LFM Book Recommendation Based on Fusion of Time Information and K-Means

Dawei Ji
School of Humanities and Art, Nanchang Institute of Technology, Nanchang 330099, China

***Abstract*—To meet the growing demand in the field of book recommendation, the research focuses on meeting the personalized needs, behavioral patterns, and interests of readers. A book recommendation algorithm that combines K-means clustering with time information is proposed to provide more convenient and efficient book recommendation services and enhance readers' reading experience. The algorithm constructs a comprehensive user preference matrix by incorporating readers' borrowing time. Then, the K-means clustering is applied to group users with similar preferences and leverages a latent factor model to train and predict user ratings. The methodological integration of clustering and latent factor model ensures a more precise and dynamic recommendation process. The experimental results demonstrated that the proposed algorithm achieved a high average recommendation accuracy of 98.7%. Additionally, the algorithm maintained an average book popularity score of 8.2 after reaching stability, indicating its ability to suggest widely appreciated books. These outcomes validate the effectiveness of the algorithm in delivering accurate and popular book recommendations tailored to individual readers' needs. This study combines K-means clustering with time sensitive preference analysis and latent factor model to introduce an innovative method in the field of book recommendation systems. The findings provide valuable insights and practical applications for libraries seeking to enhance their personalized recommendation services, offering a significant contribution to the field of intelligent information retrieval.**

***Keywords*—Book recommendation; K-means; time information; latent factor model; preference matrix**

I. Introduction

The Book Recommendation (BR) system is an important application that can provide users with personalized book recommendations. As information technology develops, people's demand for obtaining, sharing, and purchasing books continues to increase, making recommendation systems crucial in helping users discover interesting books [1]. Traditional BR systems often use Collaborative Filtering (CF) algorithm for recommendation, which analyzes users' historical behavior data, mines similarities between users or items, and makes recommendations. However, CF algorithm still has some problems. Firstly, CF is unable to handle cold start issues well, which means that there is a lack of sufficient data to accurately recommend new users or newly listed books. Secondly, CF does not consider the specific relationship between books and users, resulting in a lack of diversity and personalization in recommendation results. To overcome these issues, a Latent Factor Model (LFM) is introduced into the BR system. LFM establishes the connection between users and books by representing them as latent feature vectors [2]. This model not only considers the similarity between users and books, but also captures the implicit relationship between users and books, thereby improving the accuracy and personalization of recommendation results [3]. However, although LFM has achieved good results in solving some problems, the existing LFM-BR system cannot fully utilize the time information of users in the process of borrowing books. Meanwhile, it also cannot provide personalized recommendations for users, and still has certain limitations. Therefore, to solve these problems, a preference matrix is constructed by analyzing the borrowing time of readers to reflect their preferences. At the same time, the K-means algorithm and preference matrix are used for reader clustering to identify the reading needs of different preference reader groups. The implicit semantic model is trained and scored for prediction. An LFM-BR algorithm that integrates time information and K-Means clustering is designed. It is hoped to improve the recommendation performance of the BR system, provide users with more accurate personalized BR services, enhance user experience, and provide new ideas for the development of BR systems. This article consists of six sections. Section I is the background of the BR system. Related work is given in Section II. Section III reviews the research on recommendation systems both domestically and internationally. The sections designs the LFM-BR algorithm based on K-means and time information. It constructs a modified preference model based on time information. It optimizes the LFM recommendation algorithm based on K-means and modified preference models. Section IV analyzes the performance of the algorithm and its practical application effects. Finally, the entire article is summarized and its shortcomings are pointed out in Section VI.

II. Related Works

Due to the rapid development of the Internet, a large number of books can be obtained and read online. The number and variety of books continue to increase, but users often feel confused and exhausted. Therefore, to provide personalized BR services, many scholars have conducted in-depth research on recommendation systems. Guo Q et al. designed a knowledge graph recommendation system to address information explosion and enhance user experience in various online applications. The knowledge graph was used as auxiliary information to generate recommendations. These results confirmed that the system had higher recommendation accuracy [4]. Yi B et al. designed a deep matrix factorization model based on implicit feedback embedding to accurately recommend reader preferences. It directly generated potential factors of users and preferences from input information through feature transformation functions. These results confirmed that this model had high accuracy and training efficiency [5]. Cui Z et al. designed a CF-based personalized recommendation system to provide users with accurate and fast information over time, which analyzed user behavior to provide higher quality recommendations. These results confirmed that the system could quickly and accurately make recommendations [6]. Zhou W et al. designed a graph-based personalized recommendation algorithm for sorting to improve user preference matching accuracy in recommendation systems. It matched target users with users with similar preferences through an improved resource allocation process. These results confirmed that the recommendation performance of this algorithm was good [7]. Liu Y et al. proposed a personalized library recommendation model based on small data fusion algorithm to better grasp the needs of library users and provide more accurate knowledge services. The neural network was utilized to achieve multi-dimensional small data fusion. These results confirmed that this model could effectively achieve personalized recommendations [8]. Liu Y designed a CF information recommendation algorithm based on spatiotemporal similarity to meet the academic information recommendation needs of university libraries. An academic information demand model was established through situational awareness and combined with adaptive interest models. These results confirmed that this algorithm could effectively achieve personalized recommendations [9].

