ISSN: 2664-9640

Collaborative filtering algorithm and its application in personalized music recommendation

Huojiang Zheng, Botao Liu *, Xiong Xiao, Mingde Sun, Zizhuo Yan

Yangtze University, Jingzhou 434000, China

* Corresponding Author liubotao920@163.com

Abstract

In today's era of information boom, recommendation technology plays an important role in providing users with precision services and providing precision marketing for merchants. Taking the construction of an online music player as an example, the paper expounds the principle of the collaborative filtering algorithm, as well as the implementation flow chart, implementation steps and precautions of the algorithm in the music player, and finally gives the test results of the algorithm. From the test results, it can be seen that the recommendation algorithm implemented in the article runs basically normally and can achieve the goal of personalized recommendation based on user-based music.

Keywords

Collaborative filtering algorithm; Similarity calculation; Music recommendation system.

1. Introduction

In the Internet thinking, traffic is money, and various industrial thinking is ultimately to serve traffic, and traffic has become the lifeblood of Internet companies [1] Internet companies that monetize through traffic have gradually become the darlings of the new era. The great success of products such as Douyin, Kuaishou, and Today's Headlines is the best proof. Many of the above manufacturers make use of the user's behavior data as a parameter for the recommendation algorithm, and the recommendation algorithm delivers the items that the user may like and recommends to the user. Today, the mature recommendation algorithms include: collaborative filtering algorithm[2], recommendation algorithm based on metadata and collaborative filtering and fusion, k-means clustering recommendation algorithm, and collaborative filtering recommendation algorithm with improved cosine similarity for correction [5] etc. .

The good use of the music personalized recommendation algorithm helps users save the time of actively searching for music, expands the user song library while avoiding the embarrassing situation where users have no songs to listen to, and also fully saves network traffic resources.

2. Overview of user-based collaborative filtering algorithm

2.1. Algorithm implementation process and principles

Based on the user's collaborative filtering algorithm (CF), the main consideration is the similarity between the user and the user, as long as the other users who are similar to the target user are found, and the target user pair is predicted according to the songs that the similar user likes With the rating of the corresponding song, you can find the highest rated song collection recommended to the user. The entire recommended flow path can be described as shown in Figure 1.

ISSN: 2664-9640

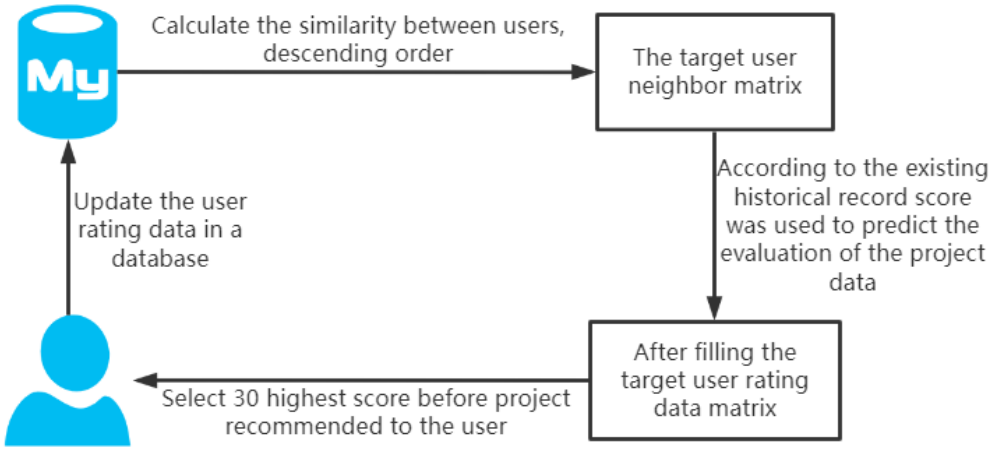

Figure 1 Recommended flow path graph based on the user's collaborative filtering algorithm

The above figure is described as follows: First, the existing user rating records of the songs are used to calculate the similarity between users; then, the resulting similarity results are sorted in descending order, the user with high similarity is taken as the target user's nearest neighbor, the target user's near neighbor matrix is generated, and the target user's near neighbor matrix is used to predict the target user's rating of the unrated items Finally, select several of the top items in the prediction score to feedback to the user as recommendation results.

