International Journal of Smart Business and Technology
Vol.6, No.1 (2018), pp.15-30
http://dx.doi.org/10.21742/ijsbt.2018.6.1.02

Research on the Design of E-Commerce Recommendation System

Daniel Yang[1] and Saman Grice[2*]

[1,2]*Deakin University, Australia*
[2*]*saman.grice@deakin.edu.au*

Abstract

In recent years, the amount of data in various fields, especially in the e-commerce industry, has continued to rise, inseparable from the rapid development of information technology and massive data processing technology applications. Using recommendation systems is very important in order to use vast amounts of data to serve users and promote user retention. Based on a hybrid recommendation idea, this paper designs an e-commerce recommendation system based on big data technology. First, the system needs analysis, and the overall architecture of the system is designed. Then, the design process of each recommended module is explained in detail. This article combines a demographic-based recommendation algorithm, content-based recommendation algorithm, ALS-based collaborative filtering recommendation algorithm, and model-based real-time recommendation algorithm to form a hybrid recommendation module to provide users with recommendation services. Among them, the demographic-based recommendation can solve the user's cold start problem, the collaborative filtering recommendation does not require the content attributes of the item to be recommended, and the model-based recommendation algorithm can provide users with real-time recommendation services.

Keywords: *E-commerce, Recommendation system, Recommendation algorithm, Collaborative filtering*

1. Introduction

Information technology has maintained a rapid development trend since its inception, and the development of Internet technology has brought closer relationships between people. As time goes by, the data in the network becomes more and more complicated, and the problem of information overload becomes more and more serious. How to obtain useful information from massive amounts of data has become an issue of concern. The recommendation system was invented to solve this problem. After years of development, the recommendation system has had relatively successful experience in fields including movie recommendation, music recommendation, social networking, and e-commerce.

The recommendation algorithm is the cornerstone of the recommendation system. At present, researchers have proposed a variety of different recommendation algorithms to adapt to different recommendation scenarios. However, no recommendation algorithm can solve all

Article history:

Received (November 16, 2020), Review Result (December 21, 2020), Accepted (January 8, 2021)

*corresponding author

Print: 2288-8969, eISSN: 2207-516X, IJSBT

the problems. Hybrid recommendation can effectively alleviate the shortcomings of a single recommendation method by integrating the strengths of multiple recommendation methods and learning from each other.

The improvement of recommendation algorithms is inseparable from the application of big data processing technology, so massive data processing technology becomes more and more adaptable to the needs of recommendation systems with the development of parallel computing ideas. Hadoop, an open-source framework that can perform distributed processing of massive amounts of data, has been sought after by people since its birth with its advantages of low cost, high scalability, and high fault tolerance [1]. However, Hadoop's core computing framework, MapReduce, must store the results in a distributed storage system every time it passes the calculation process. Therefore, the reciprocating cycle will involve huge network transmissions, which will consume time and reduce the system. Spark improves on this. Spark is a distributed batch-processing engine based on memory for computing. Its biggest features are low latency and iterative computing. Unlike Hadoop, Spark is more suitable for data processing, machine learning, and interactive analysis. It also has higher development efficiency and better fault tolerance, so it is often used in complex recommendation scenarios.

E-commerce recommendation systems [2] can provide users of e-commerce websites with an intelligent and personalized shopping experience, making users tend to buy more goods, thereby improving user satisfaction. The recommendation algorithm is the core part of the recommendation system. Currently, commonly used recommendation algorithms can be divided into collaborative filtering recommendation algorithms, demographic-based recommendation algorithms, and content-based recommendation algorithms based on different data sources.

The Group lens research group of the University of Minnesota in the United States applied the idea of collaborative filtering recommendation to the movie recommendation system Movie Lens in 1994 [3]. The successful application of this system in the recommendation field laid the foundation for the commercialization of the recommendation system. In 1997, Resnick and Varian first proposed the definition of a recommender system [4]: A recommender system is a software system that provides users with recommendations for purchases through simulated shopping guides to help users decide which products to purchase. In 2006, Netflix, the influential movie rental website at the time, held a movie recommendation system competition [5] and offered a reward of millions of dollars for the team that could design the best recommendation system. In the end, the team that proposed a model-based collaborative filtering recommendation algorithm

won this game's champion. Ghemawat S and others proposed a file system called GFS [6], a distributed file system prototype. Jeffrey Dean and Sanjay Ghemawat proposed the Map Reduce programming model [7]. Hadoop saves the intermediate results of distributed computing MapReduce on disk. When data increases rapidly, the system will have obvious performance bottlenecks. The Spark distributed memory computing framework designed by the University of California Berkeley AMP Lab solves this problem well. Spark caches the data in memory until the model uses Map and reduce functions to process merged key-value pairs so the program can run on the cluster. Subsequently, Yahoo engineer Doug Cutting created an open-source project, the Hadoop framework, based on a research report released by Google Labs. This project became a sensation. Doug Cutting also became the father of Hadoop and led the team to improve it continuously. Hadoop saves the intermediate results of distributed computing MapReduce on disk [8]. When data increases rapidly, the system will have obvious performance bottlenecks. The Spark distributed memory computing framework designed by

the University of California Berkeley AMP Lab solves this problem well. Spark caches the data in memory and writes the data to disk until the final result is calculated.

