Product Customer Demand Mining and Its Functional Attribute Configuration Driven by Big Data

Dianting Liu[1,2(✉)], Xia Huang[2], and Kangzheng Huang[1]

[1] College of Mechanical and Control Engineering, Guilin University of Technology, Guilin 541004, Guangxi, China
365565379@qq.com
[2] College of Information Science and Engineering, Guilin University of Technology, Guilin 541004, Guangxi, China

Abstract. The maturity of big data analysis theory and its tools improve the efficiency and reduce the cost of massive data mining. This paper discusses the method of product customer demand mining based on big data, and further studies the configuration of product function attributes. Firstly, the Hadoop platform was used to perform product attribute data participle and feature word extraction based on Apriori algorithm was used to mine product customer demand information. And then the MapReduce model on the big data platform was applied into efficient parallel data processing, obtaining product attributes with research value, and their weights and attribute levels. After that, the cloud model and the MNL model were employed to construct the product function attribute configuration model, and the improved artificial bee colony algorithm was used to solve the model. The optimal solution of the product function attribute configuration model was got. Finally, an example was given to illustrate the feasibility of the proposed method in this paper.

Keywords: Big data · Customer demand · Product function attribute configuration · Apriori · MNL model · Artificial bee colony algorithm

1 Introduction

At present, in traditional methods such as market survey questionnaires, household interviews, observation method, etc., consumer demand for products is obtained manually, which is not only time-consuming and laborious, but it is difficult for users to express their actual views on the product also objectively and rationally. It causes that the workload of text processing is large, which affects the validity and real-time of information in the process of requirement analysis.

With the popularization and in-depth application of the Internet, more and more users like to express their opinions on products on the network; It has broken through the time and space restrictions, and is generally non-interfering and spontaneous, which can be regarded as true and reliable. These product demand information can be automatically and efficiently collected and processed by computer systems, and then the actual needs of users can be analyzed timely and effectively to assist enterprises in product innovation.

J. Zeng et al. (Eds.): ICPCSEE 2020, CCIS 1257, pp. 145–165, 2020.
https://doi.org/10.1007/978-981-15-7981-3_11

In the e-commerce website and WeChat, blog, QQ social network space, people's comments on products have the features of big data [1] in Volume, Velocity, Variety, Value, Veracity. This type of data set is rich and huge, and cannot be collected, managed, and analyzed using traditional data processing methods. At present, big data platforms and analysis tools such as Hadoop platform and MapReduce are commonly used to perform parallel processing calculations for solving the hardware bottleneck and software performance constraints in mining algorithms [2–4].

The product function attribute configuration [5–7] can be considered as a decision process to determine the attribute level value of a new product under the premise of knowing customer preferences, competitive products and market information. Before developing a product, an enterprise can investigate customer needs and preferences, establish an optimization model to obtain the optimal product attribute level value and attribute combination, and at the same time combine product engineering performance and market performance to optimize the design of the product.

This article discusses the use of big data platforms and analysis tools to collect customers' comments on products on the network, and then to mine customer needs for products; On this basis, the theory and method of combining cloud models and MNL models to configure product function attributes is researched.

2 Product Customer Demand Mining Based on Big Data

Modern products are becoming more and more complex and with numerous product attributes. When an enterprise's product is to be positioned, the product attributes and attribute levels must be firstly determined. If the data in this step is inaccurate, it will lead to the inaccuracy of subsequent data. Because the data collected by traditional methods such as sampling surveys and questionnaires on potential customers or experts is too random, and with incomplete data types and high blindness, it cannot be used as the decisive data to determine product attributes. Therefore, big data collection platform and analysis tools should be used to collect data on a large scale, so that the collected data is sufficiently comprehensive and complete to play a decision-making role.

2.1 Data Collection

The Jingdong online shopping platform is selected as the object for crawling data in this paper. The process of crawling the data has two steps: the first step is to crawl the product comments, the product comments data segmentation is done under the Hadoop platform, and then the product attributes with research value is to be mined by Apriori algorithm; In the second step, the product attributes excavated in the first step are used to calculate the relevant product attribute level and sales, so that the product attribute level with research value will be determined.

(1) Crawling data. The required data is by Scrapy. Scrapy is a fast, high-level web crawling Web Spider framework which is developed using Python language, and used to crawl structured data from web site pages. The steps to crawl product comments are as follows: First step analyzed the page and defined the fields that

need to crawl; The second step is to analyze the interface url and parse the crawling content field through xpath and json; In the third step, write the storage method in the pipelines.py file; The fourth step is to start crawling; Finally stored in ElasticSearch database.

(2) Data preprocessing. While scraping the product information, the data is simply denoised, such as removing duplicate data and deleting blank lines. And then the data must be further processed, such as removing emoji, special characters, stop words in Chinese and English and other junk data.

2.2 Product Attribute Mining

The first part of the data is obtained through the crawling program, which is the comments data of the product on the Jingdong Mall, and then the data segmentation and Apriori feature extraction are performed.

Data Segmentation. The Chinese word segmentation tool used in this article is jieba, which is currently the most used Chinese word segmentation tool in China. The captured data stored in MySQL is imported into HDFS using Sqoop and the jieba package is imported on the Hadoop project, and the word segmentation is calculated on the mapreduce program. The process is shown in Fig. 1:

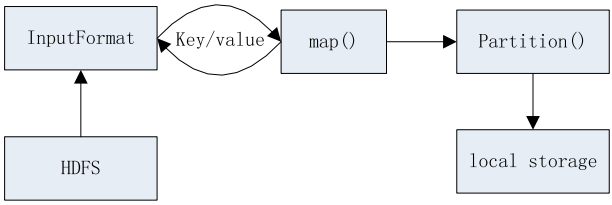

Fig. 1. Map take execution process

Only simple word segmentation is performed here, so there is no need for multiple mappers and no reducer. There is no control on the Inputformat here. If there are many files, in order to ensure the highest computational efficiency of mapreduce, the Inputformat must be controlled to limit the size of the slices. To set 'Key' for the offset of each line of text and 'value' for the content of text, and the main function is realized by the function of map().

