Research and Implementation of Precision Marketing System Based on Big Data Analysis

Ling Wang¹, Haochen Xu² and Yan Cao³

¹School of Cyber Security, Gansu Institute of Political Science and Law, Gansu, 730070, China
²Hefei University of Technology, Department of business, Xuancheng, 242000, China
³School of Electronic and Information Engineering Lanzhou Jiaotong University, Lanzhou, 730070, China

Abstract. For the data analysis in the mass data, the data processing ability of the recommendation system is very high. Various kinds of high performance computing frameworks are produced with the Spark, and the computing ability is very strong. The application of Spark to recommendation system will greatly improve the efficiency of operation. This paper discusses the processing technology of big data, and introduces some commonly used recommendation algorithms, and designs a recommendation system combined with big data.

1. Introduction
At present, mankind is in a new period of rapid development of the Internet, as people’s needs are no longer single, which began to change to diversification, personalized change. The use of the Internet will also produce massive data at the same time. How to make good use of the existing data, dig out the potential needs of users and push them to the right people in the right way has become a strategic problem affecting the development of enterprises. The idea of precision marketing is getting more and more public attention.

2. Big data technology

2.1. Spark
Spark is an open source computing framework similar to MapReduce's by the University of California, Berkeley, and AMP Labs. Because Spark is a memory-based computing framework, the intermediate computing results are placed in memory and do not need to read and write HDFSs, so the iteration efficiency is very high. Once introduced, the characteristics of high performance and ease of use attract a lot of relevant people. Spark has formed its own ecosystem based on Spark, which includes Spark SQL, MLlibSpark Streaming and GraphX, and becomes the top level project of Apache. Figure 1 is the mode of operation of the NT program under Spark.
As it is seen from the figure, Spark reads data from the file system first, and all the data is in RDD1. Then RDD1 gets a new RDD2 after the flatMap operation, which calculates the frequency of each word through the reduceByKey operation RDD3. Finally, RDD3 is written to the file system through the saveAsText method, and all operations are based on the saveAsText.

2.2. HDFS

HDFS is an open source software framework for distributed storage and big data processing using MapReduce programming model. It is designed and deployed on cheap hardware with high fault tolerance and high throughput. It is suitable for big data set project. Figure 2 shows the infrastructure diagram of HDFS.

3. Design of recommendation system based on Spark

The programming language of the recommendation system designed in this paper is Scala. Big data’s computing framework uses the Spark. Spark operation mode and running flow described in the previous article. The recommendation engine combined with the Spark framework is not to be restated here. The structural design is shown in figure 3. There are usually two kinds of tasks for offline
computing and online computing for each thread started by a node called executor-executor. For offline computing, data is first extracted from the data warehouse via SparkSQL, and then the offline data is then passed through the SparkS. QL writes back to the data warehouse. For online computing, it takes the offline data from the data warehouse first, and save the data in memory after the calculation. Then Executor returns the data to the next layer Spark Driver to continue processing.

Figure 3. Design Architecture with spark

4. Recommendation algorithm based on demography
The recommendation algorithm based on demographics is to calculate the similarity between users and users by using the attributes of users themselves. The algorithm used in this paper is to calculate the differences of mixed attributes between users. The recommendation strategy used in this paper is based on score prediction, so we first find a similar set of users \( U \{U1U2U3. Un\} \), and then judge the evaluation of the target item according to the previous behavior data of the target user.

4.1. Off-line calculation
The recommended off-line calculation based on demographics is to calculate the degree of difference between users and users. Before calculating, we should first analyze the classification of each attribute. For example, age is ordinal attribute and gender is symmetric binary attribute. Then we calculate it according to the formula described above. For the age, we need to re-code the age first. The edge coding is based on the ordinal attribute formula described above. Table 1 shows the original value and the coded value of the age.
Table 1. The original value of the age and the encoded value

| Original value | 1 | 18 | 25 | 35 | 45 | 50 | 56 |
|----------------|---|----|----|----|----|----|----|
| Coded value    | 0 | 0.167 | 0.333 | 0.5 | 0.667 | 0.833 | 1 |

Table 2. User information

| UserID | Location | Age |
|--------|----------|-----|
| 1      | nyc      | 0   |
| 2      | stockton | 18  |

Take Table 2 as an example; there are two users in the table. Each user has two attributes, one is the address, and the other one is the age according to the previous description of the common formula of calculation, we need to first calculate the age of distance, $d_{i,j} = \frac{0.167 - 1}{1 - 0} = 0.167$ Then according to the value of $d_{i,j}$, we calculate the distance between user 1 and user 2, $d(1, 2) = \frac{1 + 0.167}{2} = 0.583$.