2. Research theories and methods

2.1. Big data technology

Hadoop was originally a project of the Apache open-source community, mainly divided into three parts: distributed file storage system HDFS, distributed offline computing framework MapReduce, cluster resource management, and scheduling system Yarn. The core is HDFS and MapReduce, and Yarn is a new module added in Hadoop 2.0 [9]. The biggest advantage of Hadoop is that it helps users use the cluster's high-efficiency development programs and complete storage and calculations. Even if they do not understand the complex implementation details of the underlying architecture, it will not impact the development process. After nearly two decades of development, Hadoop has developed into the most popular open-source distributed framework with its reliable, efficient, and scalable characteristics [10].

The Hadoop ecology, in a broad sense, refers to open-source components or products related to big data technology. In addition to the HDFS, Map Reduce, and Yarn already mentioned, there are also structured distributed data warehouse Hive, unstructured distributed data warehouse HBase, and distributed applications. Program coordination service Zookeeper, log collection tool Flume, general computing engine Spark, distributed message queue Kafka, etc. The Hadoop ecosystem architecture is shown in Figure 1.

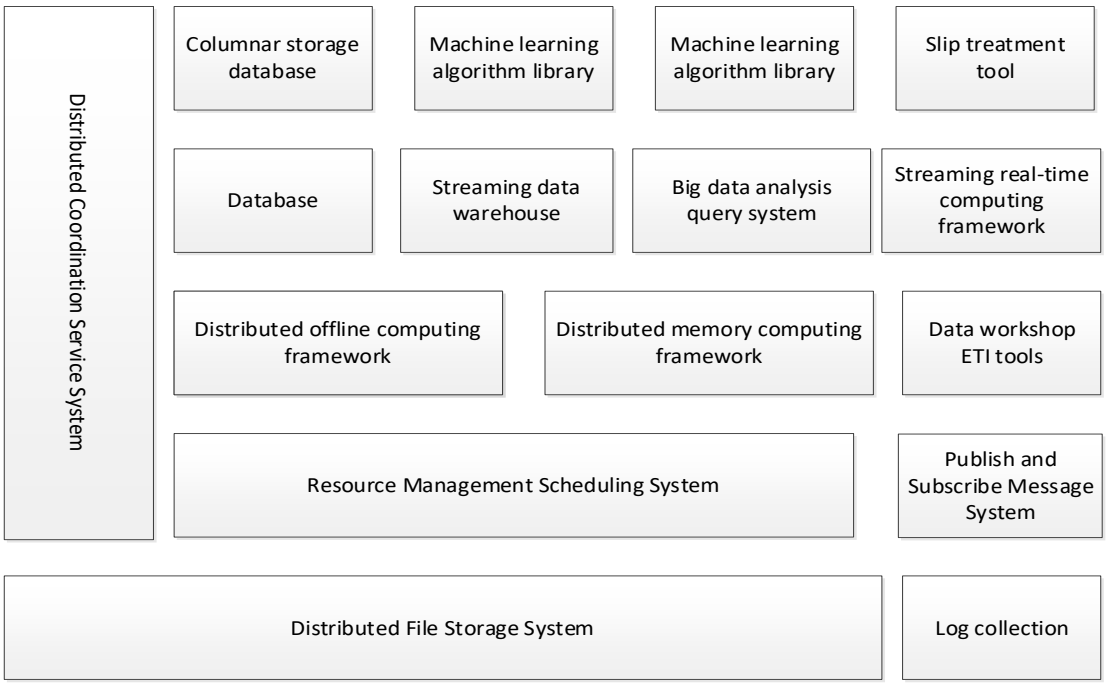

Figure 1. Hadoop ecosystem architecture diagram

The components in the ecosystem will depend on each other, but they are independent. Different elements can be selected and matched according to business needs. With the

continuous development of Hadoop ecosystem technology, some old components may be replaced by new ones.

The structural model of the distributed file system has the characteristics of "master/slave," where the "master" refers to the name node (Name Node) in the cluster. It has only one, the master node of HDFS, which is mainly responsible for managing HDFS. The namespace in the cluster maintains all file directory information in the entire file system in the namespace. And "slave" refers to several data nodes (Data Node). The data node is responsible for providing the storage of each file; it is the working node of the file system. The data node stores and retrieves data according to the schedule of the master node and sends it to the master node regularly. In HDFS, data is divided into many small blocks and stored in different nodes. The default size is 128MB, and to prevent data loss, each data block will save several copies by default and then unified scheduling management through Name Node [11].

MapReduce is a programming framework mainly oriented to extensive data parallel processing scenarios. MapReduce is mainly composed of two parts: Map and Reduce. Map analyzes and processes data into key and value pairs (key, value) and sends it out. Currently, the calculation is run in parallel on different machines, and the subsequent data passes through a series of sorting partitions. After processing, it still reaches the Reduce stage through key-value pairs. In this stage, the data is summed and stored in the file system [12].

Spark is a memory-based big data calculation and processing framework. Its birth stems from the optimization of MapReduce disk read and write. It has fast running speed, good ease of use, strong versatility, and can run anywhere. Spark is implemented in the Scala language. Scala is simple to operate, faster than scripting languages, and runs on JVM, making it easier to be compatible with big data frameworks. After years of research and development, Spark has gradually established its ecosystem, including Spark Core for offline data analysis, Spark Streaming for real-time data stream processing, structured data processing module Spark SQL, machine learning library Spark MLlib, and distribution Graphic processing framework Graphx and other important components. As shown in Figure 2, it is the Spark ecosystem diagram.