2.2. Algorithm example analysis

As it is shown in Table1 “User & Song” .If the user1 likes songs 1 and 2 while user2 also likes songs 1 and 2， It can be preliminarily determined that the degree of similarity between user1 and user2 is high. Given that the user1 hasn’t listen to the song3 yet, user2’s favorite song3 is to be recommended to user1.

Table 1 User & Song

Users/Songs	Song 1	Song 2	Song 3		Song n
User 1	liked	liked	recommended	...	
User 2	liked	liked	liked	...	like
……					
User m		liked		...	

(1) Construct all the user rating data of the song in the dataset into a "user-item" scoring matrix. where the number of users m is the number of matrix rows, and the number of songs n is represented as the number of matrix columns.

$$R_{mn} = \begin{bmatrix} R_{11} & \cdots & R_{1n} \\ \vdots & \ddots & \vdots \\ R_{m1} & \cdots & R_{mn} \end{bmatrix}$$

(2)Calculates the similarity between users.

(3) Generate predictive scores. Using sum to represent the average rating of the user, according to the Sim matrix can get the similarity with k near neighbor, while using the near neighbor to t score value, the final resulting target user score prediction formula as shown in formula (1).

$$Pre(u_a, t) = \bar{r_a} + \frac{\sum_{b \in k} Sim(u_a, u_b)\left(r_{b,t} - \bar{r_b}\right)}{\sum_{b \in k} Sim(u_a, u_b)} \quad (1)$$

ISSN: 2664-9640

(4) Descend the predicted values of each user and make recommendations.

3. The core part of the algorithm implements the solution selection

Calculating the similarity between users is a core part of the algorithm, which directly determines the accuracy of the final recommendation result.

There are four common ways to measure similarity between users: Euclidean distance, cosine similarity, and modified cosine similarity. These three ways are described in the following.

3.1. Euclidean Distance

Euclidean distance is used to calculate the degree of similarity between users, comparing the distance between two vectors, as shown in Equation (2).

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$|X| = \sqrt{\sum_{i=1}^{n} {x_i}^2} \quad (2)$$

where, euclidean distance $d(x,y)$ between $(x_1, x_2, \ldots, x_n)$ $(y_1, y_2, \ldots, y_n)$ points and points; $|X|$Euclidean distance from point to origin in n-dimensional space$(x_1, x_2, \ldots, x_n)$. For two users with similar musical tastes, if one of the users has a more conservative rating, they both like the same song very much, and user A gives 5 points and user B gives 4 points. Using the distance formula, the two users may not be neighbors, although their musical tastes are very similar.

3.2. Cosine similarity

The calculation of Cosine Similarity solves the problem posed by the above calculation of "Euclidean distance", which does not calculate the distance between the two vectors, but compares their angles, and the smaller the angle, the more similar, It is used to measure the similarity between users in the recommendation system. Suppose two close neighbor users, A and B, have taste preference vectors of a and b, then they can pass The calculation of cosine similarity to reflect the degree of similarity between user A and B, calculated by formulas such as Equation (2)：

$$sim_{ab} = \cos(a,b) = \frac{a * b}{|a| \times |b|} \quad (3)$$

3.3. Modified cosine similarity

Modified cosine similarity [10] is a traditional similarity measurement method, because it itself takes into account the factor of scoring scale when calculating user similarity, which can avoid the scoring bias caused by scoring habits, make the measurement similarity more reasonable, and more accurately excavate the similar users of the target user.

Correction cosines are similarly calculated as follows:

$$Sim(a,b) = \frac{\sum_{t \in R_{ab}}(R_{a,t} - \overline{R_a})(R_{b,t} - \overline{R_b})}{\sqrt{\sum_{t \in R_a}(R_{a,t} - \overline{R_a})^2}\sqrt{\sum_{t \in R_b}(R_{b,t} - \overline{R_b})^2}} \quad (4)$$

Whereas R_{ab} represents a common scoring item, R_a and R_b is the set of users a and b each scoring, $\overline{R_a}$ and $\overline{R_b}$ represents the average rating of users a and b respectively.

