

Retraction

Retracted: Design of Personalized Push Algorithm of Hot Social News Based on User Interest Model

Security and Communication Networks

Received 11 July 2023; Accepted 11 July 2023; Published 12 July 2023

Copyright © 2023 Security and Communication Networks. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1].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

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

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

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

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

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

References

- [1] L. QiuJun, "Design of Personalized Push Algorithm of Hot Social News Based on User Interest Model," *Security and Communication Networks*, vol. 2022, Article ID 9670775, 10 pages, 2022.

Research Article

Design of Personalized Push Algorithm of Hot Social News Based on User Interest Model

Li Qiujun 

Qingdao Vocational and Technical College of Hotel Management, Qingdao 266100, Shandong, China

Correspondence should be addressed to Li Qiujun; 9920390151@qchm.edu.cn

Received 12 May 2022; Revised 17 June 2022; Accepted 30 June 2022; Published 1 August 2022

Academic Editor: Mukesh Soni

Copyright © 2022 Li Qiujun. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As a result of rapid advances in information technology, the volume of information on the Internet is expanding at a breakneck rate. The World Wide Web has evolved into a vast and intricate information space. People have shifted from information deficiency to information overload. The characteristics of Internet information are dispersion, disorder, and mass. A challenging research topic is how to quickly, accurately, and efficiently extract vital information from vast information resources. Web search is becoming one of the Internet field's study centers and focal points. Traditional web search algorithms focus on the link structure of the web and the hierarchical weight of web pages while ignoring the behavior of users, resulting in some search results that are insufficient and inaccurate. In addition, because each web page's hub value and authority value are calculated iteratively, web search is inefficient and susceptible to dispersion and generalization. This study fully integrates the user's interest behavior and relevant, intelligent optimization algorithms to address the shortcomings of the traditional World Wide Web search algorithm, based on a synthesis and analysis of relevant domestic and international research. A method of user interest model construction and update for news recommendation is proposed to address the problem of user interest model construction and user interest drift in the news recommendation system. Initially, the original user interest model is constructed using a bisection K -means clustering algorithm and a vector space model. Subsequently, the forgetting function is constructed using the Ebbinghaus forgetting curve, and the user interest model is time-weighted to achieve the goal of updating the user interest model. User-based collaborative filtering recommendations and item-based collaborative filtering suggestions serve as the experiment's baseline. The experimental results suggest that the recommendation performance of the original user interest model is enhanced, with the F value increasing by 4%. The modified model's F value has increased by 1.3% compared to the previous version.

1. Introduction

With the advancement of communication technology and the rapid development of Internet applications in recent years, the network has amassed a vast amount of various forms of multimedia content, such as text, photos, audio, and video. With the rapid growth of social media (such as Facebook and Twitter abroad, Sina Weibo and WeChat circle of friends in China) and mobile devices supporting wireless data access (such as smartphones and tablets), people can freely create, upload, and share all kinds of multimedia content anytime and anywhere. This makes the Internet, which has conveyed a vast amount of data, ushering in a period of rapid data volume growth [1, 2]. On the one hand, the extremely rich Internet content can meet the

personalized interest needs of each user. On the other hand, the huge amount of Internet data also makes it difficult for people to quickly and properly find the information they need; as the delivery of Internet content, it is also difficult to make their content stand out from the massive amount of information and accurately deliver it to the target audience. This problem called "information overload" has become particularly serious in today's Internet era.

Faced with the escalating expansion of data volume in the information and big data era, personalized recommendation technology has become the preferred method for effectively utilizing massive resource data to provide personalized services to users in various fields. It has made significant contributions to the e-commerce, music, news, and entertainment industries, among others. User interest

modeling is one of the most important technologies for recommendation systems. The collaborative filtering recommendation algorithm was used to develop a user interest model [3–5]. However, the collaborative filtering-based algorithm does not account for the problems of poor interpretability and sparse data in news content [6, 7]. The impact of news classification and content on the recommendation effect will be substantial. Regarding news classification, the literature [8, 9] summarizes and analyzes the news clustering algorithm but does not examine the user interest drift.

In practice, news is highly timely, and users' interests fluctuate over time. Existing algorithms for user interest drift include the time window method, the forgetting function method, and the hybrid algorithm. The time window method employs the movement of the time window to eliminate the most recent user interests, as described in the literature [10]; the forgetting function method employs the forgetting function to alter the weight of items of interest to users at different times. Chung et al. [11] use the Ebbinghaus forgetting curve to represent user interest drift based on a collaborative filtering method; Liu et al. [12] created a dynamic model of user interest drift using clustering and nearest neighbor. A hybrid algorithm is a combination of distinct algorithms. Using a collaborative filtering algorithm, Ghoshal et al. [13] created a hybrid algorithm to discuss the shift in user interest. However, some of these methods only investigate the issue of user interest drift, while others investigate the issue using a collaborative filtering algorithm. In the field of news recommendation, the problem has yet to be resolved.

As a solution to the problems of data cold start and sparsity in traditional collaborative filtering techniques, a film recommendation algorithm based on the user interest model is proposed. Using user records and item information, the algorithm first constructs the user historical interest model and then uses the collaborative filtering algorithm to mine the user behavior interest model and user content interest model. The three models are then merged, and the similarity to the candidate film set is computed. When the number of users exceeds a certain threshold, the volume of calculations for user similarity becomes enormous. The conventional recommendation algorithm will encounter a significant bottleneck problem. If this issue is not resolved effectively, the recommendation system's quality will suffer. The scalability issue of the algorithm must then be resolved.