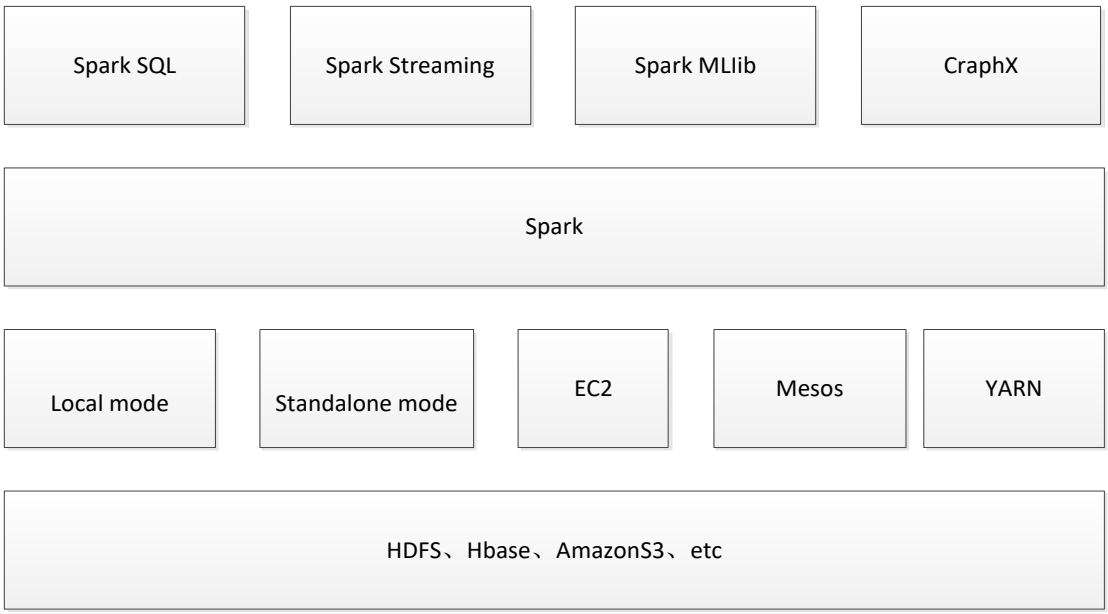

Figure 2 Spark ecosystem diagram

Spark Core: Spark Core is the core of Spark. The cluster reads data from the persistence layer and then uses different components to complete calculations. Spark introduces the data abstraction of RDD, a collection of read-only objects that supports parallel operations and has high fault tolerance.

Spark SQL: Spark SQL is used to manipulate structured data. It allows developers to directly process RDD to query external data stored in the database. Spark SQL supports the simultaneous processing of relational tables and RDDs, which enables developers to perform complex data analysis while using SQL commands for external queries.

Spark Streaming: Spark Streaming [13] is a data processing system. The core idea is to decompose stream computing into a series of short batch processing jobs, and RDD will be used as an intermediate result throughout the calculation process. The calculation engine can superimpose or store the RDD directly on an external device.

Spark MLlib: Spark MLlib [14] implements some commonly used machine learning algorithms, such as classification, regression, clustering, and collaborative filtering. Developers only need to have a certain theoretical knowledge to perform machine learning work, which reduces the threshold of learning [15].

Spark can provide users with comprehensive solutions, connect with other platforms, and use resource scheduling managers YARN, Zookeeper, etc., to complete parallel computing.

2.2. Recommendation system

The recommendation idea based on demographics is relatively simple. It searches for people with tastes similar to the user based on the demographic information of the system user. Then, it recommends the items liked by these selected people (excluding the products the recommended user has purchased) to target users [16]. The system first constructs a user feature vector based on the user's basic information (gender, age, height, etc.). Then, it selects an appropriate method to calculate the similarity between users according to the value of the feature vector. The similarity here is generally calculated using Euclidean distance. The formula is as follows:

$$sin(x,y) = \frac{1}{1+d(x,y)}, d(x,y) = \sqrt{\sum_{i=1}^{\pi}(x_i - y_i)^2} \quad (1)$$

The advantage of this demographic-based recommendation mechanism is that there is no cold start problem for new users. For a new user, this algorithm does not need to know what products the user likes to recommend, and this method does not need to collect item information, only user data, so for items in different fields, both can be used. But this algorithm also has disadvantages. First, because the user's personal information is privacy-sensitive, it is not easy to obtain. Secondly, this recommendation method is rough and more suitable for recommendation to new users.

The content-based recommendation algorithm [17] suits text fields like news recommendations. This algorithm requires the input of the metadata content of the item, finds the inner connection between the item or content, and finally makes recommendations for the user based on the items that the user liked in the past. Content-based recommendation algorithms have been applied to book recommendations, music recommendations, etc., and have achieved good results. Some websites also specially invite staff to tag. Since content-based recommendations are derived from item data information, users can intuitively understand the reasons for the recommendation, have strong interpretability, and have no cold start problems. On the other hand, the content-based recommendation algorithm relies on item tags' content, making it difficult to analyze information such as pictures, audio, and video.

Collaborative filtering is currently the most widely used and most mature recommendation algorithm. Its basic idea is to discover what kind of products the user likes by collecting the behavior data of the target user, grouping the users according to the obtained preference information, and recommending the user's Commodities close to their taste. Collaborative filtering algorithms are currently mainly divided into neighbor-based collaborative filtering recommendation and model-based collaborative filtering algorithms. The specific classification is shown in Figure 3.