Zhang S designed a personalized service method for university libraries based on data tracking technology to identify user interests and provide peer-to-peer service recommendations. The big data behavior tracking technology was utilized to analyze and track the behavioral information of user groups. These results confirmed that this method could accurately recommend and had high efficiency [10]. Frequent data scanning and excessive candidate itemset in the library lead to slow system operation. Therefore, Zhou Y proposed an information recommendation book management system based on improved Apriori. The method integrated C/S and B/S architectures to open book information to staff and borrowers. These results confirmed that the CPU usage of this system was relatively low [11]. Fu M proposed a personalized library resource recommendation system to address the low accuracy and user satisfaction of traditional library recommendation systems. It corrected the bias values and weights of visible and hidden layers in deep belief networks through contrastive divergence method. These results confirmed that this system had high accuracy, recall, and user satisfaction [12]. Chendhur K M K et al. designed an improved CF based on user preferences to improve the execution time and accuracy of prediction problems in BR. A small batch gradient descent algorithm was introduced to make predictions based on user preferences. These results confirmed that this algorithm had high prediction efficiency [13]. Anwar T et al. designed a cross domain BR for sequential pattern mining and rule mining to meet user needs in a shorter amount of time. The semantic similarity was utilized to expand domain recommendations and recommend books that users preferred through rule mining algorithms. These results confirmed that the system had a high-performance score [14]. Saraswat M et al. designed a BR model based on neural recursive network classification to consider the combination of book types and reviews in BR. It categorized book plots and comments into various categories and recommends books to users based on these categories. These results confirmed that the accuracy and F1 value of this model were relatively high [15].

In summary, scholars have proposed various innovative methods aimed at providing personalized and accurate BR services. However, most of these methods rely on the user's historical behavioral data for recommendations, ignoring their real-time or immediate needs. Therefore, the study first constructs a comprehensive preference model for reader borrowing duration. Then, the K-means algorithm is used to cluster the readers. The clustering results are trained using the LFM model. A LFM-BR algorithm that integrates time information and K-means clustering is proposed. Compared with existing research, this method emphasizes the importance of temporal information and identifies reader groups with different preferences through clustering methods. It can better handle data with obvious group and time series characteristics, thus better grasping changes in readers' interests and needs, and making timely recommendations.

III. LFM Book Recommendation Model based on Time Information and K-Means

This chapter mainly studies the improvement method of LFM-BR based on K-means and time information. Firstly, a preference model based on time information is constructed. Next is to improve the function design of BR.

A. *Construction of a Modified Preference Model Based on Time Information*

In the library, readers' borrowing preferences are a constantly changing dynamic process. Over time, readers' interests and needs will change, leading to new interests in different types of books. Faced with this situation, traditional CF often cannot provide satisfactory recommendation results [16]. To address the dynamic changes in reader borrowing preferences over time, this study analyzes the borrowing duration to deeply explore the potential preferences of readers and constructs a comprehensive preference degree model. The set of readers is $A=\{a_1,a_2,...,a_m\}$ and the set of books is $D=\{d_1,d_2,...,d_n\}$. Based on sets of books and readers, the personal reading preference in Eq. (1) can be obtained.

$$L_p(a_i,d_j)=1-\exp(-\frac{ct_{ij}}{\overline{t_i}}) \tag{1}$$

In Eq. (1), L_p refers to individual reading preferences. $\exp()$ represents an exponential function. t_{ij} means the duration of time for reader a_i to borrow book d_j. $\overline{t_i}$ is the average borrowing time of books borrowed by reader $\overline{t_i}$. c represents a parameter that can be adjusted. However, the borrowing situation of each book varies due to factors such as content, number of pages, or category. Some books may attract more readers to borrow and pay attention to them due to their in-depth and engaging content, or their popular themes.

However, other books may have relatively fewer borrowed volumes due to their large number of pages or special categories. Therefore, the next step is to calculate the preference for borrowing books, represented by Eq. (2).

$$L_b(a_i, d_j) = \frac{t_{ij} - t_j(\min)}{t_j(\max) - t_j(\min) + b} \tag{2}$$

In Eq. (2), L_b refers to the degree of preference for book borrowing. $t_j(\min)$ represents the minimum borrowing time of book d_j. $t_j(\max)$ means the maximum borrowing time of book d_j. b is a bias term. Personal reading preferences can reflect the user preference for different themes or types of books, while book borrowing preferences reflect the user's tendency to choose a certain book in actual borrowing behavior. Therefore, considering the two preferences comprehensively, a comprehensive preference model is established by weighted sum to better understand user preferences and improve the accuracy of recommendations, represented by Eq. (3).

$$L_s(a_i, d_j) = \mu \cdot L_p(a_i, d_j) + (1-\mu) \cdot L_b(a_i, d_j) \tag{3}$$

In Eq. (3), L_s represents the comprehensive preference of readers for borrowing books. μ is the weighting coefficient. The matrix in Table I shows the comprehensive preference.

TABLE I. COMPREHENSIVE PREFERENCE MATRIX

Reader /Book	d1	d2	...	dj	...	dn
a1	L (a1, d1)	L (a1, d2)	...	L (a1, dj)	...	L (a1, dn)
a2	L (a2, d1)	L (a2, d2)	...	L (a2, dj)	...	L (a2, dn)
...	...	...	...	...	...	...
ai	L (ai, d1)	L (ai, d2)	...	L (ai, dj)	...	L (ai, dn)
...	...	...	...	...	...	...
am	L (am, d1)	L (am, d2)	...	L (am, dj)	...	L (am, dn)

The interests and preferences of readers will also change over time. Therefore, by constructing preference transfer functions, different weights are assigned to preferences in different time periods. Based on the preference transfer function, the comprehensive preference model is optimized and a comprehensive preference correction model incorporating time information is designed. Considering the borrowing history of readers, recently returned books better reflect their current interests and preferences. Therefore, they should be given higher weight. Books that have been returned for a long time may reflect outdated interests and preferences, so certain punishments should be imposed on them. This is consistent with the law of human forgetting, which means that people are more likely to remember recent events, while their memory of distant events gradually becomes blurred. The Ebbinghaus forgetting curve can describe the forgetting pattern of the human brain. Fig. 1 shows the curve of human memory over time.

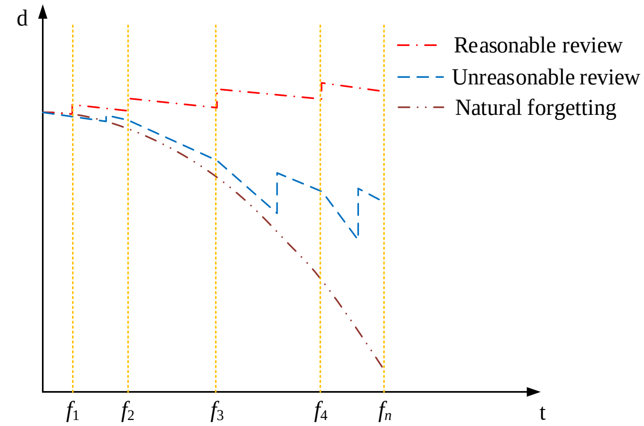


Fig. 1. Time dependent curve of human memory level.