2. Overview of Personalized Recommendation System and User Interest Model

2.1. Personalized Recommendation System. Content-based recommender systems, collaborative filtering recommender systems, and hybrid recommendation systems are the three types of recommendation systems that exist. The term "content-based recommendation" refers to a recommendation based on a user's purchase history or related text data. Its advantage is that it does not need the introduction of other information, and its disadvantage is that the recommended content lacks diversity. Model-based recommender systems and memory-based recommender systems are two types of collaborative filtering recommendation systems,

while user-based collaborative filtering (UBCF) and item-based collaborative filtering are two types of memory-based recommendation systems (IBCF). Although a collaborative filtering-based recommendation system is extensively utilized, it still has issues, including inefficiency, scalability, and sparse data [8]. A hybrid recommendation system combines a content-based and a collaborative filtering recommendation system. Research shows that the quality of recommendation results has a great impact on user satisfaction, and the accuracy of the recommendation algorithm is the main goal of algorithm research.

2.1.1. Commonly Used Recommendation Theory and Technology

(1) Collaborative Filtering by Users. The UBCF algorithm works by identifying user groups that are similar hobbies to recommend the target users based on the history of user purchase or evaluation, that is, assuming that users with similar purchase history have similar hobbies. Calculating the similarity between users is one of the most significant steps and selecting the size of the user group. If the user group introduces too much information irrelevant to the target user, it will have an impact on the simulation results. If the user group is too small, the reference content is not conducive to the final result. There are many ways, as shown in the following formula:

Cosine similarity is as follows:

$$\text{sim}(u, v) = \frac{\sum_{i=1}^m (r_{ui} \times r_{vi})}{\sqrt{\sum_{i=1}^m (r_{ui})^2} \times \sqrt{\sum_{i=1}^m (r_{vi})^2}}, \quad (1)$$

where u and v represent different users, respectively, $i = 1, 2, \dots, m$ represents all products or projects, and r_{ui} and r_{vi} represent users' scores.

Pearson correlation coefficient (can be computed by (2)) is as follows:

$$\text{sim}(u, v) = \frac{\sum_{i=1}^n (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i=1}^n (r_{ui} - \bar{r}_u)^2} \times \sqrt{\sum_{i=1}^n (r_{vi} - \bar{r}_v)^2}}, \quad (2)$$

where \bar{r}_u and \bar{r}_v represent the average score of the user.

Jaccard similarity (can be computed by (3)) is as follows:

$$\text{sim}(u, v) = \frac{|U \cap V|}{|U \cup V|}, \quad (3)$$

where U and V correspond to the set of products purchased or evaluated by u , v , respectively, and the Jaccard similarity represents the intersection of the two divided by the union.

After obtaining the similarity, the final recommendation score based on collaborative user filtering is formed as shown in the following formula (4):

$$\hat{r}_{ui} = \bar{r}_u + \frac{\left(\sum_{v \in N_{e_u}} ((r_{vi} - \bar{r}_v) \times \text{sim}(u, v)) \right)}{\sum_{v \in N_{e_u}} \text{sim}(u, v)}, \quad (4)$$

where N_{e_u} represents the user u 's neighbor set.

(2) *Collaborative Screening for Projects*. The principle of the IBCF algorithm is similar to that of UBCF, which is to find out the project group similar to the target project according to the historical records of project purchase or evaluation. That is, it is assumed that the projects with similar purchase histories are more similar [14]. The result is also affected by the group size, as shown in the following formula:

$$\hat{r}_{ui} = \bar{r}_i + \frac{\left(\sum_{j \in N_{e_i}} ((r_{uj} - \bar{r}_j) \times \text{sim}(i, j)) \right)}{\sum_{j \in N_{e_i}} \text{sim}(i, j)}, \quad (5)$$

where N_{e_i} represents the neighbor set of user i , and j is the neighbor of i .

(3) *SlopeOne Algorithm*. SlopeOne is a collaborative filtering algorithm based on items. It is calculated according to the score difference of different items and estimates the user's score in a linear manner on the item [15].

Score deviation (can be computed by (6)) is as follows:

$$R(ij) = \frac{\left(\sum_{u \in |N(i) \cap N(j)|} (r_{ui} - r_{uj}) \right)}{|N(i) \cap N(j)|}, \quad (6)$$

where r_{ui} represents the score of user u on item i , $R(ij)$ represents the average deviation of item score, $N(i)$ represents the set of users who overestimate item i , and $|N(i) \cap N(j)|$ represents the set of users who overestimate both items.

Forecast score (which can be computed by (7)) is as follows:

$$\hat{r}_{uj} = \frac{\left(\sum_{i \in N(u)} |N(i) \cap N(j)| (r_{ui} - R(ij)) \right)}{\sum_{i \in N(u)} |N(i) \cap N(j)|}, \quad (7)$$

where $N(u)$ indicates the collection of items that the user overrated.

(4) *Association Rules, AR*. Association rules mainly calculate two indicators of support and confidence. The rules for recommendation can be developed when minimal support and confidence are exceeded [16]. Assuming the rule is “ $A \rightarrow B$ ” and each record is called “transaction,” $|D|$ indicates the data set's total number of transactions, support $n(A \cup B)$ denotes the number of transactions which A and B occur simultaneously, then the support and confidence formulas are as follows:

Support:

$$su(A \rightarrow B) = \frac{\text{support } n(A \cup B)}{|D|}. \quad (8)$$

Confidence:

$$cn(A \rightarrow B) = \frac{\text{support } n(A \cup B)}{\text{support } n(A)}. \quad (9)$$

2.1.2. *Problems in Recommendation System*. There are still numerous issues to be resolved in the recommendation system, which severely limit its effectiveness.