ISSN: 2664-9640

3.4. The choice of the implementation

For the above-mentioned 3 similarity algorithms, the first algorithm by finding the Euclidean distance between two users, to show the similarity between users, the advantage is that the calculation is simple and fairly convenient, but the disadvantages are also very obvious, for two users with similar taste, through the distance formula calculation but they are not alike, which is seriously inconsistent with the facts, so in the actual work, this method is rarely used; Cosine similarity uses the cosine value of the angle between two vectors in the vector space as a measure of the size of the difference between two individuals, compared with the distance measure, the cosine similarity pays more attention to the difference in the direction of the two vectors, rather than the distance or length; although the cosine similarity can be corrected to a certain extent on the bias between individuals, but because only individuals can be distinguished between dimensions The difference, there is no way to measure the difference in the value of each dimension.

Euclidean distance can reflect the absolute difference of individual numerical characteristics, so it is more practical for analysis that needs to reflect the difference from the numerical size of the dimension, such as using user behavior indicators to analyze the similarity or difference of user value; while the cosine similarity is more from the direction to distinguish the difference, and not sensitive to the absolute value, more used to use the user's content score to distinguish the similarity and difference of user interest, Also fixed a possible metric inconsistency between users (because cosine similarity is not sensitive to absolute values). Therefore, for the implementation of the music recommendation algorithm in this paper, it is of course more appropriate to use cosine similarity to calculate the similarity between users.

4. Application and implementation of collaborative filtering algorithm in personalized music recommendation system

4.1. The implementation process and steps

After the user logs into the personalized music recommendation system, the system will sample the data of the user's behavior record. Collect the behavior information data of users downloading, playing and collecting favorite songs, analyze and form a user preference portrait matrix, and then calculate the similarity between other users in the user pool and the target users and generate a recommendation list, and the implementation process is shown in Figure 2.

The implementation steps are described as follows:

Start the system and perform various initialization operations.

When a user logs in to an account, the system determines whether the login is successful, and if it fails, ends the system.

After successful login, the system collects the user's downloads, plays, and favorite songs and stores them in the system memory for subsequent use.

Based on the data obtained in Step3, the user feature portrait is calculated, representing its characteristics, as the basis for similarity calculation.

Iterates over all users in the user pool to perform a comparison calculation of the similarity with itself, and stores the calculation result of the similarity in the system memory.

Sort from highest to lowest according to step5's calculation results, and sample a certain amount of songs for each user as a recommendation result according to the threshold rule.

The system ends, freeing all memory.

