2018 International Conference on Sensor Network and Computer Engineering (ICSNCE 2018)

Design and Implementation of Music Recommendation System Based on Hadoop

Zhao Yufeng
School of Computer Science and Engineering
Xi'an University of Technology
Shaanxi, Xi'an, China
e-mail: zyfzy99@163.com

Li Xinwei
School of Computer Science and Engineering
Xi'an University of Technology
Shaanxi, Xi'an, China
e-mail: 604013466@qq.com

***Abstract*—In order to solve the problem of information overload of music system under large data background, this paper studies the design scheme of distributed music recommendation system based on Hadoop. The proposed algorithm is based on the MapReduce distributed computing framework, which has high scalability and performance, and can be applied to the calculation and analysis of off-line data efficiently. The music recommendation system designed in this paper also includes client, server interface, database and ETL operation, which can calculate a set of complete recommendation system from user operation end to server and data calculation. In order to improve the accuracy of the recommendation algorithm, this paper introduces k-means clustering algorithm to improve the recommendation algorithm based on user-based collaborative filtering.The experimental results show that the accuracy of the proposed algorithm has been significantly improved after the introduction of k-means.**

Keywords-Music Recommendation; K-means Clustering; Collaborative Filtering; Recommendation Algorithm; Hadoop

I. INTRODUCTION

With the development of the mobile Internet, the amount of data generated by mobile APP has increased rapidly in recent years. On July 27, 2017, Trustdata, a well-known mobile big data monitoring platform in China released the "Analysis Report of China Mobile Internet Development in the First Half of 2017" [1]. Among them, mobile music as a high-frequency application, the number of the user be showed a steady growth in the first half of 2017, peak DAU (daily active users) nearly 150 million. Taking Netease cloud music as an example, since its client APP was launched in April 2013,the number of users has reached 300 million, song orders 400 million. Such a huge amount of data makes the traditional single-server data storage and processing becomes more and more clumsy. Moreover, it is more and more difficult for users to find their favorite songs in huge amounts of data. After all, favorite songs are only a few, and finding them one by one can be very difficult. The past practice is that when users want to listen to some songs, they search for them by engines, but only find songs they have known. A lot of songs that users do not know, but will probably like very much, will never be heard. If there is a system specifically for users to push songs, users will spend less time searching for songs, and the stickiness from users to the system will increase. Based on this, this paper uses Hadoop, a big data computing and storage framework, to store and calculate data, so as to solve the problem in finding songs, and also improve user's activity and stickiness. The Hadoop-based recommendation system used in this paper has the following important implications in today's Internet context:

1) To effectively solve the problem of "information overload", to provide users with interesting content by exploring the relationship between users and songs;

2) Hadoop cluster parallel computing and distributed storage technology has good scalability, can effectively handle massive data storage and computing problems;

3) For the enterprise, the system with the recommended function can enhance the user experience and increase the user's activity and stickiness.

II. ALGORITHM BASIS

A. Traditional user-based collaborative filtering recommendation algorithm

The traditional user-based collaborative filtering recommendation algorithm is based on several users that are most similar to this user's interest, and then recommends songs that the user has not heard from these similar users. The specific method is to find each user by user similarity algorithm some of the most similar users, and then summarize the similar user listening records, from which the target user to find out the songs have not been heard, and then similar users to sort the similarity to obtain The preference of each song, the order of these songs, so as to target users recommend [2]. Specific algorithm steps are as follows:

1) Construct users - song data representation matrix. The row vector represents the user, and the column vector represents each song. The matrix value indicates whether the user has heard of the song, 0 means that it has not been heard, and 1 means it has been heard. It is a 0-1 matrix.