(1) *Sparse Data*. Data sparsity is one of the most prevalent and difficult obstacles to overcome during the recommendation process and even in data mining. The primary cause of sparse data is the data source itself, followed by the process of acquiring the available data (i.e., different angles of using the data may also lead to sparse data). The latter can be overcome through experimental design and repeated validation, whereas the former relies more on reasonable algorithms. The prevalent issue is that the general recommendation process must first be subdivided according to the content, which is embodied in classification, clustering, text division, etc. However, the process of removing irrelevant data will increase the data's sparsity to some extent. In an effort to improve the accuracy of recommendations, the value range of the relevant data has a significant impact on the algorithm's final output. The selection of the number of neighbors in the nearest neighbor algorithm is a typical example [17].

Utilizing the average complement method (low efficiency, poor accuracy), employing fuzzy or overlapping communities to reuse data under different divisions, and designing algorithms that are less sensitive to data density are more common solutions.

(2) *Chilly Start*. Data sparsity issues frequently accompany cold start issues, but they have distinct meanings. Cold start generally refers to the problem of recommending new users or new products during the process of recommending. In other words, there are no available historical data for a user or product that can be used as recommendation credentials. This issue is most prevalent in collaborative filtering algorithms, as collaborative filtering itself calculates the distance between different users or products based on historical data to identify similarities, whereas new users or products cannot calculate the distance [18].

More prevalent solutions include the number of recommendations or the most popular recommendations (degradation of personalized recommendation to impersonalized recommendation). Through new user or product text information to similar product matching (matching similar product data may not be sparse), we design a good algorithm for the effect of new user or product recommendation.

(3) *Understandability*. Some algorithms in recommendation systems have poor interpretability and are frequently used in latent factor models for prediction tasks. By decomposing the matrix to represent the possible feature values, matrix decomposition yields accurate predictive results for the latent factors. The number of feature dimensions ranges from low to high dimensions, making it challenging to describe their dimensions one-to-one (even though it achieves high prediction and potential features may be of high value) and unable to explain the underlying principle of certain phenomena.

Examining the corresponding problems from the perspective of the underlying principle can provide a more complete explanation; extract the primary or content-based features and match them with potential features to improve the interpretability of the model; the observable features and potential features are combined to form an overall feature matrix, detect collinearity, eliminate redundant features, and then manually match the remaining features.

(4) *Time Productivity (Parallelization)*. Parallelization problems are included but are not limited to time efficiency issues. The time efficiency of a model or algorithm has historically been one of the most important metrics for evaluating the algorithm. The complexity of algorithm design is typically correlated with increased time efficiency. With the development of computer clusters, parallel computing is a means to increase time efficiency. However, not all algorithms support parallel processing. In general, algorithms recommended for widespread use that does not involve logical iteration can perform parallel computing. Nonetheless, the parallelization of increasingly complex algorithms has become one of the challenges in the field of recommendation.

Common solutions include designing relatively simple algorithms to improve time efficiency by reducing complexity, parallelizing available data and algorithms to a certain extent via integrated learning, designing parallelizable algorithms, or transforming common basic algorithms into parallelized algorithms to participate in the design, which can significantly improve time efficiency.

(5) *Dynamic Curiosity*. User interests are not constant but constantly evolve. From static interest modeling with a high error rate to dynamic interest modeling that vastly improves the recommendation effect, the problem's difficulty has increased significantly. How to capture the changing interests of users, measure the value of users' interests qualitatively and quantitatively, and create the possibility of purchase are the challenges of dynamic interest. In addition, models of dynamic interest modeling frequently have timeliness, and relevant research fields influence the range of prediction and the efficacy of model parameters.

Common solutions include designing an algorithm that combines long-term and short-term interest, enriching the model with elements describing user dynamics, developing adaptive dynamic parameters, and regularly updating them to ensure their effectiveness.

(6) *Incremental Data*. Another challenging aspect of a recommendation system is the incremental data problem, that is, how to handle this portion of data when the underlying modeling data changes gradually. The size of incremental data will have a significant impact on the recommended results of the algorithm. If the new data are substantially larger than the existing historical data, the previously established model will lose credibility. Similarly, incremental data will not only cause model instability but also improve the incremental model's stability.

Common solutions include adding incremental data to the training set and training the entire model (low efficiency); modeling with incremental data and then integrating it with existing models; and employing the design principles and processing methods suggested by stream data to make better use of incremental data.

(7) *Scalability*. The scalability of a recommendation system refers to its ability to utilize massive amounts of data. This is not a parallel application to large-scale clusters, but rather the processing efficiency and results of the algorithm on large-scale data. Some algorithm designs cannot even complete the process of modeling large-scale data. The era of big data is fundamentally characterized by massive amounts of data. As the primary instrument for massive data mining, the recommendation system should address this issue to the greatest extent possible.

Common solutions include considering the feasibility and time efficiency of massive data applications when designing algorithms; the improved algorithm is comprised of some proven scalable basic algorithms.

(8) *Additional Suggested Indicators*. The most essential characteristic of a recommendation algorithm is its precision. Obviously, in various application scenarios, the conversion rate for advertising recommendations can vary. However, these indicators measure the recommended algorithm based on the accuracy of its predictions. In this age of individualism, users dislike recommendations that are stereotypical or identical to those of others. The diversity of the recommendation list is another important factor for users to consider. Adding multiple indicators to the algorithm's comprehensive evaluation increases the difficulty of recommendation without question.