ISSN: 2664-9640

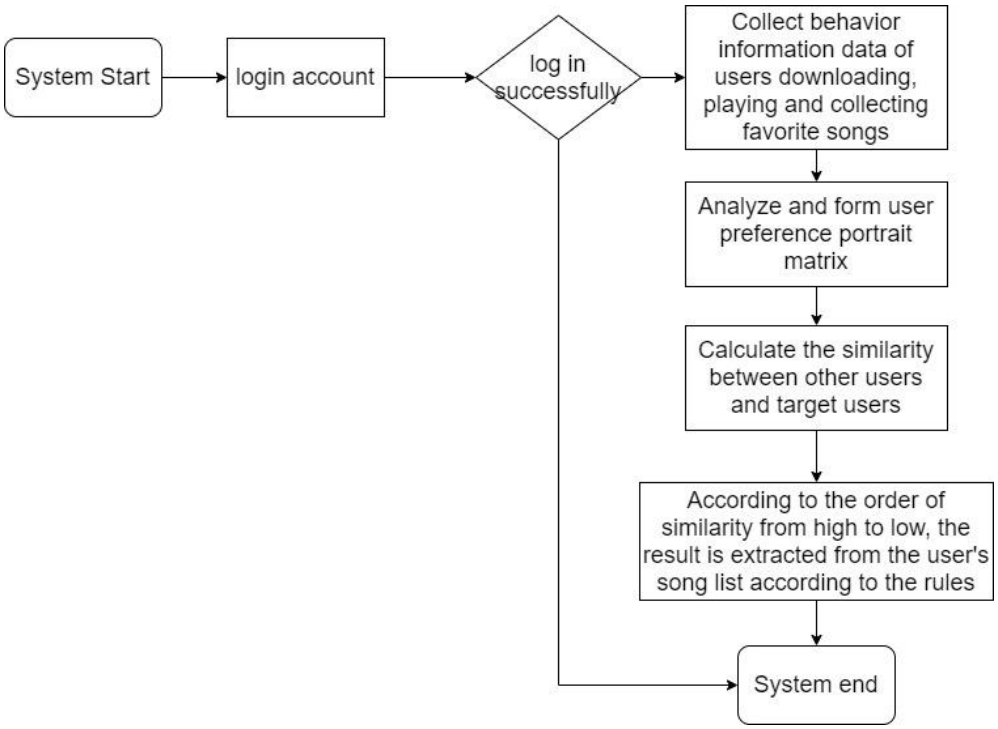

Figure 2 Flowchart of the implementation of the collaborative filtering algorithm in the personalized music recommendation system

4.2. Interface design

The recommended algorithm implements the interface and related class UML are shown in Figure 3

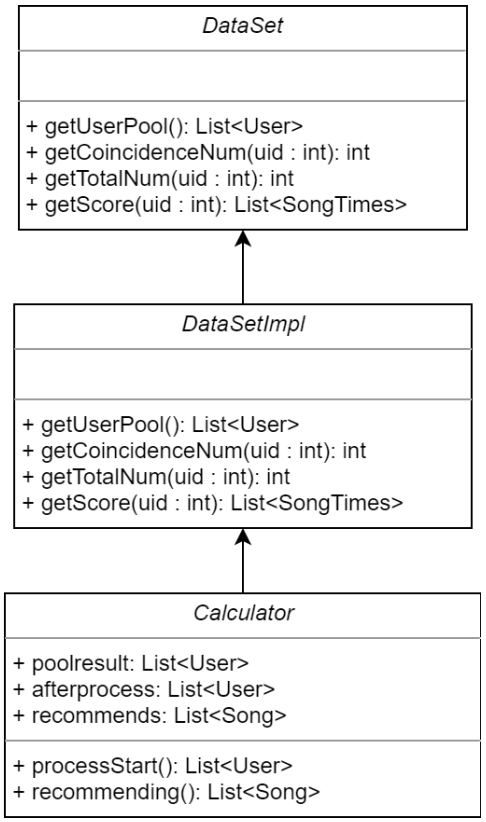

Figure 3 Interface design diagram of collaborative filtering algorithm in personalized music recommendation system

ISSN: 2664-9640

1) class DataSetImpl implementation interface description

The usage of class DataSetImpl is to build a "user-song" scoring matrix based on calculating the user's song playback frequency and the number of overlapping songs, and its main functions are introduced as shown in Table 2.

Table 2 Function description in DataSetImpl

Function signature	The meaning of the return value	The meaning of the parameter	The usage of the function
List<User>getUserPool()	Returns the user pool obtained	not	Get the user pool around the user's own for subsequent traversal
Int getCoincidenceNum(int uid)	Returns the number of song coincidences between users	The user ID of the target user	Gets the number of overlapping songs of the target user and themselves as a basis for similarity
Int getTotalNum(int uid)	Returns the total number of songs in the target user's favorites	The user ID of the target user	Gets the total number of songs in the target user's collection
List<SongTimes> getScore(int uid)	Returns an array list of a user's listening frequency scores	The user ID of the target user	Gets the listening frequency score of the target user at all times

2) Calculator class implementation interface description

The Calculator class inherits dataSetImpl, so the class can get the underlying data using all the functions in the DataSet Calculator itself implements the main recommendation algorithm logic, which is the main class of user similarity calculations. The member variables and member functions of its class are described below.