2) Generate the nearest neighbor set. According to the user-song matrix, the similarity algorithm is used to calculate the user's similarity so as to find the user set closest to the target user. Formula (1)[2] calculates the similarity of two users.

$$w_{uv} = \frac{\sum_{i \in N(u) \cap N(v)} \frac{1}{\log(1+|N(i)|)}}{\sqrt{|N(u)||N(v)|}} \quad (1)$$

3) Produce recommendations. The recommendation value of a song that a similar user has heard while the target user has not heard is determined by the similarity of the user with the target user. A user set must have many users that have heard the same song, and this song's recommended value is the sum of the similarities of these users. In this way, songs that have not been heard by the target user can be sorted according to the recommended values, and songs with high recommendation values will be preferentially pushed to the target user. Equation(2) [2] calculates the user's preference for song i.

$$p(u,i) = \sum_{v=S(u,K) \cap N(i)} w_{uv} r_{vi} \quad (2)$$

B. Introducing the k-means algorithm to optimize the traditional recommendation algorithm

Clustering is an unsupervised learning method which aggregates data with similar attributes on the basis of nomanual labeling. It combines data for similarities and the data in the same group have similarities. The data in the different group is different from each other. The improved collaborative filtering recommendation algorithm in this paper is based on the user base with high similarity, so that the algorithm has a better recommendation effect. Similar calculation directly restricts the effectiveness of the clustering effect and thus affects the final recommendation result. The principle of the algorithm is as follows: Suppose there is a group of users User, the total number of users is m, remember to be U (U_1, U_2, U_3, ..., U_m), each user U_x has n attributes, recorded as C_x (C_{x1}, C_{x2}, C_{x3},..., C_{xn}), the principle of clustering is to compare each attribute on the basis of the set U and divide into the groups of similar users [3]. The core idea of the k-means algorithm is to divide a given group of users into k groups. Within each k group, a cluster center is set to calculate the distance of each data from the center. The minimum distance is attributed to this group.

C. k-means clustering algorithm improvement

1) The removal of free points [3]. Of all the data points, the free points are those that are far away from all other points, and their existence will cause the deviation of the center point in the belonging class and thus the classification effect. The process of removing free points in this paper is as follows:

Let the total number of users be m, the total number of paths of all users to other users be calculated according to the formula (3):

$$L = \frac{m \times (m-1)}{2} \quad (3)$$

Then the sum of all users is:

$$D = \frac{1}{2} \sum_{i=1}^{m} \sum_{j \neq i} gap(C_i, C_j) \quad (4)$$

$$gap(C_i, C_j) = \sqrt{(C_{i1} - C_{j1}) * (C_{i2} - C_{j2}) * \ldots\ldots * (C_{in} - C_{jn})} \quad (5)$$

In formula (5), C_{i1}, C_{j1}, ...C_{in}, C_{jn} are n attributes of users C_i and C_j. Find the distance average from L and D, formula (6)

$$EMV = \frac{L}{D} \tag{6}$$

Formula (6) is the average of all user distances. For each user U, compute the distance U from all other users $L_u = (U_i, U_j)$. If all $L_u >$ = EMV, the user is classified as a free point, a separate category. If the number of free points is small, they can't be classified into a category for collaborative filtering, so you can classify it to a class that is closest to a type of center point.

2) The selection of random points will also have an impact on the classification results, and fall into the local optimal solution. In order to solve this problem, the method adopted in this paper is to adopt the improved algorithm of clustering: dichotomous clustering [4]. The idea of dichotomous clustering is to first classify all the points as a cluster and then divide it into two (k = 2 k-means clustering), and then select the class that can minimize the clustering cost function again and divide it into two formula until the number of clusters equals k. The cluster cost function is defined as the sum of squared error of the cluster, as shown in formula (7). The largest square error sum of a class, this kind of point in the distance from the center of the maximum distance, you need to be divided once again.

$$Ei = \sum_{p \in C_i} |p - c_i|^2 \tag{7}$$

III. RECOMMENDED SYSTEM DESIGN

This section describes the implementation and testing of the entire system and what frameworks are included in each of the system's features. First of all, introduce the top-level design; then analyze the overall framework of the system, what technologies are needed, and the overall process of the system; finally, we evaluate the recommended results from the accuracy and recall rate of recommendations.

A. Recommended process

The K-means clustering-based collaborative filtering recommendation algorithm proposed in the previous section is mainly divided into two steps, one is to use k-means to implement user clustering, and the other is to perform a recommendation algorithm based on user collaborative filtering thus generating the recommended result.