The implementation of Pareto optimization or weighted optimization for a variety of indicators in order to determine the effect of the suggested algorithm is a common solution.

2.2. *The User Interest Model*. There are four models [19]. Considering the high dimensionality of news data and the convenience of news clustering to construct a user interest model, the news feature is represented using the vector space model.

Classify the vectorized news. At present, model-based algorithms, grid-based algorithms, density-based algorithms, and distance-based algorithms are the most common clustering methods used in data mining [9]. The data studied in this paper is news data, which has the characteristics of massive and high-dimensional. At the same time, the text qualities of news are represented using the vector space model. Based on the foregoing concerns, this research uses a distance-based algorithm to cluster news. Literature [20, 21] studies show that the improved algorithm of K -means clustering algorithm bisection K -means has a faster convergence speed and better clustering effect. To sum up, this paper will use the bisection K -means clustering algorithm to implement a vector space model for news classification.

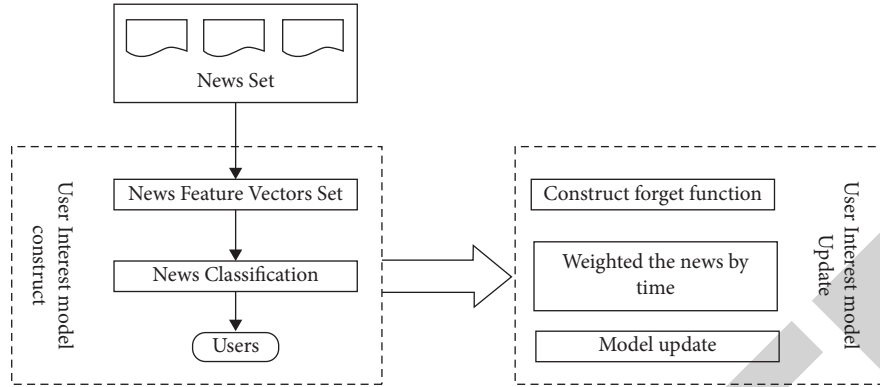


FIGURE 1: User interest model construction and updating framework.

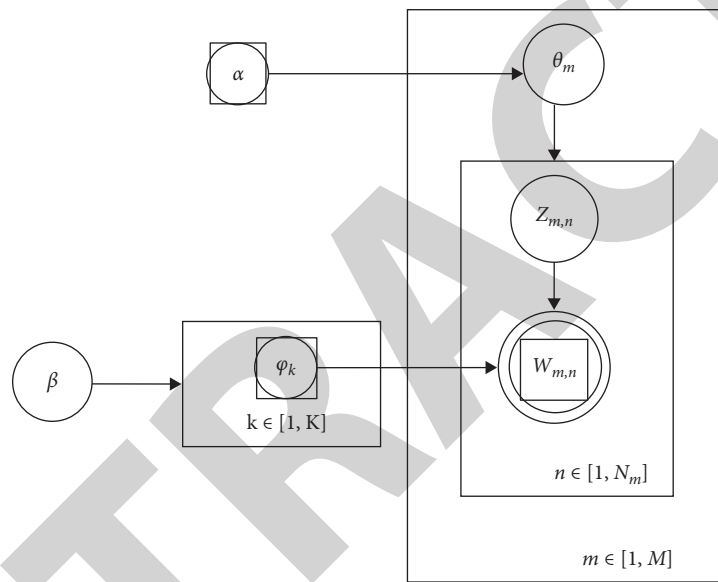


FIGURE 2: Probability graph model of LDA.

Figure 1 depicts the development and revision of the user vector model proposed in this paper. Initially, the following concepts are discussed:

- (a) *News-Related Keywords*. The most representative words in the news can represent the uniqueness and singularity of the news, which are typically extracted by an algorithm for text processing.
- (b) *News Element Vector*. Since news content belongs to the text type, a multidimensional vector D is utilized to represent news content. The result of vectorizing news text is known as the news feature vector.

3. Use an Interesting Model-Based Recommendation Algorithm

A topic model is a text implied topic modeling method that can mine the potential topics in the text. LDA is the most classical algorithm in the topic model. It is also a generation model. According to this theory, each word in a document is obtained through a process of “selecting a topic with a

certain probability and selecting a word with a certain probability in this topic.” The likelihood of terms in each document is provided in a formula according to the description of the LDA topic model’s generation process (10):

$$p(\text{word}|\text{document}) = \sum_{\text{topic}} p(\text{word}|\text{topic}) * p(\text{topic}|\text{document}). \tag{10}$$