Feature Words Extraction Based on Hadoop and Apriori Algorithm. Introduction of Apriori algorithm [8–12] to extract feature words.

Use the comments data and feature set A to construct 0-1 matrix M:

$$M = \begin{Bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{Bmatrix} \quad (1)$$

Where a_{ij} is equal to 0 or 1, this comment has this attribute feature represented by 1, if not, it is represented by 0, $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$; N attribute feature sets are represented by $I = \{I_1, I_2, I_3, \ldots, I_N\}$. According to formula (2), calculate the probability of I_j appearing in the transaction database is $p(I_j)$; calculate the weight $w(I_j)$ of I_j by formula (3).

$$p(I_j) = k/m \tag{2}$$

$$w(I_j) = 1/p(I_j) \tag{3}$$

Where the frequency of I_j appearing in the transaction set is represented by k, which is the number of 1 in column j of matrix M, and the total number of comments in matrix M is represented by m.

In formula (4), the *l-th* comment in the dataset is represented by R_l. Take the average weight of all attribute features in this comment and record it as $wr(R_l)$, which is the $w(I_j)$ sum of all aij = 1 in line i is averaged. The weight calculation method of Rl is calculated according to formula (4).

$$wr(R_l) = \sum_{j=1}^{I_j \in R_l} w(I_j)/|R_l| \tag{4}$$

In the above formula, the number of commented R_l containing attribute feature items is represented by $|R_l|$.

The weight support of attribute is denoted as *wsupport*, the weight represents the proportion of transaction weights containing attribute features to all transaction weights, and then set the lowest threshold according to the weight support of attribute features to form the optimal feature set. Calculated according to formula (5).

$$w\sup port(S) = \sum_{l=1}^{S \subseteq R_l} wr(R_l) / \sum_{l=1}^{m} wr(R_l) \tag{5}$$

In the above formula, any attribute characteristic item in the transaction database is denoted by S.

The Apriori algorithm steps are as follows:

Step 1: Scan the comment database, construct a Boolean matrix of attribute features, and calculate $p(I_j)$ according to the comment transaction matrix, which is the probability of each attribute feature appearing in the transaction database, and then calculate the weight $w(I_j)$ of each attribute feature and $wr(R_l)$ of each comment transaction.

Step 2: Calculate the weight support $wsupport(S)$ of the attribute items to obtain the candidate 1-item set. Generate frequent 1-item sets according to the weight support threshold of the minimum item, and finally get the result.

Implementation of Apriori algorithm to extract feature words under MapReduce.

When the Apriori algorithm is being calculated, a large number of candidate sets will be generated, and the database needs to be scanned repeatedly. Because there is a lot of data, MapReduce built on Hadoop for parallel calculation can improve the performance and efficiency of the algorithm. Figure 2 is the MapReduce model implementation process of the Apriori algorithm under the Hadoop framework.

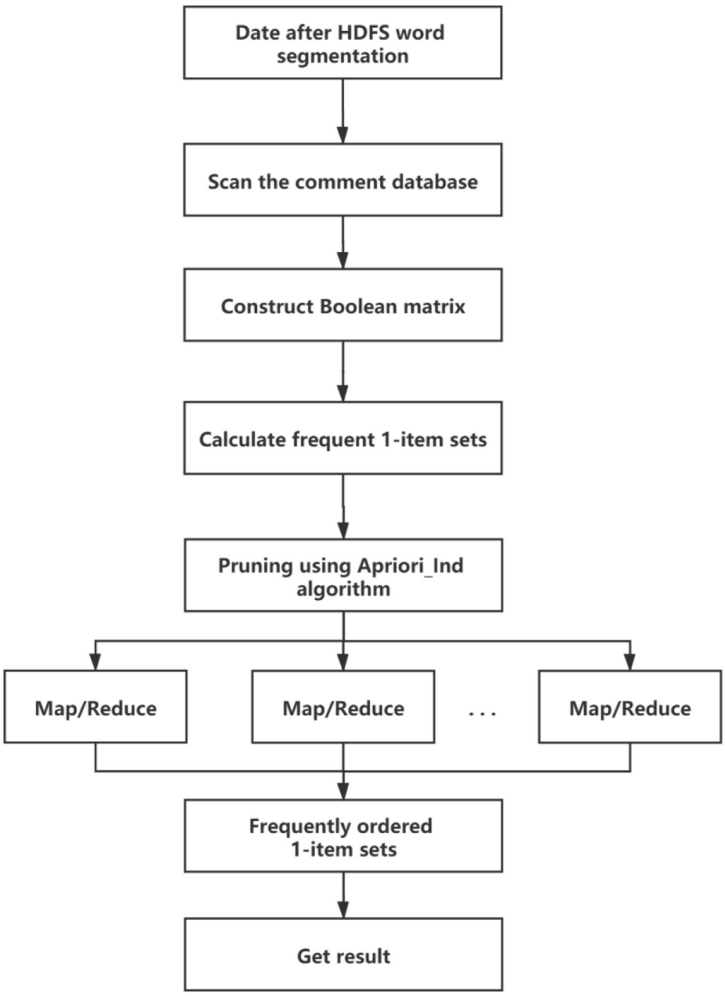

Fig. 2. Apriori algorithm under Hadoop

The MapReduce function is used to perform distributed processing on data, and a frequent 1-item set that meets the weight support threshold of the minimum item is retrieved. The traditional Apriori algorithm has the disadvantage of generating a large number of redundant candidate item sets and frequently scanned data. Therefore, in this

paper, combined with the Apriori_Ind algorithm in the Reference [12], which improves the representation of each node's block strategy and frequent item set. The feature of different items of each node realizes that each node will generate a unique candidate set and pruning based on frequent item sets, which effectively reduces the time of generating, pruning and support statistics of each node's candidate set. At the same time, the representation of frequent item sets is improved to <previous, post>, which can reduce the size of data transmission between Hadoop clusters and accelerate the generation of pruning and candidate sets.