According to Fig. 1, a function can be used to fit the Ebbinghaus forgetting curve and quantify the degree of forgetting. The study adopts Newton's cooling law, represented by Eq. (4).

$$-\frac{dT(t)}{dt} = \varphi\left[T(t) - T_C\right] \tag{4}$$

In Eq. (4), $-\frac{dT(t)}{dt}$ represents the rate at which the temperature $T(t)$ of the object decreases over time t. T_C represents the temperature of the surrounding environment at time t. φ represents the cooling coefficient, which is a proportional constant. The next step is to calculate the relationship between the object temperature at time φ and the initial time through mathematical operations, represented by Eq. (5).

$$T(t) = T_C + \left[T(t_0) - T_C\right] \cdot \exp\left[-\kappa(t - t_0)\right] \tag{5}$$

In Eq. (5), t_0 represents the initial time. The last time the reader returns the book before the current moment is used as the evaluation criterion. Eq. (5) is adjusted to obtain the preference transfer function, which is represented by Eq. (6).

$$\omega(a_i, d_j) = \varepsilon + (1-\varepsilon) \cdot \exp\left[-\zeta(t_c - t_{last})\right], \zeta \in (0,1) \tag{6}$$

In Eq. (6), ω is the preference weight. ε represents a constant. t_{last} is the last time the book is returned. ζ means a time decay coefficient. Finally, by combining the preference transfer function with the comprehensive preference model, a preference correction model based on time information can be obtained, represented by Eq. (7).

$$L'(a_i, d_j) = \omega(a_i, d_j) \cdot \left[\mu \cdot L_p(a_i, d_j) + (1-\mu) \cdot L_b(a_i, d_j)\right] \tag{7}$$

In Eq. (7), L' is the modified preference based on time information. Fig. 2 shows the modified preference matrix.

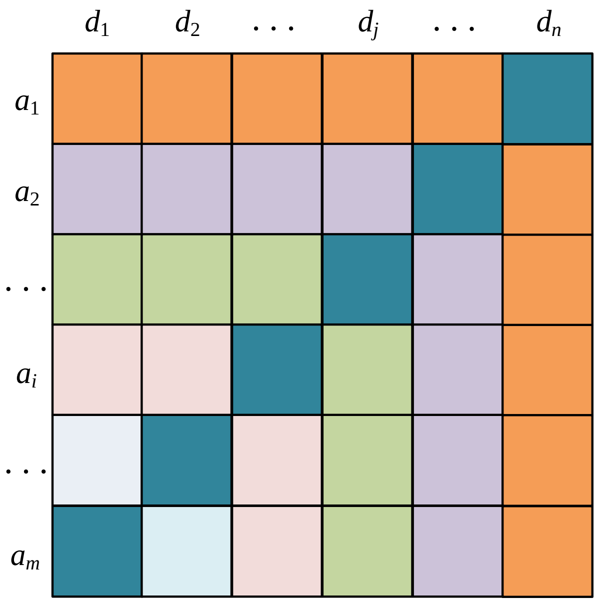


Fig. 2. Modified preference matrix diagram.

B. LFM Recommendation Algorithm Based on K-Means and Modified Preference Degree

The relationship between readers and book categories was not taken into account in the design and modification of preferences. Therefore, it is impossible to fully explore the potential preference information of users. In some libraries, there are more books. The borrowing records of readers are relatively rare [17]. Therefore, the first step is to regard the set of book categories as $G=\{g_1,g_2,...,g_z\}$ and combine the set of books with $G=\{g_1,g_2,...,g_z\}$ to form a category matrix. The elements in the j row and k column of the category matrix are set to Boolean values, either 0 or 1. When $bg_{jk}=1$, it indicates that the book b_j belongs to the g_k class. When $bg_{jk}=0$, it indicates that b_j does not belong to g_k. The category preference of books in Eq. (8) can be obtained.

$$Q_{ik}=\frac{f_{ik}}{\sum_{z=1}^{z} f_{iz}} \tag{8}$$

In Eq. (8), Q_{ik} means the reader's preference for borrowing books. f_{ik} represents the frequency of borrowing g_k books. f_{iz} is the frequency of borrowing g_z books. The next step is to cluster readers using K-means based on the category preference matrix between readers and books, as displayed in Fig. 3.

When clustering, cluster centers are selected based on the similarity between reader preferences for different book categories. The next step is to combine reader clustering with preference correction matrix to establish a preference matrix for the same cluster of readers. The cosine similarity is introduced, and the category preference matrix is inputted to calculate the similarity, which is represented by Eq. (9).

$$similarity(a_x,a_y)=\frac{\sum_{k=1}^{z}(Q_{xk}\cdot Q_{yk})}{\sqrt{\sum_{k=1}^{z}(Q_{xk})^2}\cdot\sqrt{\sum_{k=1}^{z}(Q_{yk})^2}} \tag{9}$$

In Eq. (9), $similarity(a_x,a_y)$ is the similarity in book category preferences among different readers [18]. The next step is to train using the LFM recommendation algorithm, which decomposes the matrix into two low dimensional matrices. One matrix represents the relationship between users and potential features, while the other matrix represents the relationship between items and potential features. These potential features can capture the implicit relationship between users and projects, namely implicit classification. By learning these potential features, users can predict their ratings for projects they have never interacted with before. The user rating of the project is represented by Eq. (10).

$$\begin{cases} S=U\times P^T \\ r_{zi}=\sum_{h=1}^{H} p_{zh}q_{ih} \end{cases} \tag{10}$$

