Logistic Regression Model Optimization and Case Analysis

Xiaonan Zou
Wuhan University of Technology
Hubei Key Laboratory of Modern Auto Parts Technology
Wuhan, China
1097236067@qq.com

Yong Hu
China Quality Certification Center
Wuhan Branch of China Quality Certification Center
Wuhan, China
65681073@qq.com

Zhewen Tian
Wuhan University of Technology
Hubei Key Laboratory of Modern Auto Parts Technology
Wuhan, China
tzw0290@163.com

Kaiyuan Shen
Wuhan University of Technology
Hubei Key Laboratory of Modern Auto Parts Technology
Wuhan, China
18827018002@qq.com

***Abstract*—Traditional logistic regression analysis is widely used in the binary classification problem, but it has many iterations and it takes a long time to train large amounts of data, which is not applicable. In this paper, we study the mathematical model of logistic, define the error function, find the regression coefficient by gradient descent method, and improve the Sigmiod function. Finally, the number of iterations is reduced, and the classification effect is better, and the accuracy is basically unchanged. Besides, this paper establishes a vehicle evaluation prediction model to predict whether consumers accept a certain car. It provides a certain reference for the binary classification problem.**

***Keywords*—logistic regression; binary classification; Sigmiod function; vehicle evaluation**

I. INTRODUCTION

In recent years, deep learning has developed rapidly, but deep learning requires a large amount of data sets and has high hardware requirements for computers. For some simple binary classification problems, machine learning can be completed, and the application of machine learning in this aspect is very mature, so machine learning still has important research value. Among them, Logistic regression has been widely used as a broad data processing method in terms of binary classification and predictions[1]. Domestic research on Logistic regression mainly focuses on application. A large number of researchers seldom carefully explore the underlying theoretical models and assumptions when applying Logistic regression algorithm, so there will be unreasonable applications. In the current statistical textbooks, the contents of the logistic regression model are only understated, and there is no in-depth study. Therefore, it is necessary to conduct an in-depth study on the theory of the logistic regression model in order to improve the algorithm and further improve the efficiency of the Logstic regression model[2].

II. LINEAR REGRESSION AND LOGISTIC REGRESSION

Linear regression is a statistical analysis method that uses the regression analysis in mathematical statistics to determine the quantitative relationship between two or more variables. Taking two variables as an example, (Y1, Y2, ..., Yi) is a dependent variable, and (X1, X2, ..., Xi) is an independent variable. When the dependent variable and the independent variable exhibit a linear relationship, the least squares function in the linear regression equation can be used to establish a mathematical model of the relationship between the independent variable and the dependent variable. A regression analysis based on this linear model is called linear regression. Its purpose is to find the most suitable parameters, and use a straight line to fit the hashed data points, as shown in Figure 1.

$$\sigma(z) = \frac{1}{1+e^{-z}} \qquad (1)$$

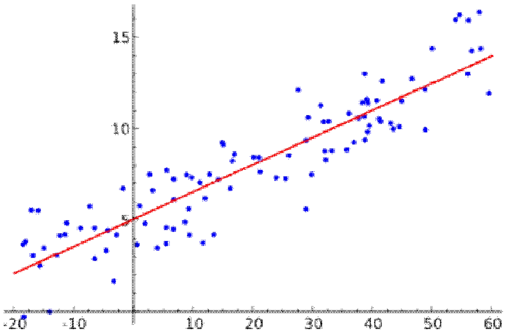

Fig 1. Linear regression diagram

Logistic regression is mainly for classification. The biggest difference between it and linear regression is that its data points are not arranged in line rows. It may be a bunch here, there is a pile, each pile represents a category, and each type of data point has the same category label. As shown in Figure 2, for logistic regression, we want to find the boundary line of the classification, which is represented by the regression formula. The training classifier uses the optimization algorithm to find the best regression coefficient in the regression formula[3] [4].Logistic regression-based classification is given an arbitrary set of inputs, and then the output is obtained by a function, which is the classification of the input data. For example, to simplify processing, when classifying, the function output 0 or 1 in the two classifications represents two classes. According to the actual needs and the above analysis, the above function argument range is from positive infinity to negative infinity. The dependent variable range is 0 or 1. There are many functions that satisfy the above conditions. The most intuitive one is the 0-1 step function. However, the step function is not steerable at the step point, which is not conducive to mathematical processing. Therefore, the Sigmoid function is now widely used, and its image is shown in Fig. 3.