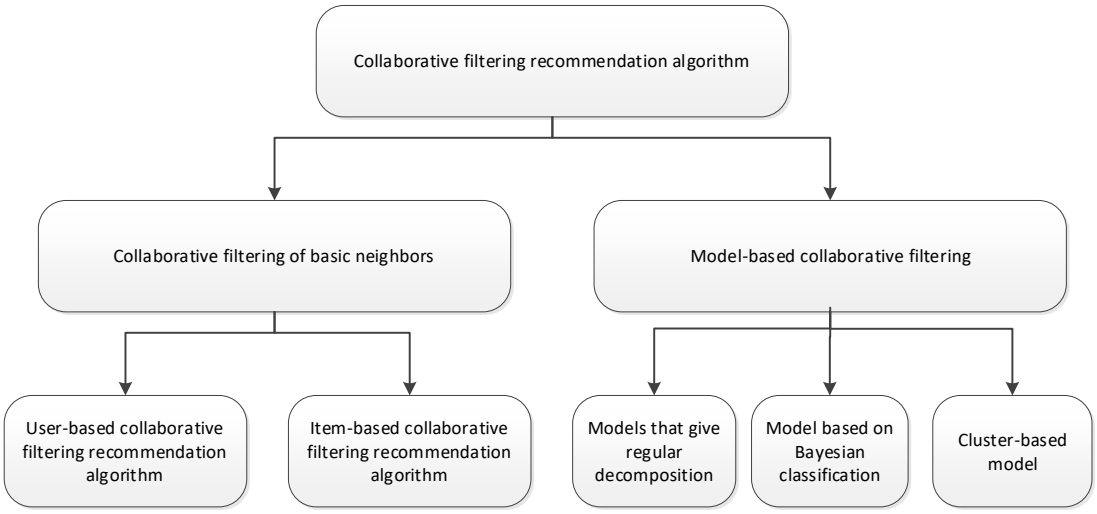

Figure 3. Classification of collaborative filtering recommendation algorithms

A single recommendation method may not necessarily be used directly in actual recommendation scenarios, and there are often different data sources. To make the recommendation system adapt to various application scenarios more quickly and efficiently, the hybrid recommendation idea is becoming more and more popular. There are currently several standard hybrid methods.

Waterfall style: This method is like flour sieving. The recommended results obtained are relatively vague when the first algorithm is introduced. The following recommendation method is used to advise based on the previous recommendation result, and the cycle continues. Different algorithms are screened layer by layer according to their granularity until the final recommendation result is obtained.

Weighted type: First, determine several recommendation methods that need to be used and select different parameters for each recommendation algorithm according to the weight; that is to say, the weights of various algorithms are different in the mixed recommendation algorithm. Then, the system needs to compare multiple times to determine whether the predicted value of the hybrid algorithm is the same as the actual result and select the most suitable parameters.

Partition mixing: This method is often used by e-commerce websites because it has the smallest restrictions on the type and number of recommendation algorithms used in mixed recommendations, and it can allocate recommendation results to different regions to recommend to users, which is very suitable for e-commerce Complex recommended scenes.

Conversion type: Also called switching type hybrid, it will adopt different recommendation mechanisms according to the situation. This requires users to have accurate judgment, such as using content-based recommendations when encountering a cold start and selecting

collaborative filtering algorithms when the amount of behavioral data is sufficient, so this method is sometimes more complicated.

A sound recommendation system allows users to obtain satisfactory product recommendations quickly, and it also makes it easier for the recommendation platform to get user resources. The following are several commonly used recommendation system evaluation indicators.

The premise of scoring prediction is that the system can get the user's historical scoring data. For example, many websites provide a scoring function. Then, the system establishes a model based on the existing data to predict the scoring of items the user has not bought. If the score is high, the product can be recommended. Based on this, if we know the real score data before, we can measure the recommendation effect by comparing the predicted and real values. There are two ways of evaluation: root mean square error (RMSE) [18]: RMSE measures the recommendation accuracy by taking the difference between all the actual scores and the predicted scores, as shown in formula 2.

$$RMSE = \sqrt{\frac{\sum_{u,i\in T}(r_{ui}-\widehat{r_{ui}})^2}{|T|}} \tag{2}$$

Among them, u represents the user, I represents the item, r_{ui} represents the user's true rating of the item, $\widehat{r_{ui}}$ represents the predicted rating, N represents the total rating, and the average absolute error (MAE): MAE and RMSE have the same idea, but the method used is different, MAE uses the absolute value to process the difference after obtaining the difference, as shown in Equation 3.

$$MB = \frac{\sum_{u,i\in T}|r_{ui}-\widehat{r_{ui}}|}{|T|} \tag{3}$$

The same as the root mean square error, u represents the user, I represents the item, r_{ui} represents the user's actual rating of the item, $\widehat{r}$ epresents the predicted rating, and N represents the total rating.

TopN recommendation: Generally, when a website provides recommendations to users, it often displays a personalized recommendation list, which is a TopN recommendation. The list recommends N products that users are most likely interested in. Accuracy (precision), recall (recall), and F-Measure value are usually used as evaluation indicators [19].