The probability diagram model of LDA is shown in Figure 2. Where M is the number of documents, K is the number of topics, V is the length of the word bag, N_m is the total number of words in the m -th document, α and β are a priori parameters, θ is a matrix of $M \times K$, and θ_m represents the topic distribution of the m -th document. The process from α to θ to Z means that when generating the m -th document, first determine the topic distribution of the m -th document, and then determine the topic of the n -th word in the m -th document. φ is a $K \times V$ matrix, φ_k represents the word distribution of the k th topic, and the process from β to φ to W represents that among the K topics, the topic numbered $Z_{m,n}$ is selected, and then, the n -th word $W_{m,n}$ in

```

Input: score matrix  $R$ , item set  $D$ , user set  $N$ , and target user  $u$ 
Output: Recommended list of the target user  $u$ 
Begin:
 $U_{sim} = \emptyset, sim = \emptyset, rank = \emptyset$ 
for  $v$  in  $N$ 
  for  $d$  in  $N(u)$  and  $d$  in  $N(v)$ 
     $sim(u, v) = sim(u, v) + 1 / (\ln(1 + len(D(d))))$ 
  end for
   $sim(u, v) = sim(u, v) / (len(N(u)) * len(N(v)))$ 
end for
 $U_{sim} = sorted(sim(u, v))[0: N]$ 
for  $v$  in  $U_{sim}$ 
  for  $i$  in  $N(v)$  and  $i$  not in  $N(u)$ 
     $rank(i) = rank(i) + sim(u, v)$ 
  end for
end for
return sorted(rank)[0: N]

```

ALGORITHM 1: Collaborative screening algorithm with users.

the m -th document is generated. The input of the LDA algorithm is a large-scale document set, two super parameters, and the number of topics. Two distributions are obtained after training the LDA topic model: document topic probability distribution and topic word probability distribution.

The user-based collaborative filtering algorithm believes that a user will like what the nearest neighbor who has similar interests and hobbies likes. It primarily uses behavioral similarity to calculate interest similarity.

The calculation of user similarity is based on the item set of the common score, which is usually calculated by cosine similarity, as shown in the following formula:

$$sim(u, v) = \frac{(\sum_{i \in N(u) \cap N(v)} 1 / \ln(1 + |D(i)|))}{\sqrt{|N(u)|} \sqrt{|N(v)|}}, \quad (11)$$

where $D(i)$ represents the user set that has acted on item i , and $N(u)$ represents the item set that user u has acted on. A user-based collaborative filtering method is shown in the following Algorithm 1:

While user-based collaborative filtering is not sensitive to the cold-start problem of items, the first driver problem, namely how the first user finds new items, needs to be addressed. If the item is displayed to the user at random, it is obviously not particularly personalized. Thus, try leveraging the item's content information to recommend new goods to users who have previously liked items with similar content.

Create a user historical interest model by looking into the user's previous scoring records, and then recommend a group of items for the user. User history is limited, and therefore data sparsity problems. In view of this problem, based on user behavior, we offer a user interest model and item content to recommend to users.

First, the film is divided into attributes by title, director, screenwriter, starring, type, and introduction, and the film attribute distribution file is generated. Then, the LDA theme model is used to model the film theme distribution, and the

film theme probability distribution is obtained, which is used to calculate the similarity.

Given the movie set $M = \{m_1, m_2, \dots, m_n\}$, each movie is regarded as a separate document. For the content information in the document, such as entities such as director and starring, these entities can be directly regarded as movie attributes. However, for example, introduction, it is necessary to segment the text content, change from word to word stream, extract named entities from word stream, and take these named entities as movie attributes to form movie attribute distribution. The LDA algorithm is used to model the film attribute distribution to obtain the topic feature sequence $F = (f_1, f_2, \dots, f_k)$, the number of subjects is set to K , and the film topic probability distribution matrix Θ is shown in the following formula:

$$\Theta = \begin{pmatrix} w_{11} & \cdots & w_{1k} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nk} \end{pmatrix}. \quad (12)$$

For any user, the probability distribution matrix Θ of the reviewed movie and movie theme is used for the mathematical operation to obtain the weight vector corresponding to F , which is called the user historical interest model. Its mathematical formula is $UHIM = (w_{11}, w_{12}, \dots, w_{1i}, \dots, w_{1k})$, where w_{1i} in UHIM signifies the weight of the theme word f_i . The weight calculation of the subject word f_i in user u 's UHIM is shown in (13). In the current circumstance, this value represents the user's interest distribution and better reflects the user's historical interests.

$$w_{1_{ui}} = \frac{\sum_{m \in M_u} w_{mi}}{|M_u|}, \quad (13)$$

where M_u is the movie collection of user u comments.

For any user, the similarity between user behaviors is calculated by using the reviewed movies, and through collaborative filtering, the user is recommended a historical

```

Input: Film-topic probability distribution matrix  $\Theta$ , user set  $N$ , target user  $u$ 
Output: The interest model for target user  $u$ 
Begin:
 $UHIM_u = (w1_{u1} \dots, w1_{ui} \dots, w1_{uk}), UAIM_u = (w2_{u1} \dots, w2_{ui} \dots, w2_{uk})$ 
 $UCIM_u = (w3_{u1} \dots, w3_{ui} \dots, w3_{uk}), UIM_u = (w4_{u1} \dots, w4_{ui} \dots, w4_{uk})$ 
 $U_{con} = \emptyset, U_{act} = \emptyset, sim_{con} = \emptyset, sim_{act} = \emptyset$ 
for  $d$  in  $N(u)$ :
     $w1_{ui} += w_{di}$ 
end for
 $w1_{ui} = w1_{ui} / \text{len}(N(u))$ 
for  $v$  in  $N$ :
    for  $d$  in  $N(u)$  and  $d$  in  $N(v)$ :
         $sim_{act}(u, v) += 1 / (\ln(1 + \text{len}(D(d))))$ 
    end for
     $sim_{act}(u, v) = sim_{act}(u, v) / (\text{len}(N(u)) * \text{len}(N(v)))$ 
     $U_{act} = \text{sorte } d(sim_{act})[0: h]$ 
end for
for  $v$  in  $U_{act}$ :  $sum_{act} = sum_{act} + sim_{act}(u, v)$ 
end for
for  $v$  in  $U_{act}$ :
     $w2_{ui} = w2_{ui} + sim_{act}(u, v) * w1_{vi} / sum_{act}$ 
end for
 $i = 0$ 
While  $i \leq k$ 
     $sum1 += w1_{ui} * w1_{vi}$ 
     $sum += w1_{ui} * w1_{ui}$ 
     $sumv += w1_{vi} * w1_{vi}$ 
end while
 $sim_{con}(u, v) = sum1 / (\text{sqrt}(sumu) * \text{sqrt}(sumv))$ 
 $U_{con} = \text{sorted}(sim_{con})[0: h]$ 
for  $v$  in  $U_{con}$ 
     $sum_{con} = sum_{con} + sim_{con}(u, v)$ 
end for
for  $v$  in  $U_{con}$ 
     $w3_{ui} = w3_{ui} + sim_{con}(u, v) * w1_{vi} / sum_{con}$ 
end for
 $w4_{ui} = (1 - \alpha - \beta)w1_{ui} + \alpha w2_{ui} + \beta w3_{ui}$ 
return  $UIM_u$ 