The first-level candidate set is generated by calculating each set during the process of reading the dataset after HDFS word segmentation, and then the first-level candidate set is pruned by the Apriori_Ind algorithm. For this candidate set, the Map function under the Hadoop framework is used to divide the entire original dataset into several subsets (first-level candidate set), and then it is distributed in parallel to the Reduce function for reduction, filter the frequent 1-item sets by defining the weight support threshold of the minimum item. Finally, sorted out the attribute feature words, which are product attributes with research value.

2.3 Product Attribute Level Mining

On the basis of the above product attribute determination, the current product sales and product attribute level data are collected and analyzed, so as to determine the product attribute level with more research value, and lay the foundation for the calculation of product attribute preference in the following. The main steps are as follows: First to crawl the product attributes and sales of Jingdong Mall by crawl program, and then preprocess the data, after that get the product attribute level of more research value by using liner statistical analysis.

To select the most valuable research attribute level is equivalent to finding the sum of sales of all products containing this attribute level. Higher sales indicate that this attribute level is more popular, and the more valuable it is for research. Calculated as follows:

$$sum(u_{ij}) = \sum_{n=1}^{6000} salevolume(u_{ij})_n * x_{ij} \tag{6}$$

In the above formula, the *j-th* attribute level of the *i-th* attribute of the product is represented by u_{ij}; x_{ij} represents the value of 0 or 1, the *j-th* attribute level with *i-th* attribute is 1, and 0 if not included; The *n-th* product contains the sales of the *j-th* attribute level of the i-th attribute is represented by $salevolume(u_{ij})_n$, and the total sales of products including the *j-th* attribute level of *i-th* attribute is represented by $sum(u_{ij})$.

3 Product Function Attribute Configuration

In general, the combination of product attribute levels is effective, and the product outline can be represented by a combination of attribute levels. For example, the product has I product attributes, and each attribute i has J attribute levels. The goal of

the new product's functional attribute configuration is to find the optimal product profile of the new product on the basis of maximizing the company's total profit.

3.1 Customer Preferences

When describing customer preferences, linearly superimpose the horizontal utility value of the product's own attributes is the most common method. Product attribute weights and product price attributes are considered in this article. In the product market, product price attributes have decisive relationship with whether customers choose to purchase this product. Some customers will only purchase the price that they can accept. As product prices rise, customer preferences will decrease and sales will decline. The price factor p is introduced when calculating U_k for customer preferences; customers pay different attention to different attributes of products, so when calculating the U_k, the product attribute weight q_i will be introduced.

$$U_k = \sum_{i=1}^{I} \sum_{j=1}^{J} u_{ij} q_i x_{ij} - \gamma p \tag{7}$$

Where the total utility value of product is represented by U_k, the partial utility value of the *j-th* attribute level selected in the *i-th* attribute of the product is represented by u_{ij}, which is calculated by combing the cloud model and the discrete selection model. $x_{ij}(i = 1, 2, .., I; j = 1, 2, ..., J)$ is a variable of 0 and 1, when the *j-th* attribute level of the *i-th* attribute is selected, $x_{ij} = 1$, otherwise $x_{ij} = 0$. The customer's attention weight for the *i-th* attribute of the product is q_i. γ is the weight of price influence, where $\sum_{i=1}^{I} q_i + \gamma = 1$; The sales price of product k is p.

3.2 Product Attribute Utility Value

(1) Language evaluation of product attributes and conversion of cloud model. Suppose customers use n-level scale to evaluate product attributes, where n = 2t + 1, $t \in N$; The evaluation value is recorded as H: $H = \{h_v | v = -t, ..., 0, ..., t, t \in N\}$ in the natural language set. For example, according to the 7-level evaluation scale, the language evaluation set for product attributes is $H = \{h_{-3} = worst, h_{-2} = worse, h_{-1} = bad, h_0 = general, h_1 = good, h_2 = better, h_3 = best\}$. As the show in Table 1, each attribute level of the product is represented by the first line, and a customer's natural language evaluation of each attribute level is represented by the second line. The natural language evaluation of the *j-th* attribute level selected in the *i-th* attribute is $(h_v)ij$.

Table 1. Qualitative evaluation of product attributes

The attribute level of the selected product k	k_{1j}	k_{2j}	...	k_{ij}	...	k_{Ij}
Customer evaluation set h_v	$(h_v)_{1j}$	$(h_v)_{2j}$	…	$(h_v)_{ij}$	…	$(h_v)_{Ij}$

Definition [13] given a language evaluation set $H = \{h_v | v = -t, ..., 0, ..., t, t \in N\}$, there is a function that coverts h_v to the corresponding value θ_v, where $\theta_v \in [0, 1]$:

$$\theta_v = \begin{cases} \frac{a^t - a^{-v}}{2a^t - 2}, -t \le v \le 0 \\ \frac{a^t + a^{-v} - 2}{2a^t - 2}, 0 \le v \le t \end{cases} \tag{8}$$

The value of a is in the interval [1.36, 1.4], which can be obtained from the experiment [14]. In this paper, $a \approx 1.37$.

The conversion process from qualitative to quantitative is shown in the following Algorithm 1:

Algorithm 1: Standard evaluation cloud generator.

Input: The attribute evaluation scale is n-level and the effective domain of attribute comment value is $[D_{min}, D_{max}]$.

Output: Standard evaluation cloud $A_v = Y(Ex_v, En_v, He_v)$, where $v = 1, 2, ..., n$.