In Eq. (10), S is the rating matrix. U represents the relationship matrix between decomposed users and potential features. P means the relationship matrix between the decomposed project and potential features. r_{zi} refers to the predicted score. H is the number of hidden classifications. p_{zh} represents interest level. q_{ih} means the association between projects and implicit classification. The next step is to obtain the values of parameters p_{zh} and q_{ih} through the objective function. Meanwhile, to prevent overfitting, a regularization term is added to the objective function, represented by Eq. (11).

$$\Phi=\sum_{(z,i)\in R}(r_{zi}-\sum_{h=1}^{H} p_{zh}q_{ih})^2+\lambda(\|p_z\|^2+\|q_i\|^2) \tag{11}$$

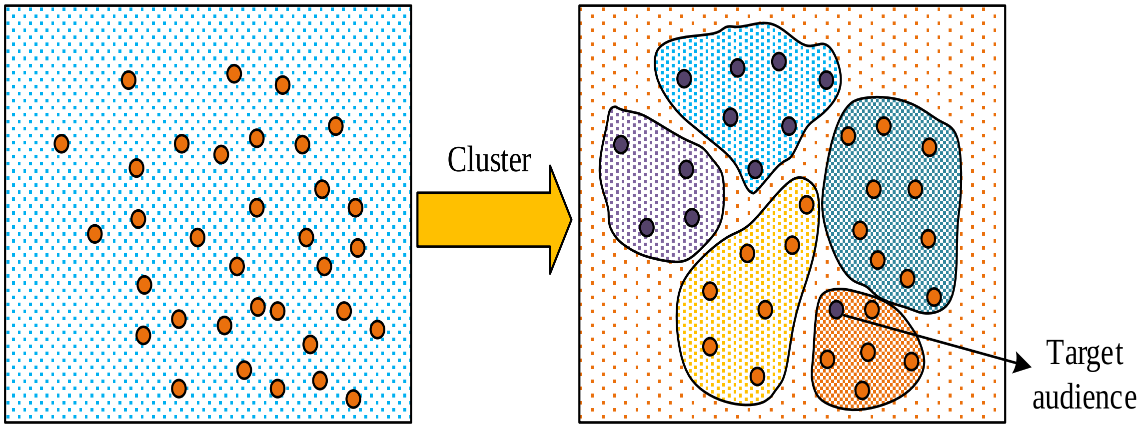


Fig. 3. K-means reader clustering diagram.

In Eq. (11), Φ represents the objective function. R refers to the scoring set. λ is the regularization coefficient. The solution of the objective function usually uses gradient descent method, which minimizes the objective function by taking its derivative and gradually reducing the parameters values. In the gradient descent method, the parameters values are first initialized. Then, the partial derivatives of each parameter in the objective function are calculated to obtain the gradient of the parameters. Next, based on the learning rate setting, the update amount of the parameter is obtained by multiplying it by the gradient value, and it is added to the current parameter value. This process continues until the specified stopping criterion is reached, that is, the objective function has converged or reached a certain number of iterations. The solution of parameters p_{zh} and q_{ih} is represented by Eq. (12).

$$\begin{cases} p_{zh} = s_z q_{ih} (q_{ih}{}^T q_{ih} + \lambda I)^{-1} \\ q_{ih} = s_z p_{zh} (p_{zh}{}^T p_{zh} + \lambda I)^{-1} \end{cases} \tag{12}$$

In Eq. (12), s_z represents the rating of user z. I is the identity matrix. The training process has been completed. Fig. 4 shows the training process of the LFM recommendation algorithm.

The next step is to introduce Sum of Squared Errors (SSE) to evaluate the clustering performance of K-means-based reader clustering, represented by Eq. (13).

$$SSE = \sum_{i=1}^{k} \sum_{x \in A_i} \left| sim(C_i, x) \right|^2 \tag{13}$$

In Eq. (13), A_i represents the set of reader clusters. C_i represents the clustering center of the reader cluster. The next step is to introduce Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to evaluate the rating accuracy of the designed recommendation system. MAE is the average absolute difference between the predicted score and the actual score. A smaller value indicates that the predicted score is more accurate. RMSE is calculated based on squared error, taking into account the error between each predicted score and the actual score, and averaging the error. A small RMSE indicates that the predicted score is close to the actual score. These two indicators are represented by Eq. (14).

$$\begin{cases} MAE = \dfrac{\sum_{z,i \in N} \left| r_{zi}{}' - r_{zi} \right|}{N} \\ RSME = \sqrt{\dfrac{\sum_{z,i \in N} (r_{zi}{}' - r_{zi})^2}{N}} \end{cases} \tag{14}$$

In Eq. (14), N represents the total amount of data. $r_{zi}{}'$ stands for the reader's true rating of the book. Finally, the accuracy, recall, and F1 score are used to predict the accuracy of the recommendation list, represented by Eq. (14).

$$\begin{cases} Accuracy = \dfrac{\sum_{o \in O} \left| S(o) \cap R(o) \right|}{\sum_{o \in O} \left| S(o) \right|} \\ Recall = \dfrac{\sum_{o \in O} \left| S(o) \cap R(o) \right|}{\sum_{o \in O} \left| R(o) \right|} \\ F1 = \dfrac{2 \cdot Accuracy \cdot Recall}{Accuracy + Recall} \end{cases} \tag{15}$$

In Eq. (15), O represents the set of users. $S(o)$ means the recommended list. $R(o)$ is a set of user preferences. Fig. 5 shows the designed LFM recommendation algorithm based on K-means and modified preference.

