,” *Journal of Computer Research and Development*, vol. 44, no. 2, pp. 296–301, 2007.

[7] S. Wu, Z. Xiaonan, and D. Yannan, “A collaborative filtering recommender system integrated with interest drift based on forgetting function,” *International Journal of U- and E-Service, Science and Technology*, vol. 8, no. 4, pp. 247–264, 2015.

[8] C. Chao, S. Qu, and T. Du, “Research of collaborative filtering recommendation algorithm for short text,” *Journal of Computer and Communications*, vol. 2, no. 14, pp. 59–66, 2014.

[9] J. Chen, C. Wang, and J. Wang, “Modeling the interest-forgetting curve for music recommendation,” in *Proceedings of the 22nd ACM International Conference on Multimedia, ACM*, pp. 921–924, Orlando, Florida, USA, November 2014.

[10] J. Wang and T. Liu, “Improving sentiment rating of movie review comments for recommendation,” in *Proceedings of the 2017 IEEE International Conference On Consumer Electronics, IEEE*, Taiwan, China, June 2017.

[11] L. Chen, G. Chen, and F. Wang, “Recommender systems based on user reviews: the state of the art,” *User Modeling and User-Adapted Interaction*, vol. 25, no. 2, pp. 99–154, 2015.

[12] Q. Lin, G. Sheng, W. Cheng, and J. Guo, “Aspect-based latent factor model by integrating ratings and reviews for recommender system,” *Knowledge-Based Systems*, vol. 110, pp. 233–243, 2016.

[13] G. Ganu, Y. Kakodkar, and A. Marian, “Improving the quality of predictions using textual information in online user reviews,” *Information Systems*, vol. 38, no. 1, pp. 1–15, 2013.

[14] J. McAuley and J. Leskovec, “Hidden factors and hidden topics: understanding rating dimensions with review text,” in *Proceedings of the 7th ACM Conference On Recommender System, ACM*, pp. 165–172, New York, NY, USA, October 2013.

[15] Y. Dehkordi, A. Thomo, and S. Ganti, “Incorporating user reviews as implicit feedback for improving recommender systems,” in *Proceedings of the IEEE Fourth International Conference On Big Data & Cloud Computing, IEEE*, Sydney, Australia, December 2015.

[16] R. Latha and R. Nadarajan, “Analysing exposure diversity in collaborative recommender systems-entropy fusion approach,” *Physica A: Statistical Mechanics and Its Applications*, vol. 533, p. 122052, 2019.

[17] J. Liu and G. Deng, “Link prediction in a user-object network based on time-weighted resource allocation,” *Physica A: Statistical Mechanics and Its Applications*, vol. 388, no. 17, pp. 3643–3650, 2009.

[18] H. Song, Y. Fan, X. Liu, and D. Tao, “Extracting product features from online reviews for sentimental analysis,” in *Proceedings Of the 2011 6th International Conference On Computer Sciences And Convergence Information Technology (ICCIT), IEEE*, Seogwipo, South Korea, December 2011.

[19] Z. Dong and Q. Dong, *HowNet and the Computation of Meaning*, World Scientific Press, New York, NY, USA, 2006.

[20] L. Yao, C. Mao, and Y. Luo, “Graph convolutional networks for text classification,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 7370–7377, 2019.