The accuracy rate indicates the ratio of the recommended products to the total recommended products. The definition of accuracy is as follows:

$$Precision = \frac{\sum_{u\in U}|P_{(u)}\cap T_{(u)}|}{\sum_{u\in U}P_{(u)}} \tag{4}$$

Where u is the user, u is the user set, and $P_{(u)}$ indicates the contents of all recommended products, $T_{(u)}$ denotes the commodity content in the test set.

The recall rate represents the ratio of the recommended products to all the products that should be recommended, as defined in formula 5.

$$Recall = \frac{\sum_{u\in U}|P_{(u)}\cap T_{(u)}|}{\sum_{u\in U}|T_{(u)}|} \tag{5}$$

In general, accuracy and recall do not increase or decrease simultaneously. To balance the two indicators, we use the harmonic mean of Precision and Recall, the F-Measure value, to judge the two. F-Measure It is defined as follows:

$$F - Measure = \frac{2 \times Presion \times Recall}{Precision + Recall} \quad (6)$$

Among them, Precision represents the accuracy rate, and recall represents the recall rate.

2.3. E-commerce

E-commerce is a business activity in which networks, information, and electronics are all linked to traditional industries. Common e-commerce can be divided into three main modes: Business-to-customer (B2C) between enterprises and consumers and business-to-business (B2B) between enterprises and consumers. E-commerce between consumers (Customer to Customer, or C2C). Nowadays, e-commerce has greatly saved the time and space of customers and enterprises by allowing consumers to complete shopping, payment, and other functions through the Internet. The e-commerce recommendation system has played an important role in this process.

3 Design of e-commerce recommendation system

3.1. Demand analysis

According to the division of users and system functions, the main tasks of users of this system include: users can log in to the system by entering a user name and password. Provide registration function for new users; the system provides unique identification for each user; some basic information is stored there. After the user enters the homepage, he can see the products recommended by the system for the user, presented in the form of a list, and the user can click on any product to browse details according to his preferences. On the product detail page, users can score or tag the product.

The following functions need to be implemented in the system: data storage function module, which processes massive scoring and tag data through the big data platform and completes offline recommendation display. The algorithm module starts after the system receives the data and updates the recommendation results in real time. When a new user logs in for the first time, the system can only provide non-personalized product recommendations for the user.

The most important thing in an e-commerce recommendation system is the accuracy and timeliness of the data. The system needs to process massive amounts of data and is accompanied by a mixture of multiple recommendation algorithms. Therefore, the recommendation system has extremely high requirements for data accuracy and fast response time. It is embodied in the following aspects: First, the system must have good stability. While it can guarantee the long-term, uninterrupted operation of the system, it can quickly recover from server disorder caused by specific emergencies. This is also a distributed architecture. Second, the system runs faster than under normal circumstances, which is the basic requirement for a big data computing platform. Finally, the system should be highly fault-tolerant, adopt redundant data storage, automatically save multiple copies of data, and be able to redistribute failed tasks automatically.

As an intermediate medium between the user and the system, the interface provides convenience for user operations. The user only needs to get the recommended information

through the familiar interface and does not need to understand the complex operating logic of the system. This article has the following requirements for the design of the system's operation interface:

(1) The operation page is clear, the functions are easy to find, and the recommended products should not be too many.

（2） The system page mainly includes the user login registration, recommendation, and product detail pages. The functional modules displayed in it should be as clear and layered as possible.

（3） The page's content should encourage users to do more operations, such as scoring and tagging, to facilitate information collection for later recommendations.