Table 3 Meanings of member variables in the Calculator class

Member variables	The variable type	The data stored by the variable
poolresult	List<User>	After getting to the user pool around the user, it is stored here
afterprocess	List<User>	The return value of the processStart function obtains the user pool after descending similarity
recommends	List<Song>	The return of the return function by the protocol function is worth getting the song to be recommended to the user

ISSN: 2664-9640

Table 4 Calculator class member functions are described

Function signature	The meaning of the return value	The meaning of the parameter	The function of the function
List<User> processStart()	Returns a sorted list of users	not	The initial user pool is processed, and a list of users is returned according to the algorithm, arranged from highest to lowest according to similarity
List<Song> recommending()	Returns a playlist with recommended songs	not	According to the user pool returned by processStart, the data in n users that meet the permutation rules are obtained

5. System test process and results

5.1. Test Example 1

For the above music recommendation system, the test steps and results are explained in the following.

First, according to the characteristic data of the logged-in user, the user pool around it is obtained;

Get the number of songs that the user likes coincide with the target user, here you can directly get the number of coincidences between the target user and your favorite songs through the crawler, as shown in Figure 4.

Figure 4 Schematic diagram of the user's favorite song coinciding with the target user

Calculate the proportion of the number of overlaps to the total number of songs of the user and record them into an array list;

After calculating the scale values of all users, arrange them from high to low, and traverse them in turn;

For each traversed user, the crawler obtains the frequency of listening to the song at all times, as shown in Figure 5.

ISSN: 2664-9640

▼{,…}
▼allData: [{playCount: 0, score: 100, song: {name: "sky -crossroads version-", id: 509728740, pst: 0, t: 0,…}},…]
▶0: {playCount: 0, score: 100, song: {name: "sky -crossroads version-", id: 509728740, pst: 0, t: 0,…}}
▶1: {playCount: 0, score: 95, song: {name: "crescendo -version2016-", id: 432821976, pst: 0, t: 0,…}}
▶2: {playCount: 0, score: 89, song: {name: "you only live once", id: 1373214360, pst: 0, t: 0,…}}
▶3: {playCount: 0, score: 88,…}
▶4: {playCount: 0, score: 88, song: {name: "one dream", id: 1373214359, pst: 0, t: 0,…}}
▶5: {playCount: 0, score: 82, song: {name: "as before", id: 1397635376, pst: 0, t: 0,…}}
▶6: {playCount: 0, score: 81, song: {name: "change your core self", id: 1397635374, pst: 0, t: 0,…}}
▶7: {playCount: 0, score: 79, song: {name: "white forces -IS3 edition-", id: 432821975, pst: 0, t: 0,…}}
▶8: {playCount: 0, score: 77,…}
▶9: {playCount: 0, score: 76, song: {name: "beyond the horizon", id: 1373212685, pst: 0, t: 0,…}}
▶10: {playCount: 0, score: 75,…}
▶11: {playCount: 0, score: 75, song: {name: "frosty breeze", id: 1397635371, pst: 0, t: 0,…}}
▶12: {playCount: 0, score: 72, song: {name: "snow of silence", id: 1373212681, pst: 0, t: 0,…}}
▶13: {playCount: 0, score: 69, song: {name: "endless entropy", id: 1397635370, pst: 0, t: 0,…}}
▶14: {playCount: 0, score: 66, song: {name: "only me and the moon", id: 509728751, pst: 0, t: 0,…}}
▶15: {playCount: 0, score: 65, song: {name: "believe in your future", id: 1397634388, pst: 0, t: 0,…}}
▶16: {playCount: 0, score: 64, song: {name: "brand new world", id: 1397634387, pst: 0, t: 0,…}}
▶17: {playCount: 0, score: 63, song: {name: "《perpetual wishes～ , id: 551502895, pst: 0, t: 0,…}}
▶18: {playCount: 0, score: 63, song: {name: "Like a blink, a short night.", id: 1373214362, pst: 0, t: 0,…}}
▶19: {playCount: 0, score: 58, song: {name: "adverse wind", id: 1373212683, pst: 0, t: 0,…}}
▶20: {playCount: 0, score: 58,…}
▶21: {playCount: 0, score: 57,…}

Figure 5 Tests the user's listening frequency at all times

It should be noted that what is shown in the above figure is not the user's listening frequency, but in the form of a score, which is because the data we crawled came from NetEase Cloud, which chose to record scores instead of playing times to protect user privacy. In order not to affect the results of our algorithm calculations, the corresponding processing is carried out in step6 online.