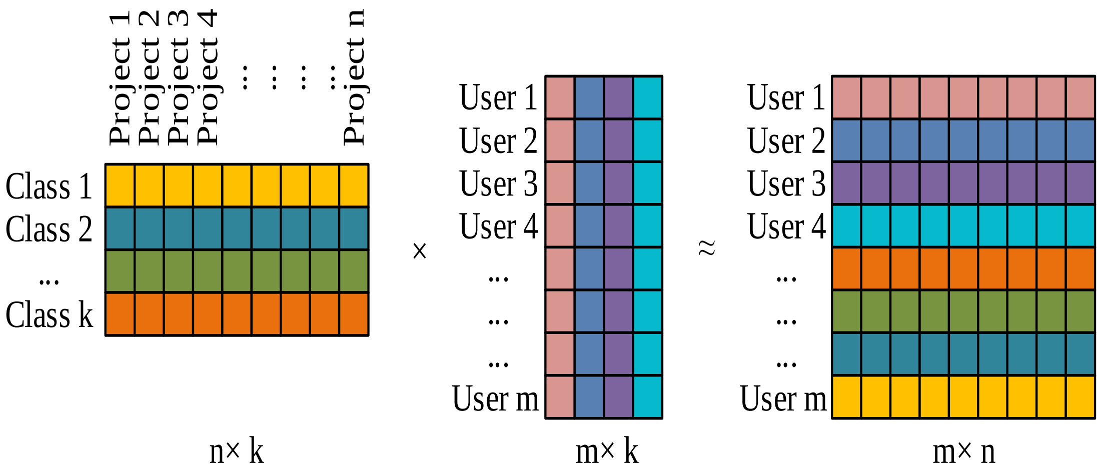


Fig. 4. The training process of the LFM recommendation algorithm.

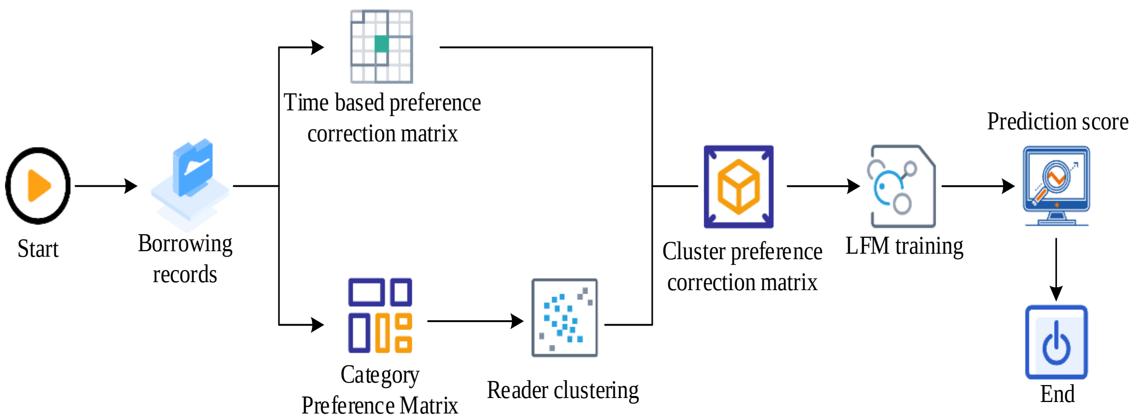


Fig. 5. LFM recommendation algorithm based on K-means and modified preference.

IV. Results of LFM Recommendation Algorithm Based on K-Means and Time Information

This section mainly analyzes the experimental results of the designed LFM-BR. The first step is to analyze the performance of the designed algorithm. The second step is to design simulation experiments for its practical application.

A. *Performance of LFM Recommendation Algorithm based on K-Means and Time Information*

To verify the performance of the designed LFM recommendation algorithm, the study first selected different numbers of clusters and calculated SSE to obtain the optimal number of clusters, as displayed in Table II.

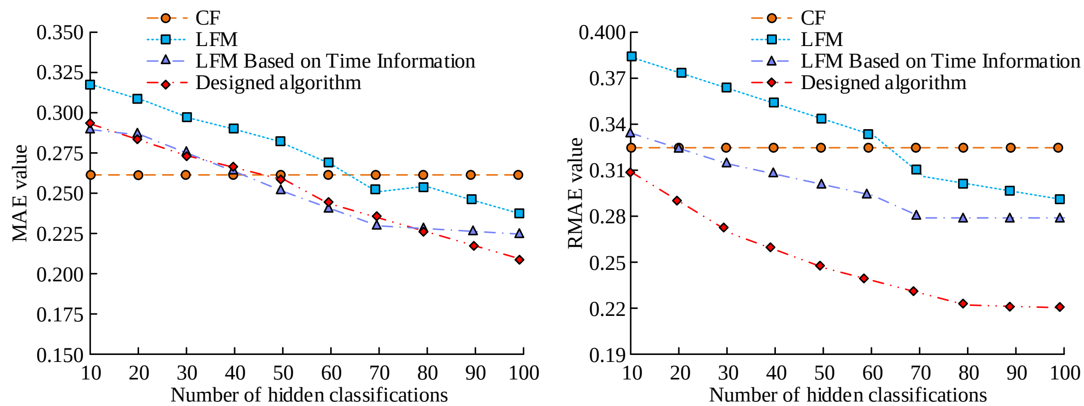
TABLE II. SSE Values for Different Number of Clusters

Number of clusters	SSE value	Number of clusters	SSE value	Number of clusters	SSE value
2	2.279	8	0.898	14	0.608
3	1.875	9	0.784	15	0.473
4	1.663	10	0.715	16	0.498
5	1.512	11	0.692	17	0.457
6	1.318	12	0.618	18	0.473
7	1.301	13	0.635	19	0.463

From Table II, as the clusters increased, the SSE of the designed BR gradually decreased. When the cluster was less than 7, the decrease rate of SSE was faster. When the cluster was greater than 7, the decrease rate of SSE became slower. Therefore, the optimal clusters were 7. The above results indicate that selecting the appropriate number of clusters has a significant impact on clustering effectiveness. In practical applications, it is necessary to choose the appropriate number of clusters based on different business needs and data characteristics to achieve the best recommendation effect. The next step is to calculate the MAE and RMSE of the designed BR, and compare them with CF, LFM, and LFM based on time information. Fig. 6 shows the results.

