Application of Logistic Regression with Fixed Memory Step Gradient Descent Method in Multi-Class Classification Problem

Yuan Sun, Zhihao Zhang
Department of Electronics and Information Engineering
Tongji Zhejiang College
Jiaxing 314051, Zhejiang Province, China

Zan Yang*, Dan Li
Department of Science
Tongji Zhejiang College
Jiaxing 314001, Zhejiang Province, China
yangzan953@163.com

***Abstract*—Logistic regression is a supervised binary classification algorithm in machine learning. It is an important part of neural network and convolutional neural network. An important part of the logistic regression algorithm is to find the optimal parameters of the loss function, which is often a non-linear convex optimization problem. Generally, the gradient descent method or the stochastic gradient descent method in linear search are used in this nonlinear problem, but these methods are easy to fall into the trap of local minimum and have weak global convergence. In order to preserve the gradient information better and enhance the global convergence of the algorithm, this paper introduces fixed memory step gradient descent method into the optimization part of logistic regression algorithm, and combines OVR strategy to solve the problem of multi-class classification of data.**

Keywords—logistic regression; fixed memory step gradient descent method; multi-class classification

I. Introduction

Logistic regression is a classical binary classification algorithm based on supervised machine learning theory. It is widely used in tasks such as click rate estimation (CTR), computational advertising (CA), and recommendation system (RS) because of its ease of implementation, good interpretability, and ease of expansion. Logistic regression has attracted the attention of many scholars, and has also obtained a lot of research results, such as [1]–[10]. The main idea of the logistic regression algorithm is to use the logistic function, that is, the sigmoid function to de-linearize the multiple linear regression function to achieve the effect of data classification and generalization of the model. If you want to make the logistic regression classification as accurate as possible, the classical logistic regression requires the optimal parameters of the loss function to minimize the error. Usually we use the gradient descent method to optimize the logistic regression loss function to find the optimal solution. Since the gradient descent method iterative gradient is only related to the previous iterative gradient, it does not take into account the influence of the previous iterations of the gradient values, causing errors. We use the fractional gradient descent of the fixed memory step to study the optimal parameters of the logistic regression loss function.

As an important branch of mathematics, the fractional gradient method is considered to be an excellent tool to improve the traditional gradient descent method, mainly because of its special long memory characteristics and non-locality. Since the common definition of the fractional derivative contains the special integral of the gamma function, it highlights the long memory feature rather than the local nature of the function.Since the fractional differentials of a function is approximately equal to the sum of the higher-order differentials of the function to some extent, the fractional gradient descent method has a faster convergence rate than the traditional gradient descent method.Because of this advantage, the fractional gradient descent has faster speed and more accurate accuracy than the conventional gradient descent method.

In this paper, inspired by literature [11], we first introduced the fractional ladder descent method with fixed memory step size based on caputo definition to optimize the cross entropy loss function of logistic regression, and made its classification accuracy higher through parameter optimization, which is the innovation point of this paper. Through parameter optimization, the classification of logistic regression has higher accuracy. At the same time, we adopt OVR (One vs Rest) strategy, that is, a one vs multi strategy of multi-class classification learning, and change the binary classification into multiple classifications to make logistic regression applictable and the area is wider. We propose a solution to the multi-class classification problem of logistic regression based on fractional gradient descent methodoptimization.

This paper is organized as follows. In section 2, we introduce the classical logistic regression theory and analyze its drawbacks. We establish an algorithm for solving the optimal parameters of loss function based on fractional gradient descent in section 3. In section 4, We use this algorithm to carry out experiments on specific problems.

II. Classical 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. The goal is to find a straight line or a plane or a higher

dimensional hyperplane that minimizes the error between the predicted and true values. It is widely used. The (1) formula is an expression of classical linear regression.

$$\hat{y}^{(i)} = \theta_0 + \theta_1 {X_1}^{(i)} + \theta_2 {X_2}^{(i)} + \ldots + \theta_n {X_n}^{(i)}. \quad (1)$$