978-1-7281-3299-0/19/$31.00 ©2019 IEEE

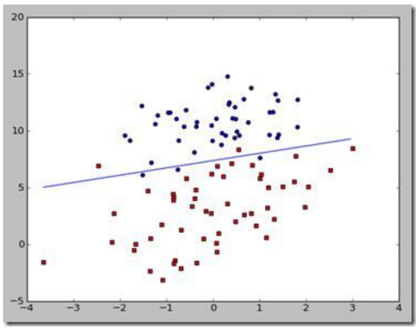

Fig. 2. Logistic regression

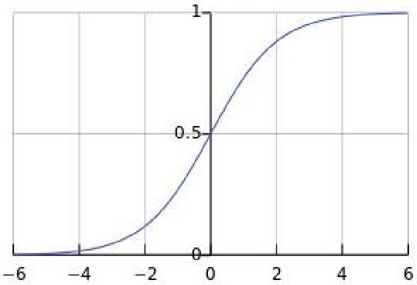

Fig. 3. Sigmoid function

As can be seen from the figure, σ(0)=0.5. When z>0, the function value approaches 1 and becomes class 1 as z increases. When z<0, the function value approaches 0 and becomes 0 class, which satisfies the above classification function requirements. The classification process of Logistic regression can be described as follows: It is assumed that the characteristics of the input data can be expressed as (x0, x1, x2, ..., xn), and each feature is multiplied by a regression coefficient (w0, w1, w2, ... , wn), then sum the input z[4]-[6] as the sigmoid function:

$$z = w_0x_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n \tag{2}$$

which is:

$$z = w^Tx \tag{3}$$

In equation (3), w is the row vector, the regression coefficient, x is the column vector,the input data of the classifier.

The output is a value between 0 and 1, the data with output greater than 0.5 is classified into class 1, and the data with output less than 0.5 is classified as class 0. What is now determined is the best regression coefficient (w0, w1, w2, ..., wn) in this classifier.

III. Analysis of Regression Coefficients

The principle of determining the regression coefficient is to make all classifications as correct as possible, that is, the value predicted by the classification algorithm (y'=σ(z)) and the actual category label (y=0 or 1) have the smallest error, so the error function Y can be defined as:

$$Y = y' - y \tag{4}$$

The error is positive, so its expression can be added to the absolute value.

$$Y = |y' - y| \tag{5}$$

For the convenience of mathematical processing, the error function can be defined as:

$$Y = \frac{1}{2}(y' - y)^2 \tag{6}$$

The error function $\frac{1}{2}(y' - y)^2$ is a function of the regression coefficient w. The value of w is required to minimize the function $\frac{1}{2}(y' - y)^2$. According to the calculus knowledge, a gradient can be used. The descent method is iterated and its expression is as follows:

$$w = w - \alpha\frac{\partial Y}{\partial w} \tag{7}$$

In equation(4) - (7), w is the regression coefficient row vector; Y is the error function; α is the iteration step size; y' is the prediction category, $y'=\sigma(z)=\sigma(w^Tx)$; y is the category label, y=0, 1.

According to the definition of the previous error function, the gradient descent method finally requires the minimum value of the error Y. According to the function definition, the actual Y increases with the number of iterations, and the infinity approaches 0, which is not equal to 0. Therefore, it is only necessary to set a certain threshold (such as e^7), Y is less than this threshold or reaches the set number of iterations is the optimal solution. The step size in the gradient descent method is an empirical value based on the error function. In theory, the smaller the actual error function, the smaller the number of iterations, and the faster the algorithm programming training samples. In combination with the first section, the 0~1 step function is unguided and the Sigmiod function is selected. However, when the Sigmiod function argument is around 0, the dependent variable is far from the real label (0, 1), that is, the error function value is larger. Although the minimum error can be found by the gradient descent method, the number of iterations is large, which affects the algorithm execution efficiency.

Therefore, this paper proposes a radical Sigmiod function, which turns the bottom e of the Sigmiod function into e^n(where n>=1), namely:

$$\sigma(z) = \frac{1}{1+e^{-nz}} \tag{8}$$

According to the nature of the function, the value of n is changed. The larger n is, the closer the function value is to 0 or 1, that is, the smaller the error function y is. When other factors are constant, the faster the optimal solution is obtained, the algorithm execution efficiency is higher.