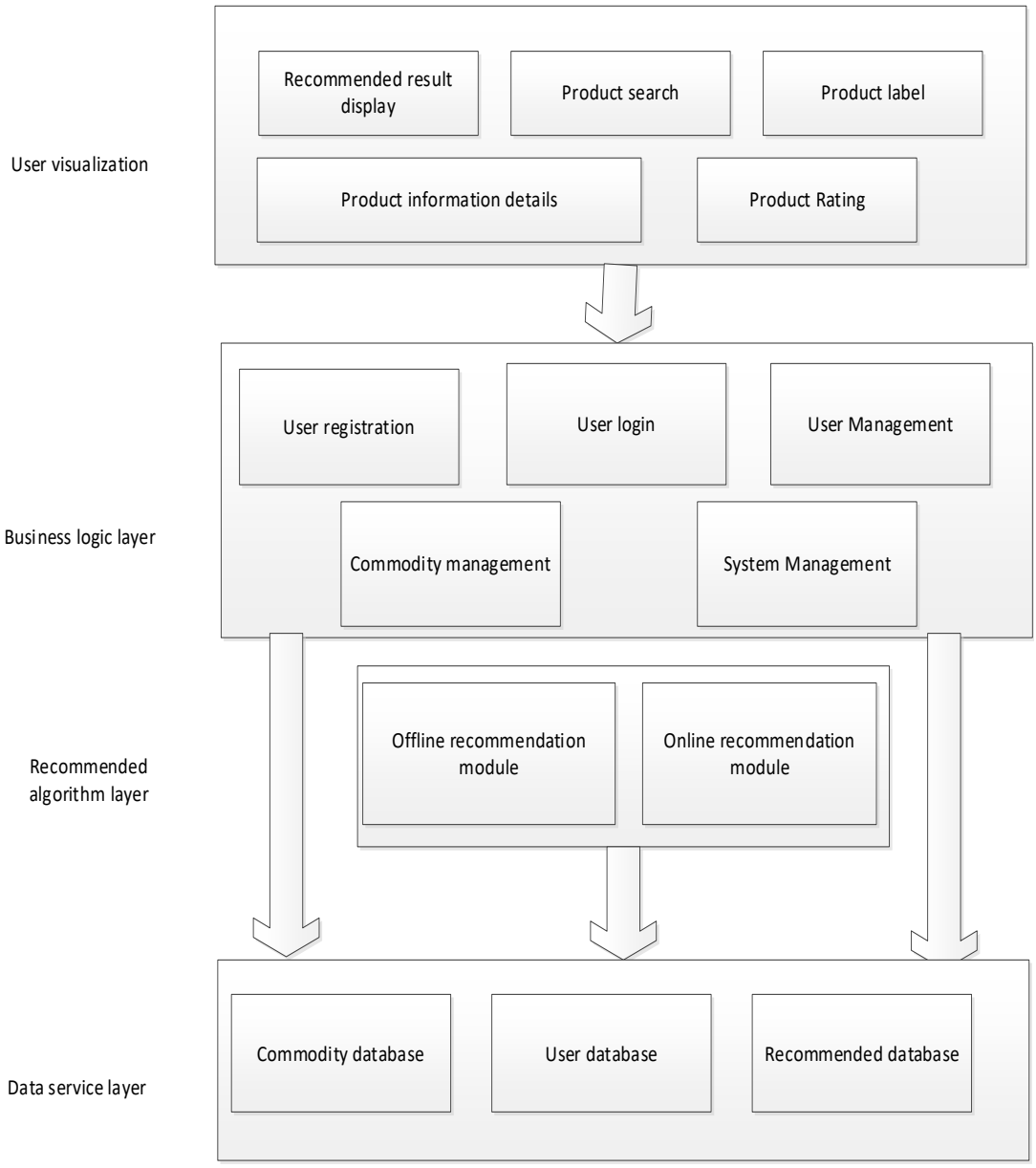

Figure 4. System architecture

3.2 Overall architecture design

According to the demand analysis of the e-commerce recommendation system based on personalized recommendation, the system architecture can be divided into the following levels: user visualization layer, business logic layer, recommendation algorithm layer, and data service layer. The system architecture is shown in [Figure 4].

User visualization layer: In this layer, users can directly interact with the system. Users can click any function module on the page to realize their desired function. Various operating interfaces are provided in the user visual interface. After the user operates, the request will be sent to the background server, and the data generated in the business system will be used as the main business database.

Business logic layer: The business logic layer connects the user and data layers and processes user operations. Business data mainly includes user registration and login information, user behavior data, product content information, label information, etc. When the user clicks on the page, the business logic layer will start the corresponding function module and return the corresponding result to the user.

Recommendation algorithm layer: The recommendation algorithm layer is where the recommendation algorithm is completed. The recommendation algorithm in this article will combine the Spark recommendation engine and use business data to provide users with recommendation services. The design of specific recommended modules will be elaborated in the next section.

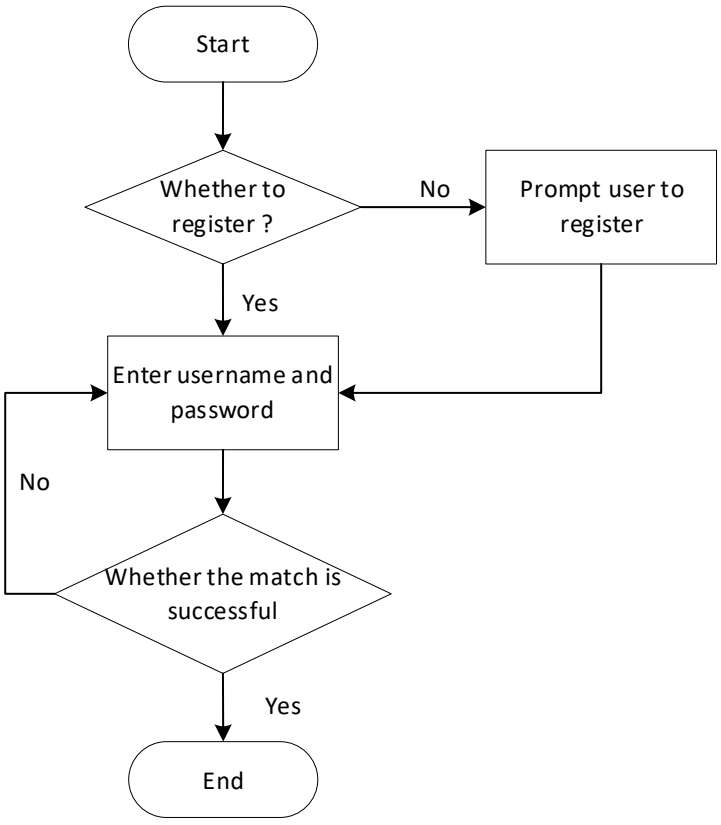

Figure 5. User login and registration module flowchart

Data service layer: mainly refers to the real-time operation of the business database. Both historical data and log data must be stored in the database. Different data types are stored in other databases according to their characteristics and combined to form a data layer as the bottom support for the entire system. The main operations of this layer include adding, modifying, querying, and deleting data in the database. This article uses a data loading service to import data files into the database directly.