```

ALGORITHM 2: The algorithm for building a user interest model.

TABLE 1: Recall rate, accuracy, and F value corresponding to different number of topics.

Number of topics	10	15	20	25	30
Recall	0.165	0.187	0.206	0.185	0.176
Accuracy	0.289	0.305	0.327	0.3	0.291
F value	0.2100	0.2318	0.2527	0.2288	0.2193

TABLE 2: Recall rate, accuracy, and F value corresponding to different h values.

Similar users	10	20	30	40	50
Recall	0.193	0.202	0.214	0.209	0.198
Accuracy	0.309	0.323	0.336	0.33	0.324
F value	0.237	0.248	0.2614	0.2559	0.2457

model of similar user groups. The weight vector corresponding to f is called the user behavior interest model, and its mathematical formula is $UAIM = (w2_1, w2_2 \dots, w2_k)$,

where $w2_i$ represents the weight of the subject word f_i in UAE. When selecting similar user groups, select the first h users with the greatest similarity.

The behavior similarity calculation of user u and user v is shown in equation (11).

The weight calculation of the subject word f_i in user u 's UAIM is shown in the following equation:

$$w2_{ui} = \sum_{v_{ar} \in U_{act}} \left(\frac{sim_{act}(u, v_{ar})}{\sum_{v_{as} \in U_{act}} sim_{act}(u, v_{as})} \right) w1_{v_{ar}}, \quad (14)$$

where U_{act} is the user group whose behavior is similar to that of user u .

The similarity between user contents is calculated for each user in combination with the content information of the movie, and the historical interest model of similar user groups is recommended to the user through collaborative filtering. The weight vector corresponding to f is called the user content interest model, and its mathematical formula is

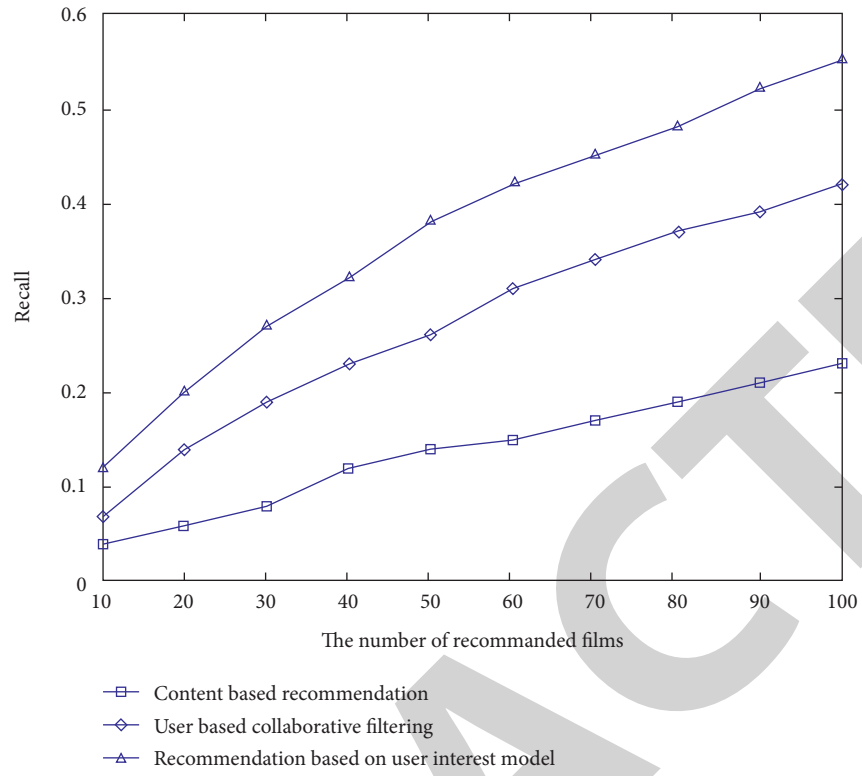


FIGURE 3: Recall rate comparison.