IV. Solving Mathematical Models

From the previous chapter, the gradient descent method finds the regression coefficient W when the error Y reaches the minimum value. The main step is to find the partial derivative $\frac{\partial Y}{\partial W}$. According to the calculus derivative, there are:

$$\frac{\partial Y}{\partial w_j} = \sum_1^m \left((y' - y) \frac{\partial y'}{\partial w_j} \right) \quad (9)$$

For the function σ(z) there are:

$$\sigma'^{(z)} = n\sigma(z) * (1 - \sigma'^{(z)}) \quad (10)$$

So:

$$\frac{\partial y'}{\partial w_j} = ny'(1 - y') \frac{\partial (w^T x_i)}{\partial w_j} \quad (11)$$

Among them::

$$\frac{\partial (w^T x_i)}{\partial w_j} = \frac{\partial (w_0 x_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n)}{\partial w_j} \quad (12)$$

When j =0, the partial derivative is obtained for w_0, and x_0 is obtained, and other items are also the same. Therefore, when$j$=0, the partial derivative is obtained for w_0, and x_0 is obtained, and other items are also the same, so:

$$\frac{\partial (w^T x)}{\partial w_j} = x_j \quad (13)$$

Therefore, by formula (9), (10), (11), (12), (13) can be introduced as follows:

$$\frac{\partial \mathrm{Y}}{\partial \mathrm{w_j}} = \sum_1^{\mathrm{m}} \left(\mathrm{y}'(1 - \mathrm{y}')\mathrm{x_j} \right) \quad (14)$$

Therefore, the gradient descent method iteration formula is:

$$w = w - \alpha \sum_1^m \left(y'(1 - y')x_j \right) \quad (15)$$

In the logistic regression model, x is the training set, each row represents one sample, $w$represents the regression coefficient, and$y$represents the category.

$$x = \begin{pmatrix} x_0^1 & x_1^1 & \cdots & x_n^1 \\ x_1^2 & x_1^2 & \cdots & x_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_0^m & x_1^m & \cdots & x_n^m \end{pmatrix} \mathrm{w} = \begin{pmatrix} w_0 \\ w_1 \\ \cdots \\ w_n \end{pmatrix} \mathrm{y} = \begin{pmatrix} y^1 \\ y^2 \\ \cdots \\ y^m \end{pmatrix} \quad (16)$$

$$\mathrm{A} = xw = \begin{pmatrix} w_0 x_0^1 + & w_1 x_1^1 + & \cdots & w_n x_n^1 \\ w_0 x_0^2 + & w_1 x_1^2 + & \cdots & w_n x_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ w_0 x_0^m + & w_1 x_1^m + & \cdots & w_n x_n^m \end{pmatrix} \quad (17)$$

Then the above error can be expressed as:

$$y' - y = \begin{pmatrix} \sigma\left(\mathrm{A}^1\right)\text{-}\mathrm{y}^1 \\ \sigma\left(\mathrm{A}^2\right)\text{-}\mathrm{y}^2 \\ \cdots \\ \sigma\left(\mathrm{A}^n\right)\text{-}\mathrm{y}^n \end{pmatrix} \quad (18)$$

According to the matrix algorithm, the integrated formulas (14), (15), (16), and (17) can obtain the gradient descent formula:

$$\mathrm{w} = \mathrm{w} - \alpha n x^T \left(y' - y \right) \left(1 - y' \right)^T y \quad (19)$$

In the previous section, the mathematical model of the gradient descent method has been obtained. According to the above analysis, the regression coefficient at the minimum of the error function can be obtained by iteration. The specific algorithm 1 is as follows:

Step1 initializes the regression coefficient w, the step size α, the number of iterations and other parameters.

Step 2 Repeat 3-5 until the termination condition is met (the termination condition is the number of iterations).

Step3 substitutes the parameters and calculates the $A = wx$ matrix.

Step4 Substitutes the result of step3 calculation into a function, finds the value of σ(A), and calculates $y' - \mathrm{y}$.

Step5 Update w by iterative formula.

In order to implement the algorithm, we need to program. The commonly used machine learning programming languages are C++ and Python. The C++ code is complex, the compilation is faster, the python code is simple, easy to implement, and the compilation is slow. The data set used in this paper is small and comprehensively considered. The algorithms in this paper are all implemented in Python.

V. Experimental Verification and Analysis

A. Algorithm efficiency verification

In order to quickly find the rules and simplify the processing, this paper selects 100 sets of data sets for classification training. Each set of data contains a category label (0 or 1). Using python to program the previous iteration algorithm 1 can find the regression coefficient$w$, according to the optimal regression coefficient w, can be listed as the best fit straight line equation, according to the nature of the sigmoid function defined above, the abscissa is 0 is the boundary between two categories (0 and 1), so that the sigmoid function The input z is 0, and the equation for the dividing line is obtained:

$$w_0 x_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n = 0 \quad (20)$$

Through python programming, draw various data points and dividing lines to clearly see the classification effect map.