3.3. System detailed design

A recommendation system's basic requirement is to provide users with an easy-to-operate and easy-to-understand operation interface. However, just satisfying this point is not enough. The system also needs to design a clear and clear operation process for users in advance. The design of the process must take the overall system architecture as the cornerstone, and it also needs to involve all the functions provided by the system to complete the recommendation. The detailed design flow chart of user registration and login modules is shown in Figure 5.

As you can see from the figure, when the user opens the page, the system needs to determine whether the user has already registered. If you are a new user, you must start the registration process before proceeding to the next step. Old users can directly enter the system. After the user enters the login information, the form will be submitted. When the submitted content matches the database, the page jumps to the recommendation system's homepage. If the match fails, the password input error will be prompted to repeat the previous step when a new user logs in. Registration is required. After the user clicks register, it will jump to the registration page. After registration, you can log in to the recommendation system.

After entering the system, users can see the display effect of the mixed recommendation designed in this article. The recommendation results obtained by different users are different. The recommended flowchart is shown in Figure 6.

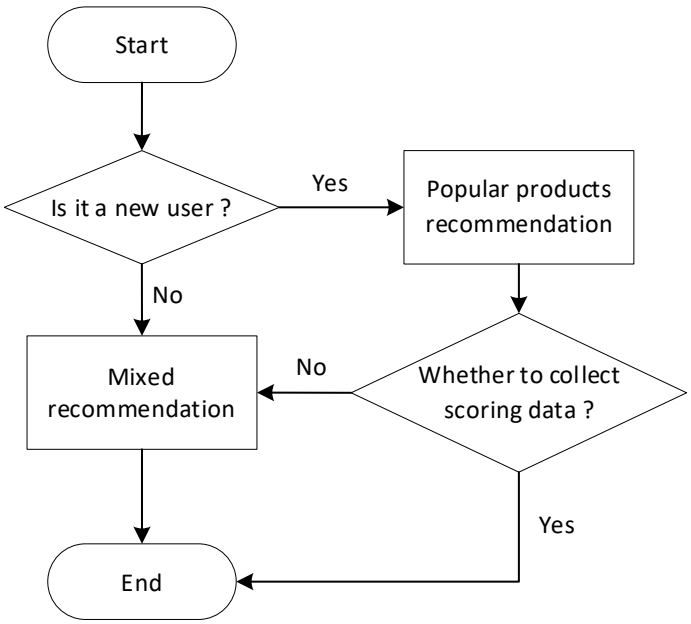

Figure 6. Recommended flowchart

For registered users, after the user logs in successfully, the system can make a comprehensive recommendation for them, provided that the user's basic information and preference information are stored in the system. The system cannot make personalized recommendations for new users because there is no data about new users. Currently, only popular product recommendations can be provided to the user. When users generate scoring data or tagging behavior, refresh the page to update personalized recommendations.

The design idea of the recommendation algorithm proposed in this paper is based on hybrid recommendation: the system is divided into two modules, offline and online. In terms of offline recommendation, this paper designs a demographic-based recommendation algorithm and a content-based recommendation algorithm to alleviate the cold start problem of the recommendation system. And designed a recommendation algorithm based on a hidden semantic model using ALS to replace the traditional recommendation algorithm based on collaborative filtering. In terms of real-time recommendation, this paper designs a model-based recommendation algorithm. This algorithm calculates based on the data produced by the implicit semantic model recommendation. It solves the problem of offline recommendations not being able to update the recommendation result immediately after the user's recent ratings. Finally, this article uses the Hadoop distributed big data processing platform to realize the processing and calculation of massive data to learn the integration of the above algorithms to form a hybrid recommendation system.

As the bottom layer of the system, the database stores all the data required for the operation of the recommendation algorithm, and the execution of any algorithm is inseparable from the support of a large amount of data. This article mainly introduces the design of database table structure, including user information table, product information table, statistical recommendation table, offline recommendation table, online recommendation table, etc. The data sheets are designed as follows:

(1) User Information Form

The user information table mainly stores the user's basic information. This recommended module records the user’s name, password, time, and other account information that is filled in when the user registers. The user information table structure is shown in Table 1.

Table 1. User Information Table

Field name	Field Type	Field description
User Id	Int	User ID
Username	String	Username
password	String	User Password
Timestamp	Lon0067	User creation time

A user corresponds to only one user ID, which will neither change nor disappear during the whole process of platform operation. The user's username and password are provided to the platform during registration. The prerequisite for logging in to the platform is that the username and password can be paired. Otherwise, the system will reject it. In this system, the user’s name is also unique.

(2) Product Information Sheet

The product information table stores the website products' names, categories, labels, URLs, etc. The design of the product information table is shown in Table 2.

Table 2. Product Information Table

Field name	Field Type	Field description	Field notes
productid	Int	Product ID	
Name	String	Product Name	
Categories	String	Product category	"|Each item is separated by "|"
Image Url	String	URL of the product image	
Tags	String	UGC label of the product	"|Each item is separated by "|"

As the primary key of the product information table, the product ID must be unique.

(3) User Rating Form

Table 3. User rating table

Field name	Field Type	Field notes
User Id	Int	User ID
Product Id	Int	Product ID
Score	Double	Product Rating
Timestamp	Long	Scoring time

(4) Product label list

Table 4. Product label list

Field name	Field Type	Field description
User Id	Int	User ID
Product Id	Int	Product ID
Tag	String	Product label
Timestamp	Long	Scoring time

The tables involved in the statistical recommendation module include a statistical table of the number of recent product ratings, a statistical table of product ratings, and a statistical table of the number of product ratings.