For (1), if the weight parameter is expressed in matrix form, the weight vector is $\theta = (\theta_0, \theta_1, \theta_2, \ldots, \theta_n)^{\mathrm{T}}$, if y is used to represent the real-value vector, $\hat{y}$ is used to represent the predicted value vector, where $\hat{y}^{(i)}$ is the ith of the predicted value vector the component, the kth eigenvector is represented by $X_k \, (k = 1 \cdots n)$, where ${X_k}^{(i)}$ is the ith component. Then (1) can be rewritten as:

$$\hat{y}^{(i)} = \theta_0 {X_0}^{(i)} + \theta_1 {X_1}^{(i)} + \theta_2 {X_2}^{(i)} + \ldots + \theta_n {X_n}^{(i)},$$

where

$$X^{(i)} = ({X_0}^{(i)}, {X_1}^{(i)}, {X_2}^{(i)}, \ldots, {X_n}^{(i)}), \; {X_0}^{(i)} \equiv 1.$$

Could be further rewritten (1) as

$$\hat{y}^{(i)} = X^{(i)} \cdot \theta. \quad (2)$$

If the formula (2) is further written as a matrix form

$$\hat{y} = X_b \cdot \theta, \quad (3)$$

where the X_b matrix is

$$X_b = \begin{pmatrix} 1 & {X_1}^{(1)} & {X_2}^{(1)} & \ldots & {X_n}^{(1)} \\ 1 & {X_1}^{(2)} & {X_2}^{(2)} & \ldots & {X_n}^{(2)} \\ \ldots & & & & \ldots \\ 1 & {X_1}^{(m)} & {X_2}^{(m)} & \ldots & {X_n}^{(m)} \end{pmatrix}.$$

We can derive the loss function of linear regression according to the least squares method as follows

$$J = \sum_{i=1}^{m} (y^{(i)} - \hat{y}^{(i)})^2,$$

We need to make it the smallest, this problem can also be equivalent to

$$J = (y - X_b \cdot \theta)^{\mathrm{T}} (y - X_b \cdot \theta), \quad (4)$$

make it the smallest.

By optimizing the loss function, we can get the weight matrix of each independent variable of multiple linear regression as

$$\theta = (X_b^{\mathrm{T}} X_b)^{-1} X_b^{\mathrm{T}} y. \quad (5)$$

When directly calculating (5), too high time complexity will hinder the calculation of the model, then (4) can be regarded as an optimization problem, and the optimal gradient vector can be found by the traditional gradient descent method.

The gradient descent method, it is a classical method for solving the optimal parameters of traditional convex optimization problems. It is often used to optimize the machine learning model. The main idea of fifithe traditional gradient descent method is to select the appropriate initial value and iteration step size and perform the iterative update loss function. The value of the function is such that the loss function is optimized to a minimum until convergence. Since the negative gradient direction is the direction in which the function value drops the fastest, the value of the loss function is updated in the negative gradient direction in each step of the iteration. Therefore, the loss function is minimized, the optimal parameters are determined, the model accuracy is maximized, and the regression effect is optimal.

Logistic regression is a typical binary classification algorithm in machine learning theory. The idea comes from linear regression. Both logistic regression and linear regression belong to generalized linear model. Under the condition of fixed model parameters, for the given independent variables, the values of the model are subject to exponential cluster distribution, linear regression value is subject to normal distribution, and logistic regression value is subject to Bernoulli distribution. The main idea is to add a layer of nonlinear function between the characteristics and results of linear regression, namely, the sigmoid function. By smoothing and non-linear processing the linear regression value, the probability value of regression classification is obtained, so as to classify the nonlinear discrete data.

We use $g\,(z)$ to represent the sigmoid function as follows:

$$g\,(z) = \frac{1}{1 + e^{-z}}. \quad (6)$$

Although the unsaturation of sigmoid function is not strong, it has strong nonlinearity and can be differentiated almost everywhere, so sigmoid function is still one of the most common activation functions. Sigmoid function can project feature space into probability space, which is its great application advantage.

Its argument takes the whole real number, and the value range is 0 to 1. Since the sigmoid function has excellent properties such as antisymmetry, bounded domain, and simple derivative, it is chosen as a mapping function from linear regression to logistic regression.