The algorithm steps are as follows:

Step1: Calculated θ_v by formula (8)

Step2: Calculated the expected value Ex_v based on the upper and lower limits of the domain $[D_{min}, D_{max}]$.

$$Ex_v = D_{min} + \theta_v (D_{max} - D_{min}) \tag{9}$$

Step3: Calculated the entropy En_v according to Step2.

$$En_v = \begin{cases} \frac{(1-\theta_v)(D_{max} - D_{min})}{3}, -t \le v \le 0 \\ \frac{\theta_v (D_{max} - D_{min})}{3}, 0 \le v \le t \end{cases} \tag{10}$$

$$En_{-v} = En_v = \begin{cases} \frac{(\theta_{|v|-1} + \theta_{|v|} + \theta_{|v|+1})(D_{max} - D_{min})}{9}, 0 < |v| \le t - 1 \\ \frac{(\theta_{|v|-1} + \theta_{|v|})(D_{max} - D_{min})}{6}, |v| = t \\ \frac{(\theta_v + 2\theta_{v+1})(D_{max} - D_{min})}{9}, v = 0 \end{cases} \tag{11}$$

Step4: Calculated the super-entropy He_v according to Step3.

$$He_{-v} = He_v = \frac{En'^{+} - En}{3} \tag{12}$$

$$En'^{+} = \max_k \left\{ En'_k \right\} \tag{13}$$

After Algorithm 1, the qualitative natural language evaluation can be converted into a quantitative value, which is the characteristic $Y(Ex_v, En_v, He_v)$ of cloud number that used standard evaluation cloud to represent each language evaluation interval. As shown in Table 2, the level value k_{ij} of each attribute of the product k is represented by the first line; The customer's natural language evaluation $(h_v)_{ij}$ of each attribute level is represented by the second line; And the converted cloud $Y((Ex_v)_{ij}, (En_v)_{ij}, (He_v)_{ij})$ of customer evaluation is represented by the last line.

Table 2. Product attribute evaluation cloud

Product k attribute level	k_{1j}	k_{2j}	…	k_{Ij}
Customer natural language evaluation h_v	$(h_v)_{1j}$	$(h_v)_{2j}$	…	$(h_v)_{Ij}$
Converted cloud	$Y((Ex_v)_{1j}, (En_v)_{1j}, (He_v)_{1j})$	$Y((Ex_v)_{2j}, (En_v)_{2j}, (He_v)_{2j})$	…	$Y((Ex_v)_{Ij}, (En_v)_{Ij}, (He_v)_{Ij})$

(2) Product attribute utility. The attribute level utility value u_{ij} is calculated based to the customer's evaluation cloud $Y((Ex_v)_{ij}, (En_v)_{ij}, (He_v)_{ij})$ for each attribute level of the product. Generally speaking, products have multiple customers, and customers have different characteristics in real life, such as gender, age, position, etc., so they need to be classified. However, the proportion of each type of customer is different, and the importance of product evaluation is also different. A weighting factor β is introduced to the calculation of attribute utility value u_{ij}. The value of β can be adjusted according to the proportion of the customer's characteristic attributes, and then combined the evaluation cloud's Ex_v calculation to enhance its rationality and credibility. Suppose that product k has L types of customers, and each type of customer has M individuals. The calculation formula of u_{ij} is as follows:

$$u_{ij} = \frac{e^{R_{ij}}}{e^{R_{ij}} + \sum_{j=1}^{J} e^{R_{ij}}} \tag{14}$$

$$R_{ij} = \frac{\sum_{m=1}^{M} \beta_l (Ex_v)_{ij}^{lm}}{\sum_{l=1}^{L} M_1} \tag{15}$$

Where the expectation of the type l customer m for the evaluation of the *j-th* attribute level of the *i-th* attribute of the product is represented by $(Ex_v)_{ij}^{lm}$, which can be obtained by formula (9). The weight of type l customer evaluation is β_l, where $\sum_{l=1}^{L} \beta_l = 1$; The number of customers of type l is denoted by M_l; The total expectation of the *j-th* attribute level of the *i-th* attribute of the product is represented by R_{ij}.

3.3 Product Selected Probability

According to the MNL model, the probability C_k of a customer choosing a new product k among many competing products is calculated by formula (16):

$$C_k = \frac{e^{\chi U_k}}{e^{\chi U_k} + \sum_{r=1}^{r-1} e^{\chi U_r}} \tag{16}$$

In the formula, the overall utility value of product k is represented by U_k; The utility value of the *r-th* competitive product is U_r; The proportionality parameter is represented by χ, if χ has a tendency to approach infinity, then this model approximates

deterministic choice, which means that the customer is absolutely rational when making a choice, and the final product performance preference is the best one. If χ is closed to zero, so this model approximates random selection, and the selection probability tends to be randomly and uniformly distributed. In this paper, the MNL proportionality parameter χ is calibrated to 0.5.

3.4 Product Function Attribute Configuration Model

Based on the customer purchase selection rules, the expected number of customers who purchase a new product k is Q_k, $Q_k = QC_k$. The product profitability index EP:

$$\begin{aligned} EP &= Q_k(p - W) = QC_k(p - W) = QC_k \\ &= \frac{e^{\chi U_k}}{e^{\chi U_k} + \sum_{r=1}^{r-1} e^{\chi U_r}} (p - \sum\nolimits_{i=1}^{I} \sum\nolimits_{j=1}^{J} f_{ij} x_{ij}) \end{aligned} \tag{17}$$

In the above formula, the meaning of C_k and U_k have been discussed in the foregoing, f_{ij} is unit cost of the *j-th* attribute level of *i-th* attribute, the number of potential customers is Q, the product cost is W, the product price is p.

3.5 Selection of Algorithm for Solving the Model

The product positioning optimization model is a discrete nonlinear model. In this model, x_{ij} is discrete variable, the price p is continuous variable. The determination of the product profile is a combination of product multiple attributes and attribute levels. Because of the variety and complexity of product attributes and attribute levels, it can be classified as NP (Non-Deterministic Polynomial) in combination optimization. Compared with GA (Genetic Algorithm), PSO (Particle Swarm Optimization Algorithm) and DE (Differential Evolution Algorithm), ABC (Artificial Bee Colony Algorithm) has the advantages of less parameter setting, fast convergence speed, and high convergence accuracy. In this paper, the improved ABC algorithm is used to solve the above product positioning design optimization model.