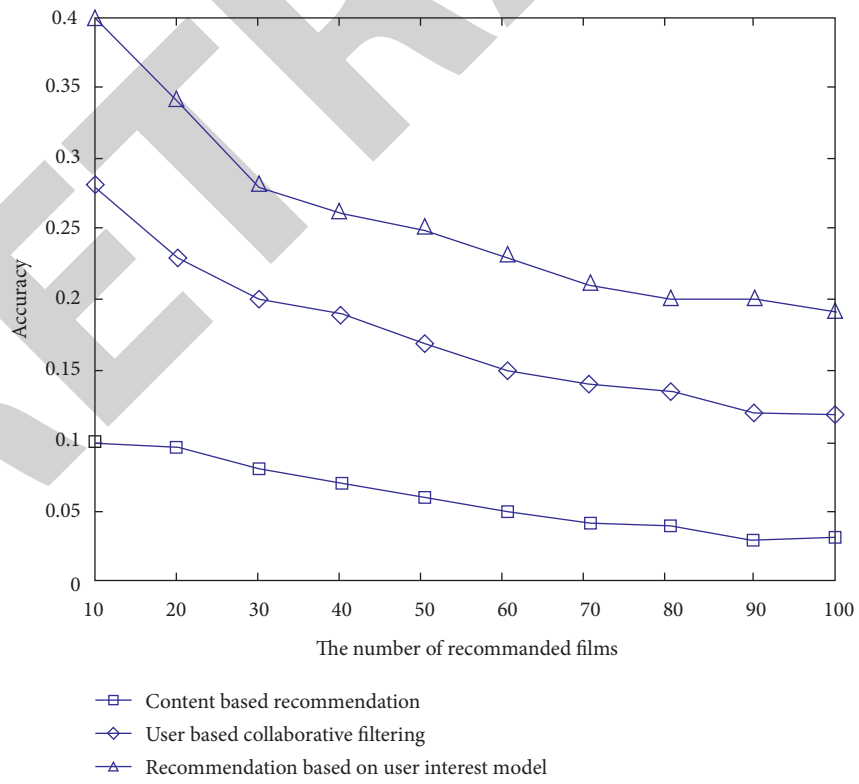


FIGURE 4: Accuracy comparison.

$UCIM = (w3_1, w3_2 \dots, w3_k)$, where $w3_i$ denotes the weight of the subject word f_i in $UCIM$. When selecting similar user groups, select the first h users with the greatest similarity.

Let user u comment on the movie set $M_u = \{m_{u1}, m_{u2}, \dots, m_{us}\}$, historical model $UHIM_u = (w1_{u1}, w1_{u2} \dots, w1_{uk})$, user v comment on the movie set $M_v = \{m_{v1}, m_{v2}, \dots, m_{vt}\}$, $UHIM_v = (w1_{v1}, w1_{v2} \dots, w1_{vk})$.

The content similarity calculation of user u and user v is shown in the following formula:

$$\text{sim}_{\text{con}}(u, v) = \frac{UHIM_u \times UHIM_v}{|UHIM_u| \cdot |UHIM_v|}. \quad (15)$$

The description of the algorithm for building a user interest model is shown in the following Algorithm 2:

4. Experiment and Results

1337 films, 1535 users, and 109398 scoring records from the Douban film network are comprised of the experimental data.

The offline experimental method is utilized to assess this study. The accuracy/recall rate and F values are selected to evaluate the accuracy of the recommended algorithm. The recall rate describes the number of items that have actually produced behavior that is included in the final list of suggestions; it accurately describes the number of recommendations in the final list that has actually produced behavior; the F value is the harmonic mean between recall and accuracy. The N items recommended to user u are recorded as $R(u)$, while the test set items on which user u has acted are recorded as $T(u)$. The formula for calculating the recall rate is as follows:

$$\text{Recall}_u = \frac{\sum_u |R(u) \cap T(u)|}{\sum_u |T(u)|}. \quad (16)$$

The accuracy calculation is shown in the following formula:

$$\text{Precision}_u = \frac{\sum_u |R(u) \cap T(u)|}{\sum |R(u)|}. \quad (17)$$

The F value is calculated as shown in the following formula:

$$F = \frac{2 \times P \times R}{P + R}. \quad (18)$$

4.1. Determination of Subject Number K . When modeling with the LDA theme model, it is necessary to set the number of themes K . Table 1 shows the impact of different K values on recall, accuracy, and F value when the number of movies recommended to each user is 20. It can be seen that K is the best value when K is 20.

4.2. Determination of Nearest Neighbor h . When building UAIM and UCIM, the number of nearest neighbors needs to be set. Table 2 shows the impact of different h values on recall rate, accuracy rate, and F value when there are 20 themes, and

each member has 20 recommended movies. It can be seen that the best value is when the number of similar users is 30.

4.3. Recall Rate, Accuracy Rate, and F Value under Different Recommended Methods. Figure 3 depicts the effect of the varying number of recommended movies on the recall rate of the three recommendation algorithms. As the number of recommended movies increases, the recall rate increases when the number of themes is 20, and the nearest neighbor is 30.

Figure 4 depicts the effect of the varying number of recommended movies on the F value of the three recommendation algorithms when the number of topics is 20, and the nearest neighbor is 30. The precision declines as the number of recommended films increases.

5. Conclusion

The recommendation system can assist users in selecting suitable alternatives from the vast product space, thereby significantly reducing their selection expenses. A recommendation system has already established itself as an essential component of e-commerce websites due to the continuous growth of information. The personalized recommendation system can not only suggest solutions that are tailored to the individual's needs based on their personal interests and increase user loyalty to the website but it can also guide users' purchases and increase the conversion rate of users. However, the dynamic user interest makes it difficult to model the recommendation system, which ultimately impacts the algorithm's precision. The primary objective of this study is to improve the accuracy of the recommendation algorithm. This paper tracks the dynamic changes in user interest by introducing information about user behavior, such as interest forgetting and knowledge acquisition, and ultimately achieves an improvement in the recommendation effect.