The prediction function after mapping linear regression to logistic regression is as follows:

$$h_\theta\,(X_b) = g\,(X_b \cdot \theta) = \frac{1}{1 + e^{X_b \cdot \theta}}. \quad (7)$$

For classification problems, the following formula can be used to represent classification results.

$$P\,(\widehat{y} = 1|\, X_b; \theta) = h_\theta X_b,$$
$$P\,(\widehat{y} = 0|\, X_b; \theta) = 1 - h_\theta X_b.$$

When $P \geq 0.5$ is considered to be a positive example($\hat{y} = 1$). when $P < 0.5$, the classification result is considered to be a counterexample($\hat{y} = 0$). Integrate two types to get the following formula,

$$P\,(y|\, X_b; \theta) = (h_\theta\,(X_k))^y (1 - h_\theta\,(X_k))^{1-y}. \quad (8)$$

We use the maximum likelihood estimation to determine the parameter θ in logistic regression. We obtain the maximum likelihood function by maximal likelihood estimation of its

probability

$$L(\theta)=\prod_{i=1}^{m}P(y_i|X_b;\theta)=\prod_{i=1}^{m}(h_\theta(X_b))^{y_i}(1-h_\theta(X_b))^{1-y_i}. \tag{9}$$

To simplify the operation, we construct a log-likelihood function, that is, the maximum likelihood function takes the logarithm of both sides.

$$l(\theta)=\sum_{i=1}^{m}(y_i\log h_\theta(X_k)+(1-y_i)(1-h_\theta(X_k))). \tag{10}$$

When the log-likelihood function reaches its maximum, the classification effect is the most accurate, so we construct a logistic regression based on the log-likelihood function to get the loss function as shown below:

$$J(\theta)=-\frac{1}{m}\sum_{i=1}^{m}(y_i\log h_\theta(X_k)+(1-y_i)(1-h_\theta(X_k))). \tag{11}$$

(11) is the loss function of logistic regression. In general, we use the traditional gradient descent method to optimize it and solve the optimal parameter theta. However, the traditional gradient descent method has certain limitations. Since it only depends on the current point and has no globality and memory, if the step size is set incorrectly during the solution process, it will easily lead to the problem of falling into the local minimum and the operation too slow. To solve this problem, we introduce a gradient descent method with fractional derivatives to optimize the algorithm to achieve a better model solution.

III. Logistic Regression With Fractional Gradient Descent

We can turn the loss function minimization problem into the following unconstrained optimization problem.

$$\min J(\theta),$$

where $\theta\in\mathbb{R}^n, J(\theta)\in\mathbb{R}$ is a real-valued function.

There are usually many ways to solve such problems.The traditional gradient descent method is generally used to optimize the loss function. A nonlinear, unconstrained optimization method is employed. This method usually has two kinds of optimization directions, one is the line search method, and the other is the trust region method. Since the line search method is beneficial to numerical calculation to some extent, most of the optimization methods are based on line search optimization methods, such as traditional gradient descent method, stochastic gradient descent method, etc., because the traditional gradient descent method depend on only local points, has locality and no memory characteristics, and the algorithm converges slowly. To solve this problem, we introduce the gradient descent method with fractional derivative.

Here are some preliminary knowledge of fractional calculus.There are three general definitions of fractional calculus, namely the Riemann-Liouville definition, the Grünwald-Letnikov definition, and the Caputo definition. The fractional derivatives defined by these three fractional calculus are as follows.

$${}_{c}^{\mathrm{RL}}\mathcal{D}_x^\alpha f(x)=\frac{1}{\Gamma(n-\alpha)}\frac{\mathrm{d}^n}{\mathrm{d}x^n}\int_c^x\frac{f(\tau)}{(x-\tau)^{\alpha-n+1}}\mathrm{d}\tau, \tag{12}$$

$${}_{c}^{\mathrm{GL}}\mathcal{D}_x^\alpha f(x)=\lim_{h\to0}\frac{1}{h^\alpha}\sum_{i=0}^{\frac{b-x}{h}}(-1)^x\binom{\alpha}{i}f(x+ih), \tag{13}$$

$${}_{c}^{\mathrm{C}}\mathcal{D}_x^\alpha f(x)=\frac{1}{\Gamma(n-\alpha)}\int_c^x\frac{f^{(n)}(\tau)}{(x-\tau)^{\alpha-n+1}}\mathrm{d}\tau. \tag{14}$$

Where,$n-1<\alpha<n$,$n\in\mathbb{Z}_+$,c is the lower limit,and the Gamma function is $\Gamma(\alpha)=\int_0^{+\infty}\mathrm{e}^{-t}t^{\alpha-1}\mathrm{d}t.\binom{\alpha}{\beta}$ is the binomial coefficient,specifically written as$\binom{\alpha}{\beta}=\frac{\Gamma(\alpha+1)}{\Gamma(\beta+1)\Gamma(\alpha-\beta+1)},\alpha\in\mathbb{R},\beta\in\mathbb{N}$.In order to facilitate computer calculation, we perform $f(x)$ expansion on Taylor, then equations (12) and (14) can be rewritten as follows.