In the ABC algorithm, the initial solution is randomly generated twice. Once is when the population is initialized; And the other is when a food source is not updated within the maximum limit times, then the initial solution is generated by the detective bee. Therefore, the initialization will be improved separately in this article. It also proposes improved methods for search strategies. The specific improvement methods are as follows:

(1) Improvement of population initialization. Because the initial solution is randomly generated, there may be excessive concentration of individuals in the random solution, and reducing the global search performance of the algorithm and relying more on the detection bee. In this paper, the reverse learning strategy is combined to improve the initialization, the improvement ideals are as follows:

Randomly generate N/2 food sources within the search space (set the population to $N = r * M$), and set the space solution to $g_{i,j} \in (g_{i,min}, g_{i,max})$, and calculation formula of reverse solution is as follows:

$$g'_{i,j} = (g_{i,\min} + g_{i,\max} - g_{i,j}) \tag{18}$$

Calculated the fitness of all food sources, including reverse food sources, and the best r food source are selected and used as the center points of the subpopulations in thinking evolution. The distribution is based on r center points, and each generates M random food sources that obey the normal distribution.

(2) Improvement of detection Bee initialization. The traditional ABC algorithm is too random for the position of the food source generated by the initial detection of the detection bee, which leads to slow convergence and easy to fall into the local optimal. However, the Gaussian distribution has strong perturbation. In the Gaussian distribution, the application of random perturbation terms can solve the problem of individuals falling into local optimality, and can improve the accuracy of the solution. The improved formula used is shown in (19).

$$g_{i,j} = g_{best,j} + g_{best,i} \cdot N(0,1) \tag{19}$$

(3) Improvement of search strategy. In the original artificial bee colony algorithm, when detection bees and following bees to search, the search strategy adopted is better ability for global search, but it ignores the ability for local search. Therefore, by referring to the PSO (Particle Swarm Optimization Algorithm) and introducing the current optimal and suboptimal solutions, and a new search method is proposed:

$$v_{i,j} = g_{best,j} + \varphi_{i,j}(g_{w,j} - g_{k,j}) + \delta_{i,j}(g_{\sec ondbest,j} - g_{i,j}) \tag{20}$$

In the formula, the candidate food sources are represented by $v_{i,j}$, $g_{w,j}$ and $g_{k,j}$ are randomly generated unequal known solutions, $\varphi_{i,j}$ and $\delta_{i,j}$ are random value on [−1, 1], the current optimal food source position is $g_{best,j}$, the second best food source position is $g_{secondbest,j}$. After introducing the current optimal and suboptimal solutions, the local search ability of the algorithm is improved to a certain extent, and the convergence speed is accelerated.

4 Examples and Analysis

4.1 Software and Hardware Environment

The experimental cluster is composed of 5 PCs, one of which is a computer with a higher CPU frequency, configured as a Master and used as a Slave at the same time, and the remaining 4 computers are isomorphic only as Slave. The configuration is shown in Table 3, using a Windows64 system, Using Hadoop 2.6.0-cdh 5.7.0 version, jdk is using version 1.7.0.

Table 3. Experimental equipment configuration

Node type	Node name	CPU	RAM
Master/Slave	Namenode	i5-8250U 3.4 GHz	8G
Slave	Datanode	I5-3210 M 2.50 GHz	4G

4.2 Example Application

Suppose an enterprise performs the configuration design of the functional attributes of a certain model of smartphone. After investigation and statistics, this model of smartphone has more than 20 attributes and more than 60 attribute levels. The product attributes and attribute levels with research value are analyzed through big data mining, and then the customer preferences of these attributes are obtained through questionnaires, and then the product function attribute configuration model is solved using an improved ABC (Artificial Bee Colony) algorithm to obtain the optimal product function property configuration.

Determination of Phone Attributes. By crawling the mobile phone comments of Jingdong Mall, and then performing word segmentation and Apriori algorithm feature extraction on the Hadoop platform. The weight support of each attribute feature item is calculated by formula (5), and then top 9 phone attribute features are extracted, which are the mobile phone attributes selected in this paper. As shown in Table 4.

Table 4. Mobile phone attribute feature extraction results

Attributes	wsupport
CPU	0.3462
RAM	0.3211
Price	0.2886
Mobile phone pixel	0.2412
Fingerprint unlock	0.2133
Screen size	0.1739
Battery	0.1621
ROM	0.1380
Resolution	0.1258

After the mobile phone attribute feature item is selected, the weight of each attribute feature item is calculated according to formula (3). As shown in Table 5.

Table 5. Weights of mobile phone attributes

Attributes	Weights
CPU	0.1713
RAM	0.1640
Price	0.1370
Mobile phone pixel	0.1127
Fingerprint unlock	0.1055
Screen size	0.0981
Battery	0.0909
ROM	0.0662
Resolution	0.0542

Through the feature item extraction of the Apriori algorithm, the top 9 phone attribute features are extracted, indicating that these 9 attributes are also the mobile phone attributes that users are most concerned about. Among them, the other 8 attributes except the price belong to the hardware attributes of the phone, which are included in the next section of the attribute level research. The weight of the influence of price is 0.137 from Table 5, that is, $\gamma = 0.137$ in the previous chapter.

Determination of Mobile Phone Attribute Level. On the basis of the determination of the above product attributes, the current mobile phone product sales and product attribute level data are collected and analyzed, and then to determine the product attribute level that is more valuable for research.

The relevant attributes and sales of mobile phone products are used by the spider program to crawl in Jingdong Mall, and then the 23 product attribute levels with the most research value are finally determined according to formula (6). The specific product attributes and attribute levels are shown in Table 6.

Phone attribute Preferences. (1) Questionnaire of mobile phone attribute preferences. The content of this questionnaire is to set the preference of 23 attribute levels of the mobile phone to be studied above. The answers included seven levels: best, better, good, general, bad, worse and worst. This questionnaire is for students at school, and 200 students are randomly selected as the object of the survey. According to the investigation, the number of valid questionnaires among the 200 statistical results obtained is 186. Among the valid questionnaires, there are 100 men and 86 women. After analyzing 186 valid questionnaires, the 23 attribute level preference values of the mobile phone were obtained.