Figure 4 shows the operation of the offline algorithm under Spark. First, SparkSQL reads the user information from the data warehouse. As the discussion earlier, every step of Spark is a RDD, so the user information read from the data warehouse is also a RDD. It can be called UserRDD. Then every user in UserRDD is combined with each other to provide a Cartesian method. You can directly perform a Descartes product operation on RDD and then generate a new RDDs, which can be called UserCartesian. Then we calculate the degree of difference between the two users through the map operation, and finally write the calculated results back to the data warehouse. For the data calculated offline, the data can also be written to the hdfs through the saveAsText operation. The data needed in offline data can be taken out and loaded into memory directly during online calculation.

Figure 4. The process of calculating the degree of dissimilarity between users
4.2. On-line computation

The recommendation strategy used in this paper is score prediction, before which two parameters need to be determined. The threshold of user dissimilarity and how many items of data need to be extracted from the recommended results. For example, the user dissimilarity can be 0.3, which means the similarity between the two users is 70. Here is a detailed description of the steps for online computing:

a) Firstly, the user information whose degree of difference between the target user and the target user is less than the specified threshold is taken out of the data warehouse. The RDD obtained by this step is processed and named userAntisim (user, antisim), indicating the degree of dissimilarity between each user and the target user.

b) Take out the data of all users who have scored too much on the target product from the user score table is for the purpose of obtaining rating RDD(user, rating), indicating each user’s rating on the target product.

c) The rating RDD and user Antisim are joined according to the user to obtain a new RDD (user, (rating, antisim), in which rating represents the user's score on the target product, and antisim represents the degree of dissimilarity between the user and the target user.

d) The RDD obtained in the third step is subjected to a map operation to obtain a new RDD (antisim, user.rating)). Then the records in the RDD are sorted according to antisim, and the specified number is loaded into the memory.

e) The scoring formula for evaluating the target commodity is as follows:

\[ \text{Score} = \sum_{i \in R} \frac{(1-\text{antisimi})}{\text{sim}_\text{all}} \times \text{ratings}_i \]

Sim_all represents the sum of user similarities in RDD in the fourth step. Since the previous calculation is dissimilarity, the sum of similarity is calculated as \( \Sigma (1-\text{similarity}) \), and R represents all the records in the fourth step of RDD. The above formula can be used to calculate the target user’s possible score for the target product. After the result is calculated, the result is loaded into memory. The algorithm flow is shown in Figure 5.

5. Summary

The purpose of this paper is to realize the precision marketing system based on big data. Aiming at how to realize the precision marketing, this paper designs and implements the recommendation system based on Spark, and introduces the related algorithm and the recommendation strategy of the recommendation system. Then the recommendation system is designed.
References

[1] Zaharia M, Chowdhury M, Franklin M J, et al. Spark: cluster computing with working sets [C]// Usenix Conference on Hot Topics in Cloud Computing. USENIX Association, 2014: 10-10.

[2] Gopalswamy N, Hanaoka Y, Hudson H S. Structure and dynamics of the corona surrounding an eruptive prominence [J]. Advances in Space Research, 2016, 25 (9): 1851-1854.

[3] Zaharia M, Chowdhury M, Das T, et al. Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing [C]//Usenix Conference on Networked Systems Design and Implementation. USENIX Association, 2012: 2-2.

[4] Shvachko K, Kuang H, Radia S, et al. The Hadoop Distributed File System [C]//IEEE, Symposium on MASS Storage Systems and Technologies. IEEE Computer Society, 2010: 1-10.

[5] Lops P, Gemmis M D, Semeraro G. Content-based Recommender Systems: State of the Art and Trends [M]//Recommender Systems Handbook. Springer US, 2011: 73-105.