A film recommendation algorithm based on the user interest model is proposed as a solution to the problems of data cold start and sparsity in traditional collaborative filtering techniques. Using user records and item information, the algorithm first constructs the user historical interest model, and then, the user behavior interest model and user content interest model are mined using the collaborative filtering algorithm. Finally, the three models are merged, and then, the similarity with the candidate film set is calculated. When the number of users surpasses a certain threshold, the volume of user similarity calculations becomes enormous. The conventional recommendation algorithm will experience a severe bottleneck issue. If this issue is not effectively resolved, the quality of the recommendation system will suffer. The algorithm's scalability problem must then be resolved.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] Q. Liu, M. Huang, B. Guo, and C. Na, "Research on browsing pattern of group users based on ACO[C].Computer application and system modeling (ICCASM), 2010 international conference on," *IEEE*, vol. 5, 2010.
- [2] H. Hou and L. Zhang, "Application Research on the Information Filtering Technology in the Network Information transmission," in *Proceedings of the Computer Science and Service System (CSSS), 2011 International Conference on. IEEE*, pp. 2120–2123, Nanjing, China, June 2011.
- [3] U. Manber, A. Patel, and J. Robison, "Experience with personalization of yahoo!" *Communications of the ACM*, vol. 43, no. 8, pp. 35–39, 2000.
- [4] I. Rowlands, D. Nicholas, P. Williams et al., "The Google generation: the information behaviour of the researcher of the future," *ASLIB Proceedings*, vol. 60, no. 4, pp. 290–310, 2008.
- [5] J. Teevan, C. Alvarado, M. S. Ackerman, and K David R, "The perfect search engine is not enough: a study of orienteering behavior in directed search," in *Proceedings of the SIGCHI conference on Human factors in computing systems. ACM*, pp. 415–422, Vienna Austria, April 2004.
- [6] M. Gan and R. Jiang, "Constructing a user similarity network to remove adverse influence of popular objects for personalized recommendation," *Expert Systems with Applications*, vol. 40, no. 10, pp. 4044–4053, 2013.
- [7] G. Adomavicius and A. Tuzhilin, "Personalization technologies: a process-oriented perspective [J]," *Communications of the ACM*, vol. 48, no. 10, pp. 83–90, 2005.
- [8] X. Chen, X. Liu, Z. Huang, and H Sun, "Regionknn: A Scalable Hybrid Collaborative Filtering Algorithm for Personalized Web Service recommendation," in *Proceedings of the Web Services (ICWS), 2010 IEEE International Conference on. IEEE*, pp. 9–16, Miami, FL, USA, July 2010.
- [9] T. Zhou, Z. Kuscsik, J. G. Liu, M. Medo, J. R. Wakeling, and Y. C Zhang, "Solving the apparent diversity-accuracy dilemma of recommender systems," *Proceedings of the National Academy of Sciences*, vol. 107, no. 10, pp. 4511–4515, 2010.
- [10] C. Veena and B. V. Babu, "A user- based recommendation with a scalable machine learning tool," *International Journal of Electrical and Computer Engineering*, vol. 5, no. 5, p. 1153, 2015.
- [11] K. Y. Chung, D. Lee, and K. J. Kim, "Categorization for grouping associative items using data mining in item-based collaborative filtering," *Multimedia Tools and Applications*, vol. 71, no. 2, pp. 889–904, 2014.
- [12] Y. Liu, D. Liu, H. Xie, and L Wang, "A research on the improved slope one algorithm for collaborative filtering," *International Journal of Computing Science and Mathematics*, vol. 7, no. 3, pp. 245–253, 2016.
- [13] A. Ghoshal, S. Menon, and S. Sarkar, "Recommendations using information from multiple association rules: a probabilistic approach," *Information Systems Research*, vol. 26, no. 3, pp. 532–551, 2015.
- [14] R. Rafeh and A. Bahrehmand, "An adaptive approach to dealing with unstable behaviour of users in collaborative filtering systems," *Journal of Information Science*, vol. 38, no. 3, pp. 205–221, 2012.
- [15] R. L. Tang and Y. J. Fang, "Modification of particle swarm optimization with human simulated property," *Neuro-computing*, vol. 153, pp. 319–331, 2015.
- [16] C. L. Huang and W. L. Huang, "Handling sequential pattern decay: developing a two-stage collaborative recommender system," *Electronic Commerce Research and Applications*, vol. 8, no. 3, pp. 117–129, 2009.
- [17] Y. Ding and X. Li, "Time weight collaborative filtering [C]. Proceedings of the 14th ACM international conference on information and knowledge management," *ACM*, pp. 485–492, 2005.
- [18] D. Godoy and A. Amandi, "Interest drifts in user profiling: a relevance-based approach and analysis of scenarios," *The Computer Journal*, vol. 52, no. 7, pp. 771–788, 2009.
- [19] R. Kumar, B. K Verma, and S. Sunder Rastogi, "Social popularity based SVD++ recommender system," *International Journal of Computer Application*, vol. 87, no. 14, pp. 33–37, 2014.
- [20] A. Tsymbal, "The problem of concept drift: definitions and related work," *JJ Computer Science Department*, vol. 106, no. 2 Trinity College Dublin, 2004.
- [21] L. Cagliero, "Discovering temporal change patterns in the presence of taxonomies," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 3, pp. 541–555, 2013.