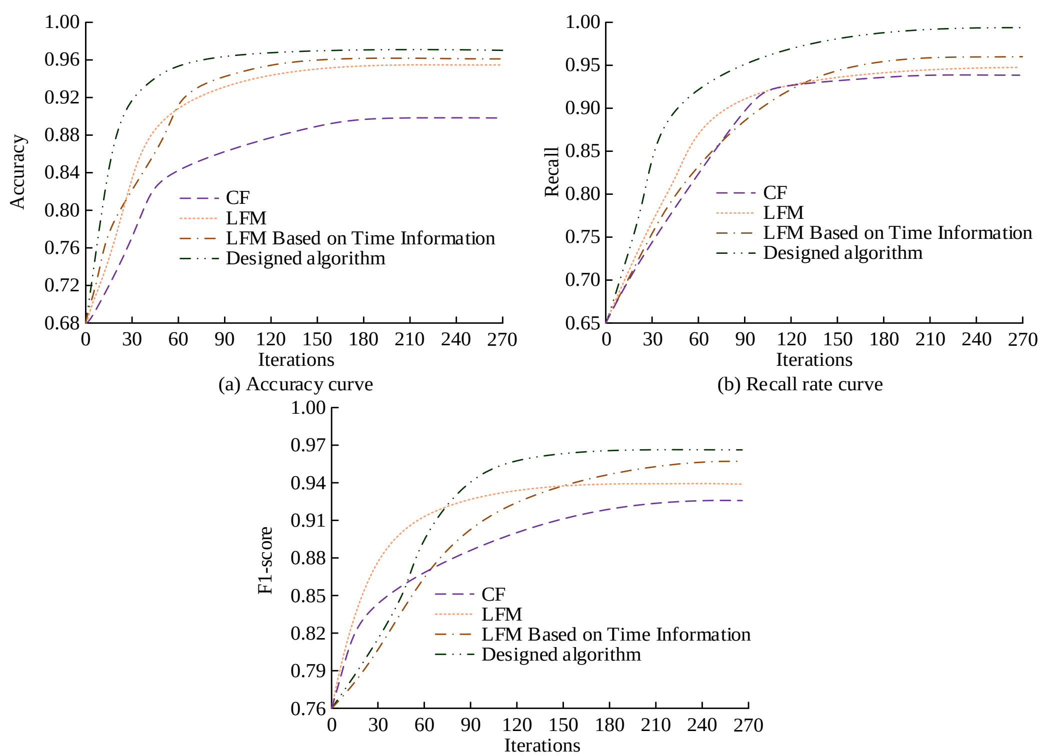
From Fig. 6 (a), as the hidden classification increased, the MAE of CF remained unchanged, while the MAE of other three algorithms showed a decreasing trend and gradually flattened out. Among them, the MAE of CF was always 0.26. The maximum MAE of LFM was 0.32 and the minimum value was 0.24. The maximum MAE of LFM based on time information was 0.29 and the minimum value was 0.23. The maximum MAE of the designed LFM recommendation algorithm based on K-means and time information was 0.29, and the minimum value was 0.21. From Fig. 6 (b), the RMSE of CF still did not vary with the hidden classifications, and its value remained at 0.33. The maximum RMAE of LFM was 0.39 and the minimum value was 0.30. The maximum RMAE of LFM based on time information was 0.34 and the minimum value was 0.28. The maximum RMAE of the designed BR was 0.31 and the minimum value was 0.23. The above results indicate that the designed recommendation algorithm has good performance on prediction accuracy. Finally, to further validate the performance of the designed BR, the accuracy, recall, and F1 score of the four recommendation algorithms were calculated, as displayed in Fig. 7.

From Fig. 7 (a), as the iteration increased, the accuracy of all four algorithms showed an upward trend but gradually stabilized. When CF reached a flat state, the accuracy was 87.8%, and the maximum accuracy of LFM was 93.5%. The LFM based on time information achieved a stable accuracy of 95.7%. The maximum accuracy of the designed BR was 97.3%. From Fig. 7 (b) and 7 (c), the recall and F1 score trends of these four algorithms were consistent with the accuracy trends. When CF reached stability, the recall rate and F1 score were 92.1% and 0.91, respectively. When LFM reached stability, the recall rate and F1 score were 92.6% and 0.93, respectively. When LFM based on time information tended to flatten, the recall rate and F1 score were 95.1% and 0.953, respectively. When the designed algorithm tended to flatten, the recall rate and F1 score were 98.2% and 0.965, respectively. The three indicators of this designed algorithm are significantly higher than the other three algorithms, proving its good overall performance.



(a) MAE values for different algorithms
(b) RMAE values for different algorithms

Fig. 6. MAE and RMSE values of four book recommendation algorithms.



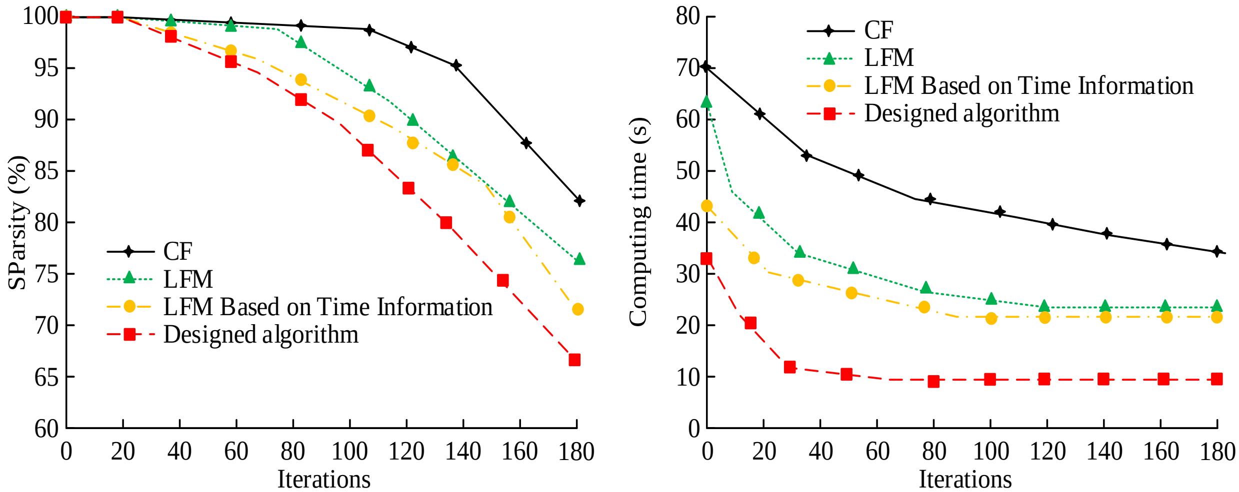
(a) Accuracy curve
(b) Recall rate curve
(c) F1 score curve

Fig. 7. Results of different indicators.