Table 6. Smartphone product attributes and attribute levels

Attribute	Attribute level
Screen resolution	1920 * 1080
Screen resolution	2340 * 1080
Screen resolution	1440 * 720
CPU core + RAM (running memory)	Eight core+6g
CPU core + RAM (running memory)	Eight core+4g
CPU core + RAM (running memory)	Eight core+3g
ROM (body memory)	128g
ROM (body memory)	64g
ROM (body memory)	32g
Screen size	5–5.5 in.
Screen size	5.5–6 in.
Screen size	6–6.5 in.
Front camera pixels	5–10 million
Front camera pixels	10–16 million
Front camera pixels	20–25 million
Rear camera pixels	800 or less
Rear camera pixels	1200–1900
Rear camera pixels	2000–2400
Battery capacity	3000–3500 mAh
Battery capacity	3500–4000 mAh
Battery capacity	4000–5000 mAh
Fingerprint recognition	Support
Fingerprint recognition	Unsupport

Table 7. Survey results show

Research objects	CPU			Body memory			…
	Eight core+3g	Eight core+4g	Eight core+6g	128g	64g	32g	…
1	h_{-2}	h_1	h_2	h_{-1}	h_3	h_{-2}	…
2	h_{-3}	h_0	h_2	h_3	h_2	h_0	…
3	h_1	h_2	h_{-1}	h_0	h_1	h_{-1}	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
186	h_0	h_3	h_{-1}	h_{-2}	h_3	h_{-3}	…

(2) Analysis of questionnaire results. Due to too much data, part of the survey data is shown in Table 7. The language evaluation set of product attributes is $H = \{h\text{-}3 = worst,\ h\text{-}2 = worse,\ h\text{-}1 = bad,\ h0 = general,\ h1 = good,\ h2 = better,\ h3 = best\}$, after sorting, Table 7 is obtained:

① According to formula (8), θ_i can be obtained, and the qualitative evaluation language is converted into a cloud model, when $t = 3$, then: $\theta_{-3} = 0$, $\theta_{-2} = 0.221$, $\theta_{-1} = 0.382$, $\theta_0 = 0.5$, $\theta_1 = 0.618$, $\theta_2 = 0.779$, $\theta_3 = 1$.

② The three digital features are calculated using formulas (9)–(13), assuming that the domain: $[D_{min}, D_{max}] = [2, 8]$, then: $Ex_{-3} = 2$, $Ex_{-2} = 3.326$, $Ex_{-1} = 4.292$, $Ex_0 = 5$, $Ex_1 = 5.708$, $Ex_2 = 6.674$, $Ex_3 = 8$. $En_{-3} = 1.779$, $En_{-2} = En_2 = 1.598$, $En_1 = En_{-1} = 1.265$, $En_0 = 1.157$, $En_3 = 1$. $He_{-3} = He_3 = 0.074$, $He_{-2} = He_2 = 0.134$, $He_{-1} = He_1 = 0.245$, $He_0 = 0.281$.

③ The seven-scale language value is converted to seven clouds: $Y_{-3}(2, 1.779, 0.074)$, $Y_{-2}(3.326, 1.589, 0.134)$, $Y_{-1}(4.292, 1.265, 0.245)$, $Y_0(5, 1.157, 0.281)$, $Y_1(5.708, 1.265, 0.245)$, $Y_2(6.674, 1.598, 0.134)$, $Y_3(8, 1.779, 0.074)$.

④ Calculated u_{ij} by formula (14)–(15), the results are shown in Table 8.

Table 8. Partial utility value result at attribute level

Attributes i	Attribute level j	Partial utility value u_{ij}
CPU	Eight core+3g	0.1030
CPU	Eight core+4g	0.1778
CPU	Eight core+6g	0.4008
Body memory	128g	0.4163
Body memory	32g	0.0527
Body memory	64g	0.1878
battery capacity/mAh	3000–3500	0.1725
battery capacity/mAh	3500–4000	0.3862
battery capacity/mAh	4000–5000	0.1396
Screen size/inch	5–5.5	0.2075
Screen size/inch	5.5–6	0.3533
Screen size/inch	6–6.5	0.1610
Front camera pixels/10,000 pixels	500–1000	0.1652
Front camera pixels/10,000 pixels	1000–1600	0.3775
Front camera pixels/10,000 pixels	2000–2500	0.1636
Rear camera pixels/10,000 pixels	800 or less	0.0341
Rear camera pixels/10,000 pixels	1200–1900	0.4609
Rear camera pixels/10,000 pixels	2000–2400	0.0990
Fingerprint recognition	Support	0.4962
Fingerprint recognition	Unsupport	0.0149
Resolution	1920 * 1080	0.2797
Resolution	2340 * 1080	0.3334
Resolution	1440 * 720	0.1003

Cost of Mobile Phone Attribute Level. The research in this paper only considers the cost of mobile phone hardware. Different brands of mobile phones use different devices, and it is difficult to uniformly demarcate their attribute levels. Therefore, it is assumed that all types of mobile phones use the same accessories, such as speakers, from the same manufacturer. Through the investigation and analysis of the mobile phone bill of materials, the cost price of the hardware of different attribute levels of the mobile phone can be known. The statistical results are shown in Table 9.