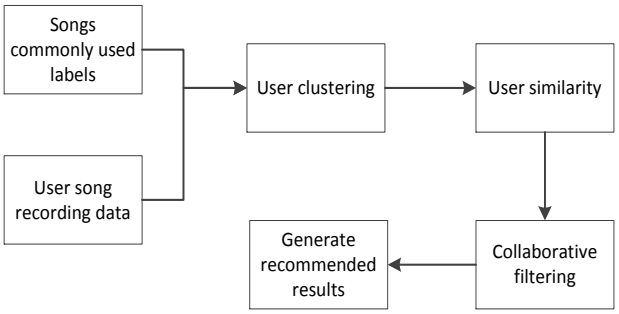

Figure 1. Distributed recommended algorithm flow

Algorithm parallelization of the flow chart is shown in figure 1. Clustering algorithm is divided into three steps. The first step is to create a user tag model: user log table and song list of commonly used tags through a step MapReduce process to generate user-tag model, the tag file as a cache file for each user's song recording tag statistics in the tag vector Increase the number of each position to generate a user label matrix. The second step is to use k-means algorithm to calculate the cluster center point of user label matrix. After several iterations, the relatively stable center point of each cluster is determined. The third step reads the central point file as a cache, in order to classify users, by calculating the user from which one of the nearest center, put this user into which cluster. The user-based collaborative filtering algorithm will recommend collaborative filtering for users in each cluster.This step is divided into a number of steps, including counting the number of a song being listened to, counting the number of a user listening to music, calculating user similarity, generating recommended results.

B. Recommended system architecture design

The recommended system is divided into the recommended algorithm layer, server-side and client. The recommended algorithm layer uses Hadoop cluster for distributed recommendation, the server side uses the Java Servlet and MySQL database for development, and the client side uses Android for display[5]. Log collection is

collected using the ETL tool Sqoop. System overall framework is shown in Figure 2.

It can be seen from the figure that the recommended system is divided into 4 major parts: Top-level client, server, database and Hadoop layer. The client uses the Android system for display. Server development uses Java EE development. Database uses MySQL. MySQL is the intermediate data link between the client and the recommendation system. Data is stored in the database so that front-end servers and Hadoop are decoupled. The transfer of data between the database and Hadoop takes the Sqoop. Sqoop is a distributed log collection system based on Hadoop, which requires Hadoop to run, which can also speed up data transfer. The collected data is stored in HDFS. The Hadoop layer is divided into two parts, one is HDFS data storage, and the other is MapReduce distributed computing. MapReduce reads the data from HDFS and calculates it, and then saves the results back to HDFS. Sqoop reads data from the HDFS and transfers to MySQL, the client requests the server at irregular intervals, the server reads data from the database and returns it to the client, and the user's operation of songs is fed back to the server and saved to the database[6].

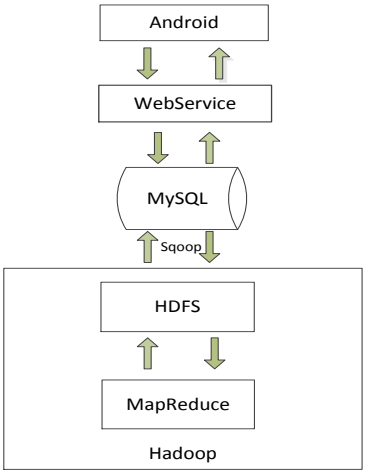

Figure 2. Recommended system architecture

C. *Recommended system function module analysis*

When users use the recommendation system, the system recommends different songs to different users, which requires users to log in the system with different user names. In order to log in, the system also has to provide the user registration function. To listen to different songs, users must also be able to search for songs and add tags to the songs. Because we have no songs to play, we also need a collection or a favorite feature for users[7]. The server side designs the corresponding interfaces based on these requirements, and the database also designs different tables to store the content. Accordingly, we have designed the function diagram of the system, as shown in Figure 3.