The score of the above figure takes the average as its threshold N (its threshold will be adjusted later according to the number of user pools), and the songs that exceed the threshold are returned as recommended songs. Here we will get the songs recommended to the target user from the most similar n user playlists obtained, so when the number of n increases, the value of N should increase. That is to say, the value of n is proportional to the value of N.

After passing the above calculations, the returned playlist is obtained, as shown in Figure 6.

1	Nympho	03:48	Christopher	Told You So
2	Re:make	03:23	ONE OK ...	Re:make / NO S...
3	Never Gonna Give You Up	03:32	Rick Astley	Simply The Best ...
4	Glad You Came	03:15	Boyce Av...	Cover Sessions, ...
5	Oops	03:24	Little Mix/...	Glory Days (Delu...
6	one dream	05:06		
7	white forces -IS3 edition-	06:39	fripSide	infinite synthesis 3
8	1983-schwarzesmarken- (IS3 vers...	05:23	fripSide	infinite synthesis 3
9	sky -crossroads version-	05:23	fripSide	crossroads

Figure 6 The playlist returned during the test

5.2. Test Example 2

In Test Example 2, the test user will be replaced, intended to verify that the songs recommended by different users are different. Follow the calculation steps above to perform it again.

First obtain the song that the user coincides with the target to calculate its similarity, as shown in Figure 7. After all users in the pool have been calculated, they are sorted in descending order according to similarity.

ISSN: 2664-9640

Figure 7 Schematic diagram of overlapping songs

For each traversed user, the crawler obtains the frequency of listening to the song at all times, as shown in Figure 8.

{allData: [{playCount: 0, score: 100,…}, {playCount: 0, score: 65,…},…], code: 200}
allData: [{playCount: 0, score: 100,…}, {playCount: 0, score: 65,…},…]
0: {playCount: 0, score: 100,…}
1: {playCount: 0, score: 65,…}
2: {playCount: 0, score: 55, song: {name: "Unshaken", id: 1372315031, pst: 0, t: 0,…}}
3: {playCount: 0, score: 49, song: {name: "Modern Crusaders", id: 1297870, pst: 0, t: 0,…}}
4: {playCount: 0, score: 46,…}
5: {playCount: 0, score: 42,…}
6: {playCount: 0, score: 40, song: {name: "Wrong (feat. Emilia Ali)", id: 512358397, pst: 0, t: 0,…}}
7: {playCount: 0, score: 40, song: {name: "Salt", id: 500686106, pst: 0, t: 0,…}}
8: {playCount: 0, score: 40,…}
9: {playCount: 0, score: 38, song: {name: "Tropic Love", id: 31514328, pst: 0, t: 0,…}}
10: {playCount: 0, score: 38,…}
11: {playCount: 0, score: 37, song: {name: "Pray For Me", id: 536870488, pst: 0, t: 0,…}}
12: {playCount: 0, score: 37, song: {name: "", id: 468015459, pst: 0, t: 0,…}}
13: {playCount: 0, score: 34, song: {name: "Love & War (Original Mix)", id: 440208433, pst: 0, t: 0,…}}
14: {playCount: 0, score: 34, song: {name: "Angetenar (Emre Kabak Remix)", id: 1432160937, pst: 0, t: 0,…}}
15: {playCount: 0, score: 32, song: {name: "The cyborg fights", id: 41632422, pst: 0, t: 0,…}}
16: {playCount: 0, score: 32, song: {name: "Fearless Pt. II", id: 529391087, pst: 0, t: 0,…}}
17: {playCount: 0, score: 32, song: {name: "Starboy", id: 431610014, pst: 0, t: 0,…}}
18: {playCount: 0, score: 31, song: {name: "Freek'n You (Radio Edit)", id: 32922629, pst: 0, t: 0,…}}
19: {playCount: 0, score: 31, song: {name: "The Chain", id: 473602853, pst: 0, t: 0,…}}
20: {playCount: 0, score: 31, song: {name: "canzoni preferite", id: 1334778977, pst: 0, t: 0,…}}
21: {playCount: 0, score: 31, song: {name: "All Falls Down", id: 515453363, pst: 0, t: 0,…}}
22: {playCount: 0, score: 31,…}

Figure 8 Schematic diagram of the frequency of listening to songs (scoring).