When n is equal to 1, 5, 10 in the Sigmiod function, the classification effect is as shown below.

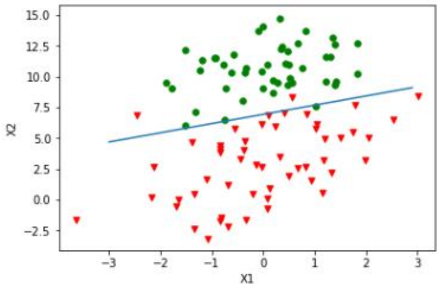

Fig. 4. Classification effect diagram when n=1

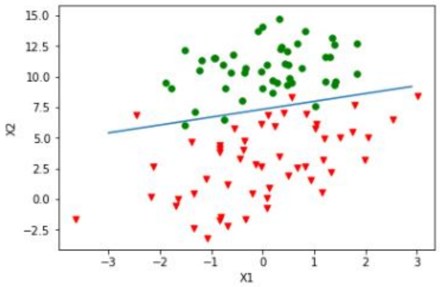

Fig. 5. Classification effect diagram when n=5

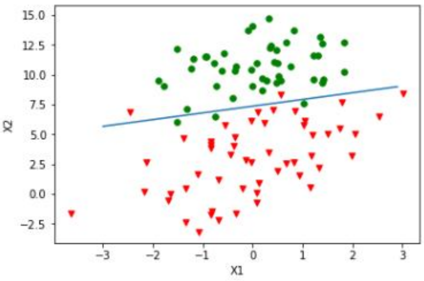

Fig. 6. Classification effect diagram when n=10

According to the actual meaning of the above figure, the straight line in the figure will separate the two categories of red and green. For the points near the regression line, the error point should be classified as close as possible to the straight line. Comparing with Figures 4, 5 and 6, it can be seen that the classification is n=10. Better, the classification is even worse when n=1.

In order to verify the efficiency of the execution, we draw the relationship between regression coefficient and iteration number through pythonprogramming,as shown in the following figure:

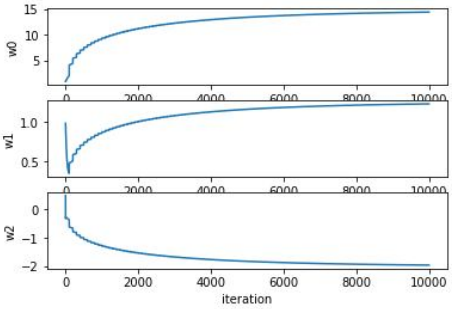

Fig. 7. Relationship between regression coefficient and iteration number when n=1

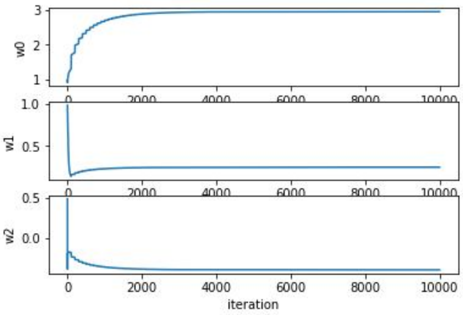

Fig. 8. Relationship between regression coefficient and iteration number when n=5

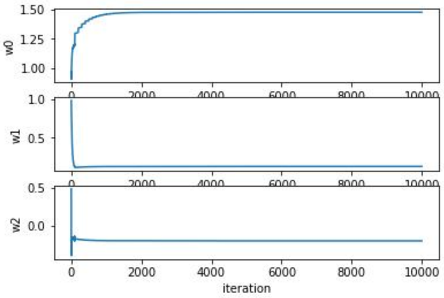

Fig. 9. Relationship between regression coefficient and iteration numberwhen n=10

It can be seen from the above figure that when n=1, iterative 8000 regression coefficients w can converge. When n=5, iterative 2500 regression coefficients converge. When n=10, iterative 1000 regression coefficients can converge, and the comparison results can be seen. The larger n is, the smaller the number of iterations is needed, that is, the algorithm execution efficiency is higher.

B. Accuracy verification

In order to verify the reliability of the algorithm, another 500 sets of data sets were tested. The first 90% of the data set was used as the training sample. It was solved iteratively by the gradient descent method to find the optimal weight coefficient w, and the remaining 10% was used as the test set. After the optimal regression coefficient w(w0, w1, w2, ..., wn) is obtained by the algorithm 2, the accuracy is obtained by the algorithm, and the algorithm 2 is as follows:

Step1 Initialize the number of tests, test the correct number of times, and test the category (0 or 1).