B. Application Effectiveness of LFM Recommendation Algorithm based on K-Means and Time Information

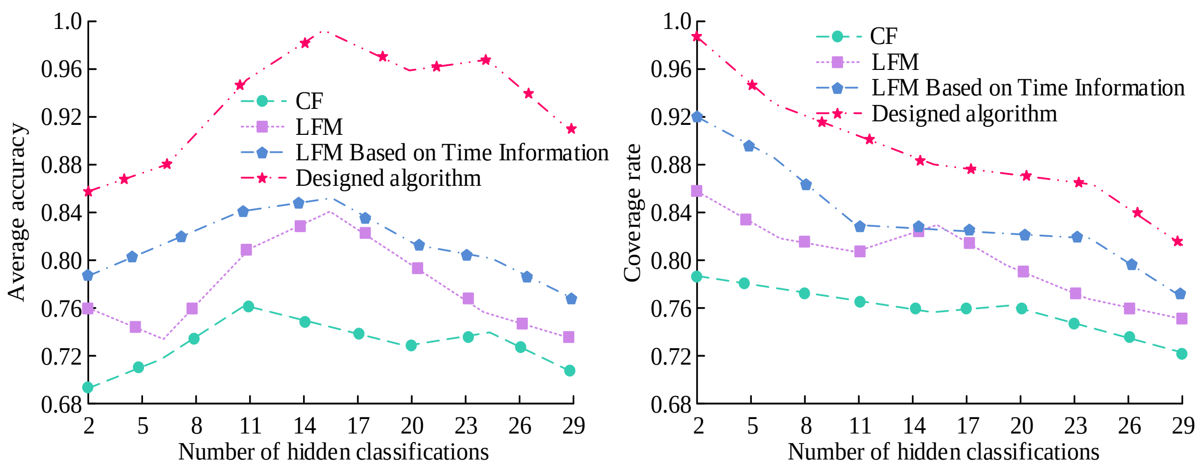
To verify the effectiveness of the designed BR in practical applications, a simulation experiment was conducted using Python 3.9 in a hardware environment with a 64 bit Windows 10 operating system, Intel (R) Core (TM) i5-12500K processor, 8GB of RAM, and 1TB of hard disk capacity. Firstly, the sparsity and computation time of the four algorithms are calculated. Fig. 8 shows the comparison results.

From Fig. 8 (a), as the iteration increased, the sparsity of different algorithms was effectively reduced. The sparsity reduction effect of the designed BR was significantly better than other algorithms, with a sparsity of 66.5% at 180 iterations. From Fig. 8 (b), the computation time for using all four algorithms for recommendation showed a decreasing trend and tended to stabilize after reaching a certain iteration. The calculation time when the designed algorithm reached stability was 10.2s. The above results demonstrate that the designed algorithm can better solve the data sparsity in BR, and also prove its high computational efficiency. The next step is to calculate the average accuracy and coverage of different algorithms separately, as displayed in Fig. 9.



(a) Sparsity of different algorithms (b) Calculation time of different algorithms

Fig. 8. Sparsity and computational time of different algorithms.



(a) Average accuracy of different algorithms (b) Coverage of different algorithms

Fig. 9. Average accuracy and coverage of different algorithms.

From Fig. 9 (a), the average accuracy of the designed algorithm was significantly higher than other algorithms, reaching a maximum of 98.7%, further proving its recommendation accuracy. From Fig. 9 (b), as the number of hidden classifications increased, the coverage trends of these four algorithms were consistent. However, the coverage curve of the designed algorithm had always been above that of other algorithms, indicating that the designed algorithm could cover more items in the recommendation process, proving its comprehensiveness and diversity. Finally, the average popularity of different algorithms was calculated to evaluate the effectiveness of the designed recommendation algorithm, as displayed in Fig. 10.

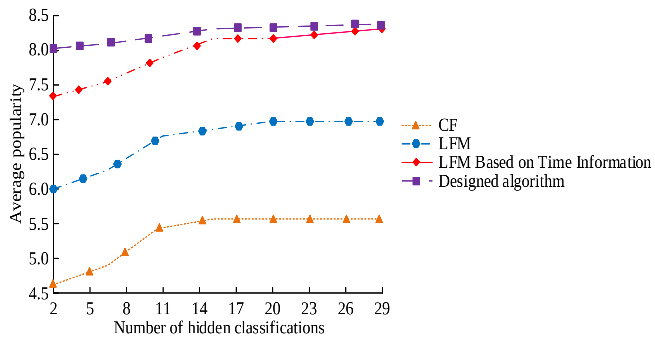


Fig. 10. Average popularity of different algorithms.

From Fig. 10, the average popularity of these four algorithms gradually increased and tended to stabilize. The average popularity of CF reaching a flat state was 5.5. The average popularity of LFM reaching a flat state was 6.8. The average popularity of LFM based on time information after stabilizing was 7.9. The average popularity of the designed recommendation algorithm after reaching stability was 8.2. The above results indicate that the book recommended by the designed algorithm has a high popularity, proving its good recommendation effect.