As can be seen from the figure, the system function is decomposed into four modules. Clients have user registration, login, search songs, tag songs and play features. Play actually refers to favorites or likes. Because the song involves copyright issues, it can only be replaced by favorites or likes. These functions correspond to the server-side interface to provide data. Data is read from the database, so the database is designed with three tables: the user table, used to register and log in; the table for recording users' listening to songs, with the largest amount of data; song table storage song's basic information, including tag information, search songs and add tags[8].

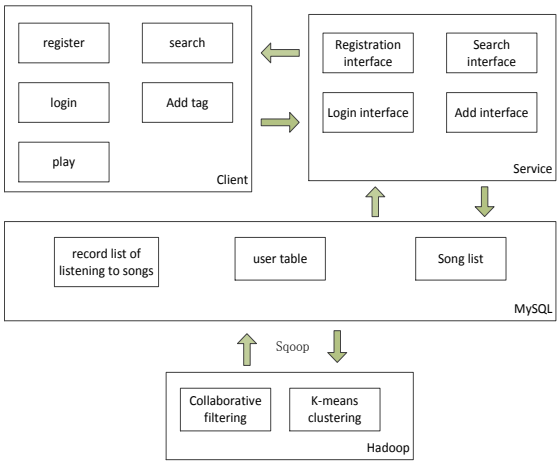

Figure 3. Recommended system function block diagram

IV. DESIGN OF HADOOPBASED RECOMMENDATION SYSTEM

Algorithm parallelization is based on the k-means clustering algorithm introduced in the previous chapter and user-based collaborative filtering recommendation algorithm is implemented in the distributed system, the distributed

recommendation algorithm is divided into two modules, one is distributed k-means Clustering algorithm, one is the distributed collaborative filtering recommendation algorithm, and finally the two algorithms connected to achieve the work of the entire distributed recommendation algorithm[9].

A. K-means clustering of parallel design

The data input of the K-means algorithm is a matrix, because here the user is clustered, so the user-label matrix is first formed by data preprocessing. User listening records are large files and need to be designed in parallel. Listener recorded data into the HDFS behind the addition of each song's label, the formation of (user id, song id, song time, label) this format records. Then use MapReduce to parallelize the user-label matrices in one step. Then the user-label matrix is clustered. The clustering and parallelization algorithm is designed as follows: The first step scans all the original points and randomly selects k points as the center point of the first step cluster. The second step is to calculate the cluster of all the points to each center point, and to point each point to the closest center point cluster. The third step to repeat the second step to meet the termination conditions[10]. The fourth step is to calculate all the user points, the user assigned to their respective clusters. Algorithm architecture is shown in Figure 4.

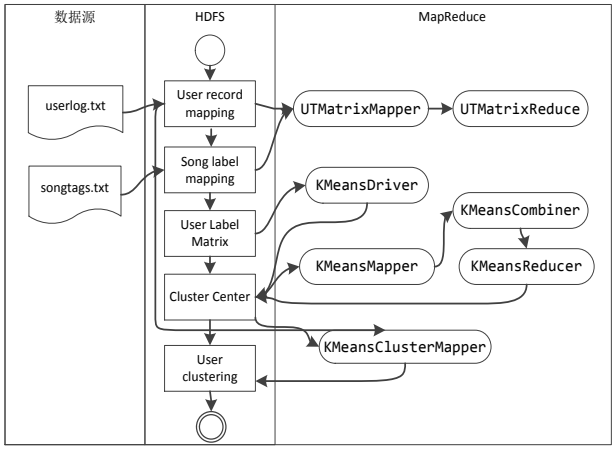

Figure 4. Distributed k-means clustering algorithm architecture

B. Concurrent Design of User-based Collaborative Filtering Recommendation Algorithm

According to the formula (1) and (2) in the first section, combined with the rules of Hadoop distributed design, the following steps are designed for the recommendation algorithm: The first step is to count how many songs each user listens to; the second step is to count the number of times each song is heard; the third step is to calculate the similarity of every two users, but only the similarity of the two users who have heard the same song need to be computed; the fourth step, is to calculate each user's recommending value for each song to form a recommend list; The final step is to sum up the recommending values for a user listening to the same song. The above steps are implemented based on MapReduce, and the entire work flow requires a number of Map and Reduce to complete. Figure 5 shows the MapReduce architecture based on the user collaborative filtering algorithm.