Step 2 Repeat 3-5 until the termination condition is met (travel the test set).

Step3 The number of tests is increased by one. According to the best regression coefficient obtained above, the value of $z = w_0x_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n$ is calculated and substituted into the Sigmiod function.

Step4 If σ(z)>0.5, the judgment is classified into 1 class, and the test category is equal to 1; otherwise, the test category is equal to 0.

Step5 If the test category is equal to the category label, the correct number of tests is incremented by 1.

Step6 Returns the number of correct tests divided by the number of tests.

According to the above algorithm, when n=1, 5, 10 respectively, the accuracy rate is 96.14% when n=1, the accuracy rate is 96.14% when n=5, and the accuracy rate is 96.14% when n=5, which shows that the change The value of n has little effect on accuracy.

According to the above analysis, increasing the value of n in the Sigmiod function can reduce the number of iterations, the classification effect is better, the classification accuracy rate is basically unchanged, and the execution efficiency of the algorithm can be improved.

VI. Automotive Evaluation Application Examples

Many auto companies predict that consumers accept a certain car, the variables and influencing factors are more, the general data analysis method is difficult to accurately predict whether consumers accept a certain car, Logistic regression model can solve such problems well. .

First, comprehensive consumer purchasing habits, select the price, appearance, maintenance costs, number of seats, engine displacement, safety, comfort, power, door number, domestic or imported and other influencing factors as characteristic values, of which for non-value Type factor, metaphor safety is divided into A, B, C categories, which can be converted into 100, 010, 001 values to process, and finally 21 eigenvalues can be obtained.

This article selects the dataset of a website, the first 90% as the training set, and the last 10% as the test set. Select the initial value $w=(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,111,1,1,1)^T$, step size $\alpha=0.001$, Substituting the above training set into algorithm 1 for iteration, and finding the best regression coefficient after programming in python

W0=78.59, w1=25.94962911, w2=833.06, w3=-68.75, w4=247.2, w5=-69.00, w6= -81.55, w7=-215.85, w8=-17.5, w9=-161.20, w10=47.45, W11=-73.75, w12=84.40, w13=-46.25, w14=99.415, w15=50.1, w16=-59.30, w17=8.75, w18=240.265, w19=-57.2, w20=-59.415. After determining the regression coefficient, it is possible to predict whether the consumer accepts the car based on various parameters of a certain car.

In order to detect whether the prediction model is accurate, the above regression coefficient is substituted into the algorithm 2, and the accuracy rate is finally determined by python programming to be 95.15%.

VII. Conclusion

Based on linear regression, this paper studies the mathematical model of logistic model, defines the error function, obtains the regression coefficient by gradient descent method, and improves the Sigmiod function, so that the binary classification effect is better, and the number of iterations is reduced, and the accuracy is basically unchanged. The algorithm of the binary classification is optimized, and finally it is concluded that the larger the n in the Sigmoid function σ(z)=1/(1+e^(-nz)), the smaller the number of iterations required to achieve the same precision. Finally, combined with the example, the vehicle evaluation prediction model is established to predict whether the consumer accepts a certain car.

Acknowledgment

First of all, I would like to thank the teacher Tian Zhewen for giving me careful guidance in study, research and life. In the study, Mr. Tian is rigorous in studying and thinking sharply. He and us learn to participate in the whole project and teach us to think about problems scientifically. I would also like to thank the same lab brothers, who helped me a lot in the writing of the thesis, which enabled me to overcome a series of difficult problems in scientific research and made progress again and again.

References

[1] S. Rongheng, Applied Mathematical Statistics (3rd Edition) ,CA: Science Press, 2014.

[2] C. Zhenhai, Liu. Wei, “Logistic Regression Model and Its Application,” Journal of Yanbian University(Natural Science Edition), vol. 38(01), pp 28-32, 2012.

[3] Z. Qian, Meng. Deyu, Xu. Zongben, “L_(1/2) Regularized Logistic Regression,” Pattern Recognition and Artificial Intelligence, vol. 25(05), pp. 721-728, 2012.

[4] Bowes, X. Yu, Statistical methods for classification data analysis, CA: Social Sciences Academic Press. 2009

[5] Y. Meiyu, Data Mining and Machine Learning - WEKA Application Technology and Practice (Second Edition), CA: Tsinghua University Press. 2016.

[6] Y. Juan, Python machine learning actual combat, CA: Science and Technology Literature Publishing House,2018.