Perform the calculation according to the above Step6 to get the return song list as shown in Figure 9

Figure 9 All recommended songs obtained

From the above test results, it can be seen that the online music system that implements the recommendation algorithm can recommend different songs according to different users, and the recommendation algorithm function works normally.

ISSN: 2664-9640

6. Conclusion

After an online music system implements a recommendation algorithm, it will be upgraded from a static ordinary music system to a personalized music recommendation system that can truly achieve dynamic service for people. On the one hand, it can provide users with more accurate services and increase the stickiness of users; on the other hand, it can serve the precision marketing strategy of merchants and increase the use of system platform traffic. Therefore, the implementation of recommendation algorithms in online music systems has practical significance and certain economic value.

The collaborative filter recommendation algorithm introduced in the article is a relatively simple recommendation algorithm, its essence is to summarize the characteristics of the user according to the user behavior data, and use this feature to calculate the similarity between users, if I and a user have a very similar taste, then the user has heard and I have not heard the song can be recommended to myself, which is the collaborative filter recommendation algorithm introduced in this article according to the user.

Of course, if the degree of user liking the same song is different, then it will inevitably cause errors in judging the similarity between two users, which is not considered in the text and needs to be improved later.

Acknowledgments

This research is supported by five projects: The demonstration construction projects of “Ideological and Political Course and Ideological and Political Course” in Yangtze University in 2021(No.41);The project of young people in the Education Hall of Hubei(No.Q20161311);The Yangtze Youth Fund(No.2015cqn53);The Yangtze University Students' Innovation and Entrepreneurship Training Program Project(No. Yz2021124).

References

[1] Shen Yuhang, Xiao Wen, Li Tian Exploration of New Modern Internet Hot Traffic Thinking[J]. China Business Theory, 2019(12):13-14+21.

[2] Hua Ze, Ye Yuhang Collaborative filtering algorithm and its application in personalized music recommendation[J]. Modern Computer, 2021, (22): 43-46+54.

[3] Cai Haidong, Zhan Haoxuan, Shu Zhimin, Guo Tianxiong Music recommendation algorithm based on metadata and collaborative filtering and fusion[J]. Information and Computers (Theoretical Edition), 2021, 33(23): 51-54.

[4] ZHANG Chao,GUO Xiujuan,ZHANG Kunpeng. K-means algorithm cluster center selection [J]. Journal of Jilin University (Information Science Edition),2019,37(04):437-441.].

[5] Chu Honglin, Liu Qicheng, Mou Chunxiao Collaborative filter recommendation algorithm for improving cosine similarity[J]. Journal of Yantai University (Natural Science and Engineering Edition), 2021, 34(03): 330-336.

[6] Liu Jianguo, Zhou Tao, Wang Binghong Research progress of personalized recommendation system[J]. Advances in Natural Science,2009,19(01):1-15.].

[7] Liang Xiangyang, Zhang Bolun Classification analysis and discussion of recommended technology for collaborative filtration[J]. Computer and Modernization,2016,(12):67-72.]

[8] ZHANG Chao,GUO Xiujuan,ZHANG Kunpeng. K-means algorithm cluster center selection [J]. Journal of Jilin University (Information Science Edition),2019,37(04):437-441.].

[9] Hua Ze, Ye Yuhang Collaborative filtering algorithm and its application in personalized music recommendation[J]. Modern Computer, 2021, (22): 43-46+54.

[10] Wang Hailong,Liu Lin,Lin Min,Pei Dongmei. Music Personalized Recommendation Algorithm Based on Information Retrieval and K-Means Clustering[J]. Journal of Jilin

ISSN: 2664-9640

University (Engineering Science),2021,51(05):1845-1850.].
[11] Ren Lei Research on key technologies of recommended systems [D]. East China Normal University, 2012.