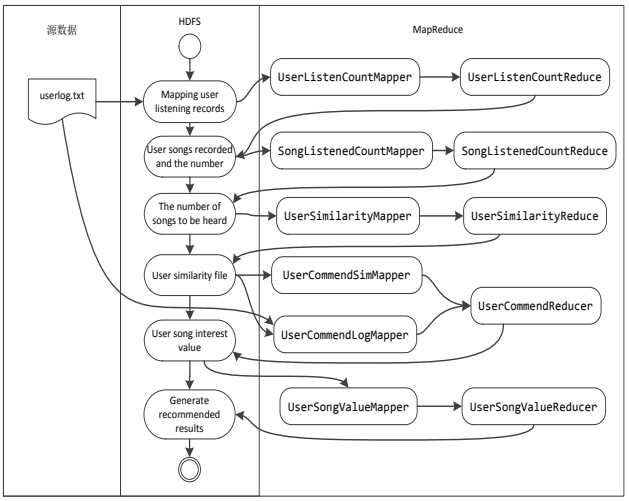

Figure 5. Distributed architecture based on user collaborative filtering algorithm

V. EXPERIMENTS AND EVALUATION

In the same situation as the data-set, many times of repeated experiments were done on the traditional user-based collaborative filtering algorithm, the recommendation algorithm after introducing K-means clustering and the recommendation algorithm after the improved clustering. And the stable results were selected to analyze [11].

A. Experimental environment and data set

Environment construction and configuration include Hadoop cluster, Sqoop, development environment and Web server. The cluster environment is built using VMware virtual machines.

The cluster is built using three nodes, one Master and two Slaves.

- Host configuration Hardware environment: CPU Intel i5-4590, quad-core, 3.30 GHz, RAM 8 GB;
- Software Environment: OS Centos7, java Environment jdk1.8, server tomcat8.0, hadoop 2.7.4;
- Development environment: Eclipse, windows10, hadoop.plugin;
- Music data used in this experiment are from the network, including more than 50,000 users, more than 1,700,000 user operations, and 730 tags.

B. *Results Show*

Figure 6 shows a screenshot of the recommended results for the Android phone.

Figure 6. Android mobile music recommended results

C. *Evaluation Indices*

The Precision and Recall [12] are used as evaluation indices.Each user's song date is sorted in descending order, with the top 80% as the training set, and the remaining 20% as the test set for experimental evaluation.

$$\text{Precision} = \frac{\text{The correct number of recommended results}}{\text{The number of recommended results}} \times 100\% \quad (8)$$

$$\text{Recall} = \frac{\text{The correct number of recommended results}}{\text{The number of users like}} \times 100\% \quad (9)$$

Formula (8) and formula (9) are calculated formulas for the Precision and Recall respectively.

D. *The line contrast chart of three algorithms*

After repeated experiments that the precision of three algorithms changes with the K-value is shown in line chart as Figure 7:

As shown in Figure 7, the precision after introducing the K-means clustering algorithm is better than the traditional collaborative filtering algorithm, and when the K value is 4, the precision is increased by about 0.65%, which is the best classification. After the clustering algorithm is improved, the precision is nearly 0.15% higher than unimproved when the K value is 5.

Figure 8 is the Recall change chart for three algorithms. The Recall of K-means clustering algorithm is better than that of the traditional collaborative filtering algorithm, and when the K value is 4, the Recall can increase about 0.65%. When the K value is 5, the Recall increases nearly 0.15% after the clustering algorithm is improved; but as the K value increases, the improved clustering is lower than the unimproved clustering algorithm. Because the user number is fixed, after removing the free point, each classification will be affected. Some classification number is very few, then the recommended result is inaccurate, thus the whole effect of the recommendation algorithm is affected.