Table 9. Cost of attribute level hardware

Attributes i	Attribute level j	Attribute level cost f_{ij}/thousand yuan
CPU	Eight core+3g	0.2
CPU	Eight core+4g	0.25
CPU	Eight core+6g	0.3
Body memory	128g	0.2
Body memory	32g	0.05
Body memory	64g	0.1
battery capacity/mAh	3000–3500	0.03
battery capacity/mAh	3500–4000	0.04
battery capacity/mAh	4000–5000	0.08
Screen size/inch	5–5.5	0.12
Screen size/inch	5.5–6	0.15
Screen size/inch	6–6.5	0.2
Front camera pixels/10,000 pixels	500–1000	0.08
Front camera pixels/10,000 pixels	1000–1600	0.1
Front camera pixels/10,000 pixels	2000–2500	0.12
Rear camera pixels/10,000 pixels	800 or less	0.09
Rear camera pixels/10,000 pixels	1200–1900	0.13
Rear camera pixels/10,000 pixels	2000–2400	0.15
Fingerprint recognition	Support	0.08
Fingerprint recognition	Unsupport	0
Resolution	1920 * 1080	0.1
Resolution	2340 * 1080	0.14
Resolution	1440 * 720	0.08

Phone Price. In this paper, the price p of the mobile phone is set to 120%, 150%, 180%, 210%, and 230% of the cost price, and the weight of price influence is set to $\gamma = 0.1372$ according to Table 5.

4.3 Experimental Results and Analysis

In this paper, the improved ABC (Artificial Bee Colony) algorithm is used in MATLAB, and the test function is used to make a detailed comparative analysis of the results before and after the improvement. It is concluded that the improved ABC (Artificial Bee Colony) algorithm can effectively compensate for the shortcomings of the original algorithm local optimization, and the convergence speed has also increased to a certain extent.

Algorithm Comparison Results and Analysis. The original version of the ABC algorithm, the Reference [15] algorithm (IABC) and the improved ABC algorithm in this paper were simulated using MATLAB in this section. The initial solutions of the three algorithms are randomly generated. Set the size of the population $N = 100$, the search spatial dimension $Dim = 50$, the maximum number of iterations $MCN = 2000$, and the number of cycles $limit = 100$. The original ABC algorithm, the Reference [15] algorithm (IABC) and the improved ABC algorithm in this paper were tested on the Rosenbrock function and the Rastrigin function, and the test results were compared one by one. The parameters of each test function are shown in Table 10. The test results of the three algorithms are shown in Fig. 3 and Fig. 4.

Table 10. Expressions, search interval, and minimum values of the four test functions

Function name	Function expression	Search space	Minimum value
Rosenbrock	$f_1(x) = \sum_{i=1}^{n} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2$	[−100, 100]	0
Rastrigin	$f_2(x) = \sum_{i=1}^{n} (x_i^2 - 10(\cos(2\pi xi)) + 10)$	[−5.12, 5.12]	0

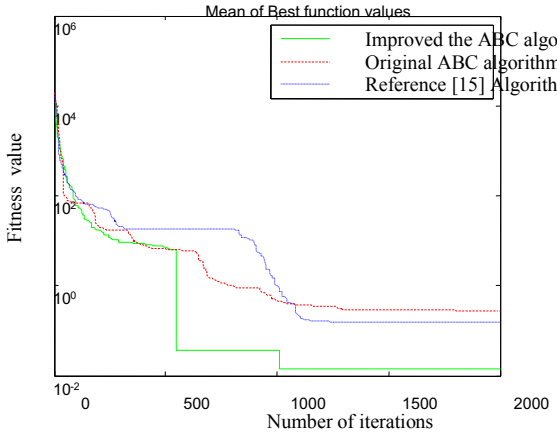

Fig. 3. Fitness changes of Rosenbrock function

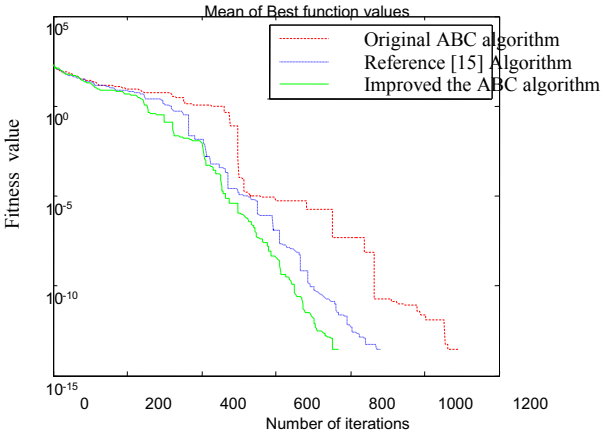

Fig. 4. Fitness changes of Rastrigin function

From Fig. 3 and Fig. 4, the original ABC algorithm will fall into local optimum and slow convergence rate in the test function can be obtained; Although the algorithm of Reference [15] is reduced in the number of iterations and the convergence rate is improved compared with the original algorithm, but it still lacks in the global search ability; The improved ABC algorithm in this paper combined with the reverse learning strategy to improve the population initialization, and has a good global search ability. In terms of search strategy, from Fig. 4, after introducing the current optimal and suboptimal solution, the local search capability of the algorithm has been improved to a certain extent, the convergence rate has also been accelerated, and the number of iterations has been reduced. When the detection bee is initialized, form Fig. 3, after introducing the Gaussian distribution factor, it can help individuals jump out of the local optimal solution, thereby improving the accuracy of the solution.

According to the above analysis, it can be obtained that through the experimental comparison of the two test functions, the improved ABC (Artificial Bee Colony) algorithm has improved resolution and convergence rate compared to the Reference [15] algorithm and the original ABC (Artificial Bee Colony) algorithm. To a certain extent. The defect that the ABC (Artificial Bee Colony) algorithm is easy to fall into the local optimal solution and the shortcoming of the later convergence rate is relatively slow are solved to a certain extent.