(5) Statistics on the number of recent product ratings

Table 5. Statistics of the number of recent product ratings

Field name	Field Type	Field description
Product Id	Int	Product ID
Count	Int	Number of product ratings
Year Month	Int	Scoring period

(6) Statistics on the number of product ratings

Table 6. Statistics of the number of product ratings

Field name	Field Type	Field description
Product Id	Int	Product ID
Count	Int	Average product rating

(7) Product average rating table

Table 7. Product average rating table

Field name	Field Type	Field description
Product Id	Int	Product ID
Avg	Double	Average product rating

The tables involved in the offline recommendation module include a commodity similarity matrix and a user commodity recommendation matrix. The commodity similarity table prepares for subsequent real-time recommendations.

(8) Commodity similarity matrix

Table 8. Commodity similarity matrix

Field name	Field Type	Field description
Product Id	Int	User ID
Recs	Array [(product Id: Int, score: Double)]	Similar products collection

(9) User product recommendation matrix

Table 9. User product recommendation matrix

Field name	Field Type	Field description
User Id	Int	User ID
Recs	Array [(product Id: Int, score: Double)]	Recommended product collection

(10) User real-time product recommendation matrix

Table 10. User real-time product recommendation matrix

Field name	Field Type	Field description
User Id	Int	User ID
Recs	Array [(product Id: Int, score: Double)]	Recommended product collection

The above data includes demand analysis, overall architecture design, and detailed system design for the e-commerce recommendation system. In the detailed design of the system, the user login and system recommendation processes were first designed. Then, the levels and functions of the system's recommended function modules were designed. Finally, the database's more commonly used and important data tables were explained in detail.

4. Conclusion

Based on the idea of hybrid recommendation, combined with multiple recommendation algorithms, this paper designs and implements an e-commerce recommendation system based on big data technology. The third chapter systematically taught the knowledge of big data technology and recommendation systems. It made an introduction to related theories and technologies, including Hadoop ecology, distributed file systems, and distributed computing engines, and an introduction to commonly used recommendation algorithms. Several mixed recommendation ideas are expounded to lay the foundation for the design and implementation of the recommendation system.

References

[1] D. Goldberg et al., "Using collaborative filtering to weave an information tapestry," Communications of the Acm, **(1992)**

[2] P. Pu, L. Chen, and P. Kumar, "Evaluating product search and recommender systems for E-commerce environments," Electronic Commerce Research, vol.8, no.1-2, pp.1-27, **(2008)**

[3] B. N. Miller, I. Albert, and S. K. Lam, "Movie lens unplugged: Experiences with an occasionally connected recommender system," International Conference on Intelligent User Interfaces, (AN), pp.263-266, **(2003)**

[4] P. Resnick and H. R. Varian, "Recommender systems. Commun ACM," Communications of the ACM, vol.40, no.3, pp.56-58, **(1997)**

[5] A. Carlos and Gomez-Uribe, "The Netflix recommender system: Algorithms, business value, and innovation," Acm Transactions on Management Information Systems, **(2016)**

[6] S. Ghemawat, H, Gobioff, and S. T. Leung, "The Google file system," ACM SIGOPS Operating Systems Review, ACM, **(2003)**, vol.37, no.5, pp29-43

[7] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," Sixth Symposium on Operating System Design and Implementation. USENIX Association, **(2004)**

[8] K. Shvachko, H. Kuang, and S. Radia, "The Hadoop distributed file system," IEEE Symposium on Mass Storage Systems and Technologies, IEEE, **(2010)**

[9] D. Xicheng. "Hadoop technology insider-in-depth analysis of YARN architecture design and implementation principles," Machinery Industry Press, **(2013)**

[10] S. Narayan, S. Bailey, and A. Daga, "Hadoop acceleration in an open flow-based cluster," (**2013)**

[11] N. Bansal, D. Upadhyay, and U. Mittal, "Concurrency control techniques in HDFS," Confluence the Next Generation Information Technology Summit, IEEE, **(2014)**

[12] X. Liang, "Recommendation system practice," People's Posts and Telecommunications Press, **(2012)**

[13] X. Junluan, S. Saisai, and Spark "Streaming: The newbie of large-scale streaming data processing," Programmer, **(2014)**, no.2, pp.44-47

[14] D. Siegal, J. Guo, and G. Agrawal, "Smart-mllib: A high-performance machine-learning library," IEEE International Conference on Cluster Computing (CLUSTER), IEEE, **(2016)**

[15] H. Li, A. Ghodsi, and S. Shenker, "Tachyon: Reliable, memory speed storage for cluster computing frameworks," **(2014)**

[16] R. He and J. Mcauley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class Collaborative filtering" **(2016)**

[17] R. Ronen, N. Koenigstein, and E. Ziklik, "Press the 7th ACM conference - Hong Kong, China (2013.10.12-2013.10.16) Proceedings of the 7th ACM conference on Recommender systems - RecSys 13 - Selecting content-based features for collaborative filtering recommenders," pp.407-410, **(2013)**

[18] C. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," Climate Research, vol.30, no.1, pp.79-82, **(2005)**

[19] D. M. Powers, “Evaluation: From precision, recall, and f-measure to ROC, informedness, markedness, and correlation,” **(2011)**