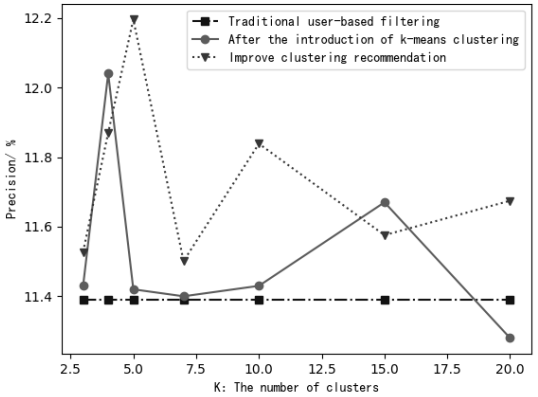

Figure 7. The change of precision with K value

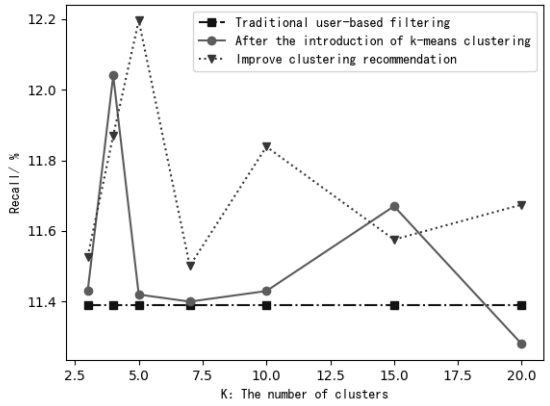

Figure 8. The change chart of recall with K value

VI. CONCLUDING REMARKS

This paper puts forward the design and implementation of the Hadoop-based music recommendation system, and improves the traditional user-based collaborative filtering recommendation algorithm, and improves the precision of the recommendation algorithm by using the song tag for users clustering. Hadoop, a scalable and high-performance distributed computing platform, provides a reference for the design of a music recommendation system in the background of large data. The recommendation algorithm of K-means clustering and collaborative filtering is designed based on MapReduce distributed framework, which has some reference significance for the distributed design of the recommendation algorithm.

REFERENCES

[1] Trustdata. China Mobile Internet Development Analysis Report for the First Half of 2017 [EB/OL]. (2017-07-30). http://itrustdata.com/#service

[2] Xiang Liang Recommended system practice[M].BeiJing:People Post Press,2012.6:1-3Zhang Xin-sheng Zhang Hai-ying Mao Qian.Hadoop[EB/OL].(2016-12-19).https://baike.baidu.com/item/Hadoop/3526507

[3] Wu Hongchen, WangXinjun,ChengYong,PengZhaohui. Advanced Recommendation Base on Collaborative Filtering and Partition Clustering[J].Computer Research and Development,2011,48(S3):205-212.

[4] ZhengJie.Machine Learning Algorithm Principles and Programming Practice[M].BeiJing.Electronic Industry Press,2015.10:141

[5] Weston J, Bengio S, Hamel P, et al. Large-Scale Music Annotation and Retrieval: Learning to Rank in Joint Semantic Spaces.[J]. arXiv: Learning, 2011.

[6] Den Oord A V, Dieleman S, Schrauwen B, et al. Deep content-based music recommendation[C]. neural information processing systems, 2013: 2643-2651.

[7] Su J, Chang W, Tseng V S, et al. Personalized Music Recommendation by Mining Social Media Tags[J]. Procedia Computer Science, 2013: 303-312.

[8] Avidson J, Liebald B, Liu J, et al. The YouTube video recommendation system[C].conference on recommender systems, 2010: 293-296.

[9] FENG Ya-li,JIANGJie,TianFeng.Research on the combined recommendation algorithm based on item and user[J].Information Technology,2017,(10):69-73.

[10] CHANG Xiao yu,YU Zheng sheng.Point-of-interest Recommendation Algorithm Introducing Time Attenuation Item[J].Journal of Hangzhou Dianzi University(Natural Sciences),2016,36(03):42-46. [2017-09-19]. DOI : 10.13954/j.cnki.hdu.2016.03.009

[11] Zhao Z, Shang M. User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop[C]. knowledge discovery and data mining, 2010: 478-481.

[12] Chen Yaxi.Music recommendation system and related technologies [J]. Computer Engineering and Applications,2012,48(18):9-16+47.