Comparison of Solution Results of Product Function Attribute Configuration Model

Figure 5 shows an iterative graph of product line profits. As shown in the figure, when the original ABC algorithm is used to solve the model, it takes 51 iterations to find the optimal solution, and the improved ABC algorithm in this paper finds the optimal solution after 29 iterations, indicating that the algorithm in this paper is better and the speed of convergence is accelerated, and the global search ability is also improved. Through continuous iteration, the value of the objective function is increasing until the optimal solution is found, that is, the value of the objective function is the largest, and

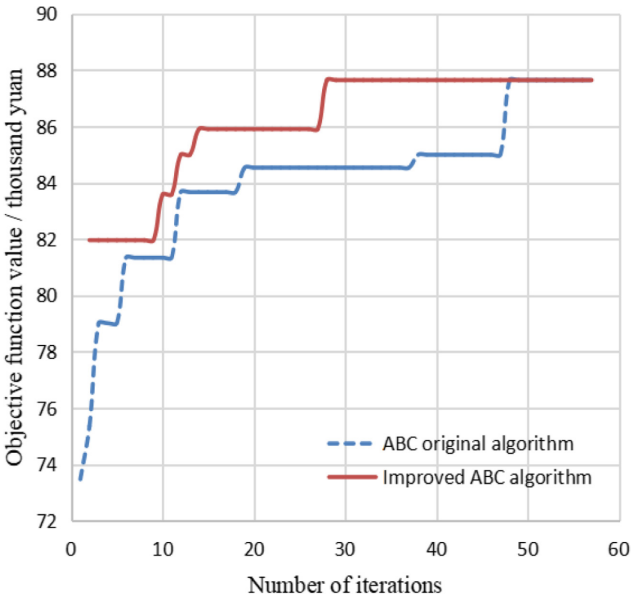

Fig. 5. Iterative graph of product profit

Table 11. Optimal solutions for product positioning

Attribute	Attribute level
CPU and running memory	8 cores+6g
Body memory	128g
Screen resolution	2340 * 1080
Front camera pixels	10–16 million
Rear camera pixels	12–19 million
Fingerprint recognition	Support
Battery capacity	3500–4000 mAh
Screen size	5.5–6 in. screen
Cost	1.140 thousand yuan
Price	2.070 thousand yuan

the product profit is also the largest. According to the operation results of the algorithm, when the expected number of customers is $Q = 200$, the maximum total profit value of the new product generated is $EP = 87.64$ thousand yuan, and the profit of each new mobile phone product is 0.93 thousand yuan. In Table 11, the optimal solution of the ABC (Artificial Bee Colony) algorithm is given, and the optimized configuration and price of the new product are obtained.

The above table shows that the positioning of the product is not a combination of all the optimal attributes, but to re-match the attribute levels of these attributes, which reduces the product attribute configuration that some users do not pay much attention

to, it also reduces the price of the product, and finally the goal of maximizing product profit is achieved. In this article, the es database is mainly used for massive data storage, and the big data hadhoop platform mapreduce parallel computing for comment word segmentation mining, which speeds up the running speed and calculation accuracy.

5 Conclusion

In order to improve the accuracy of product function attribute configuration, this paper proposes a method of mining product customer demand and function attribute configuration driven by big data. The mapreduce parallel computing mining was used on the Hadoop platform to determine the product attributes, attribute levels, etc., and the efficiency of calculation was greatly improved. Then customer preferences were obtained through the questionnaire, and the customer preference function is improved. The product function attribute configuration model was established based on the discrete selection model MNL, and the improved ABC (Artificial Bee Colony) algorithm was used to solve the model. Finally, an empirical analysis was carried out on the case of mobile phone products.

Acknowledgments. This work was supported by the National Natural Science Foundation of China granted 71961005 and the Guangxi Science and Technology Program granted 1598007-15.

References

1. David, L., Ryan, K., Gary, K., Alessandro, V.: Big data. The parable of Google Flu: traps in big data analysis. Science (New York, N.Y.) **343**(6176), 1203–1205 (2014)
2. Jin, S.: Based on Hadoop Practice. Machinery Industry Press, Beijing (2011)
3. Alyass, A., Turcotte, M., Meyre, D.: From big data analysis to personalized medicine for all: challenges and opportunities. BMC Med. Genomics **8**(1), 1–12 (2015)
4. Yang, J.: Data mining technology in the view of cloud computing. Electron. Technol. Softw. Eng. (05), 151 (2019)
5. Zhai, G., Cao, Y.: Research on brand choice of mobile phone consumers in China: empirical analysis based on discrete choice model. Mod. Commer. Ind. **20**(01), 55–56 (2008)
6. Zhai, X., Cai, L., Kwong, C.K.: Multi-objective optimization method for product family design. Comput. Integr. Manuf. Syst. **17**(07), 1345–1355 (2011)
7. Luo, X., Cao, Y., Kuang, Z.: New product positioning model and algorithm considering negative utility. J. Syst. Eng. **28**(06), 820–829 (2013)
8. Li, C., Pang, C., Li, M.: Application of weight-based Apriori algorithm in text statistical feature extraction method. Data Anal. Knowl. Discov. **1**(09), 83–89 (2017)
9. Wang, Q.S., Jiang, F.S., Li, F.: Multi-label learning algorithm based on association rules in big data environment. J. Comput. Sci. **47**(05), 90–95 (2020)
10. Yang, Q., Zhang, Y.W., Zhang, Q., Yuan, P.L.: Research and application of multi-dimensional association rule mining algorithm based on hadoop. Comput. Eng. Sci. **41**(12), 2127–2133 (2019)
11. Luo, Z.H., Che, Y., Yang, Z.W.: Research and analysis of massive data mining algorithm based on hadoop platform. Digit. Commun. World (07), 67–68 (2019)

12. Wang, Q.S., Jiang, F.S.: An improved Apriori algorithm under Hadoop framework. J. Liaoning Univ. (Nat. Sci. Ed.) **46**(03), 257–264 (2019)
13. Wang, J.Q., Peng, L., Zhang, H.Y., et al.: Method of multi-criteria group decision-making based on cloud aggregation operators with linguistic information. Inf. Sci. **274**(274), 177–191 (2014)
14. Bao, G., Lian, X., He, M., et al.: A two-dimensional semantic improvement model based on new language evaluation scale. Control and Decis. **25**(05), 780–784 (2010)
15. Yu, J., Zheng, J., Mei, H.: K-means clustering algorithm based on improved artificial bee colony algorithm. J. Comput. Appl. **34**(04), 1065–1069+1088 (2014)