V. Discussion

With the progress of the times, the number of books has increased sharply, and the readership has become more diverse. Traditional BR methods have been unable to meet the diverse needs. The research aims to provide more accurate and personalized BR services, and proposes an LFM-BR algorithm that integrates time information and K-means clustering. The results showed that the MAE and RMSE values of the proposed algorithm were 0.21 and 0.23, respectively, which were higher than other algorithms, proving its high accuracy. Similar conclusions were drawn by Yi B et al. [5], who proposed an implicit feedback embedding deep matrix factorization model that can capture potential features of users, but it cannot effectively handle cold start problems. The proposed method quantifies the impact of time factors on user preferences by constructing a preference transfer function, effectively alleviating the cold start problem. The sparsity of the proposed algorithm was 66.5% at 180 iterations, indicating that it could better solve the data sparsity problem in BR. This conclusion is similar to the conclusion drawn by Fu M et al. [12], but the proposed algorithm is significantly better. This is because the proposed algorithm introduces the K-means algorithm to cluster readers with similar preferences and incorporates time information to enhance the system robustness, thus better addressing the data sparsity. The proposed algorithm took 10.2 seconds to reach a steady state, significantly lower than other algorithms, demonstrating its high computational efficiency. This is similar to the conclusion drawn by Zhou Y [11], and the proposed algorithm is superior. This is because the proposed algorithm optimizes the objective function and uses gradient descent to update parameters, ensuring efficient computation on large-scale datasets. In summary, the proposed method effectively improves the effectiveness of BR by integrating time information, optimizing clustering strategies, and enhancing LFM models, which is of great significance for promoting personalized library services.

VI. Conclusion

In the era of information explosion, BR has become an important tool to help users quickly find suitable reading interests. To improve the accuracy and personalization of BR, an LFM-BR based on K-means and time information was designed. Firstly, a comprehensive preference model based on time was constructed through borrowed books, followed by reader clustering using K-means, and finally training using LFM. These results confirmed that the maximum MAE of the designed algorithm was 0.29 and its minimum value was 0.21. The maximum RMSE value was 0.31 and its minimum value was 0.23, all higher than other algorithms, proving its high prediction accuracy. In terms of accuracy, recall, and F1 score calculation, the designed algorithm was 97.3%, 98.2%, and 0.965, respectively, proving its good overall performance. In terms of sparsity and computation time, the designed recommendation algorithm had a sparsity of 66.5% at 180 iterations, and a computation time of 10.2s to reach a stable state. These prove that it can effectively solve the data sparsity in BR and has high computational efficiency. The above results confirm that the designed algorithm has high accuracy and personalization in the BR problem, and can accurately reflect the reading interests and needs of users. However, the study does not consider other behavioral data of readers, which may have a certain impact on the results. Future research will incorporate more user behavior data to construct more comprehensive user preference models, and introduce new natural language processing techniques to further enhance the performance of BR systems.

References

[1] Wu S, Sun F, Zhang W, Xie X, Cui B. Graph neural networks in recommender systems: a survey. ACM Computing Surveys, 2022, 55(5): 1-37.

[2] Wu D, Shang M, Luo X, Wang Z. An L 1-and-L 2-norm-oriented latent factor model for recommender systems. IEEE Transactions on Neural Networks and Learning Systems, 2021, 33(10): 5775-5788.

[3] Nengem S M. Symmetric Kernel-Based Approach for Elliptic Partial Differential Equation. Journal of Data Science and Intelligent Systems, 2023, 1(2): 99-104.

[4] Guo Q, Zhuang F, Qin C, Zhu H, Xie X, Xiong H, He Q. A survey on knowledge graph-based recommender systems. IEEE Transactions on Knowledge and Data Engineering, 2020, 34(8): 3549-3568.

[5] Yi B, Shen X, Liu H, Zhang Z, Zhang W, Liu S, Xiong N. Deep matrix factorization with implicit feedback embedding for recommendation system. IEEE Transactions on Industrial Informatics, 2019, 15(8): 4591-4601.

[6] Cui Z, Xu X, Fei X U E, Cai X, Cao Y, Zhang W, Chen J. Personalized recommendation system based on collaborative filtering for IoT scenarios. IEEE Transactions on Services Computing, 2020, 13(4): 685-695.

[7] Zhou W, Han W. Personalized recommendation via user preference matching. Information Processing & Management, 2019, 56(3): 955-968.

[8] Liu Y, Xu T W, Xiao M. Small Data Fusion Algorithm for Personalized Library Recommendations. International Journal of Information and Communication Technology Education (IJICTE), 2023, 19(1): 1-14.

[9] Liu Y. Survey of Intelligent Recommendation of Academic Information in University Libraries Based on Situational Perception Method. Journal of Education and Learning, 2020, 9(2): 197-202.

[10] Zhang S. Personalized Service for University Library Users Based on Data Tracking. Voice of the Publisher, 2022, 8(2): 41-49.

[11] Zhou Y. Design and implementation of book recommendation management system based on improved Apriori algorithm. Intelligent Information Management, 2020, 12(3): 75-87.

[12] Fu M. The design of library resource personalised recommendation system based on deep belief network. International Journal of Applied Systemic Studies, 2023, 10(3): 205-219.

[13] Chendhur K M K, Priya V, Priya R M, Lakshmi S L. Book recommender system using improved collaborative filtering. International Journal of Research in Engineering, Science and Management, 2021, 4(4): 51-56.

[14] Anwar T, Uma V. CD-SPM: Cross-domain book recommendation using sequential pattern mining and rule mining. Journal of King Saud University-Computer and Information Sciences, 2022, 34(3): 793-800.

[15] Saraswat M, Srishti. Leveraging genre classification with RNN for Book recommendation. International Journal of Information Technology, 2022, 14(7): 3751-3756.

[16] Luo X, Yuan Y, Chen S, Zeng N, Wang Z. Position-transitional particle swarm optimization-incorporated latent factor analysis. IEEE Transactions on Knowledge and Data Engineering, 2020, 34(8): 3958-3970.

[17] Yi B, Shen X, Liu H, Zhang Z, Zhang W, Liu S, Xiong N. Deep matrix factorization with implicit feedback embedding for recommendation system. IEEE Transactions on Industrial Informatics, 2019, 15(8): 4591-4601.

[18] Choudhuri S, Adeniye S, Sen A. Distribution Alignment Using Complement Entropy Objective and Adaptive Consensus-Based Label Refinement For Partial Domain Adaptation//Artificial Intelligence and Applications. 2023, 1(1): 43-51.