$${}_{c}^{\mathrm{RL}}\mathcal{D}_x^\alpha f(x)=\sum_{i=0}^{+\infty}\binom{\alpha}{i}\frac{f^{(i)}(x)}{\Gamma(i+1-\alpha)}(x-c)^{i-\alpha}, \tag{15}$$

$${}_{c}^{\mathrm{C}}\mathcal{D}_x^\alpha f(x)=\sum_{i=n}^{+\infty}\binom{\alpha-n}{i-n}\frac{f^{(i)}(x)}{\Gamma(i+1-\alpha)}(x-c)^{i-\alpha}. \tag{16}$$

Since the fractional derivative defined by Grünwald-Letnikov is in the limit form, it is inconvenient to calculate, and the fractional derivative defined by Riemann-Liouville is first subjected to integral operation and then differential operation. The Caputo derivative is first subjected to differential operation and then integrated operation. Due to the operation order, Caputo derivatives are more efficient in engineering calculations, so most engineering calculations use fractional derivatives defined by Caputo.

Since the traditional gradient descent method only relies on the current information for iteration, but the fractional descent method relies on the common information of the first n points, and has good long memory characteristics and globality. Inspired by the literature [10], we adopt a fixed step sizefiand the fractional descent method defined by Caputo performs an iterative optimization of the loss function, and its iterative format is as follows.

$$x_{k+1}=x_k-\mu_{x_{k-K}}\mathcal{D}_x^\alpha f(x)\big|_{x=x_k}, \tag{17}$$

where $K\in\mathbb{Z}_+$.

The optimal parameter problem of loss function is usually a convex optimization problem. Linear search and trust region method are the main methods to solve this kind of problem. Considering the computational cost, the linear search optimization method is suitable for machine learning and deep learning. The main idea of linear search method is to search optimally along the descending direction and with a certain step size. The main idea of linear search method is to search optimally along the descending direction and with

a certain step size. At present, the first-order linear search method is the mainstream, and the second-order effective method still encounters a lot of bottlenecks, but recently there has been a first-order linear search method combined with negative curvature filtering algorithm. The descent direction is not unique. Negative gradient direction, Newton direction, Quasi-Newton direction and conjugate gradient direction are all descent directions. In view of the computational cost, the negative gradient iteration scheme has strong applicability, but the traditional gradient descent method is easy to fall into the trap of local minimum, and the global convergence is weak. It is difficult to make a breakthrough in the basic theory of operational research at this stage, so scholars try to apply the good theory of analytical mathematics to convex optimization or non-convex optimization theory. Because the fractional derivative can retain the gradient information better, it also has strong global convergence. In this paper, we introduce the fixed memory step gradient descent method in [11] into the multi-class classification problem of logistic regression.

We can draw the following conclusions from the influence of the literature [12]. When the algorithm is convergent, it will converge to its actual extreme point. The main idea of fifithis method is called as fixed memory principle. We can do this by (17) help (14), so our fractional gradient descent method can be expressed as

$$x_{k+1} = x_k - \mu \sum_{i=1}^{+\infty} \begin{pmatrix} \alpha - 1 \\ i - 1 \end{pmatrix} \frac{f^{(i)}(x_k)}{\Gamma(i + 1 - \alpha)} (x_k - x_{k-K})^{i-\alpha}, \tag{18}$$

where μ is the step size.

We use the data of numerical simulation experiment in [13] to compare the performance of various optimization algorithms. For convenience of expression, we record the number of iterations as NIT. Fixed memory step gradient descent method is recorded as FMSGD. The gradient descent method is recorded as GD. The stochastic gradient descent method is recorded as SGD. $\rho_s(\tau)$ form [11] is the standard indicator of the possibility of solution. It is used to describe the specific performance of the algorithm. τ is the number of calculations or iterations of function values. Figures 1 and 2 show that the CPU time performance profile of FMSGD algorithm is better than that of GD and SGD. Figures 3 and 4 show that FMSGD has better performance profile of iteration times than GD and SGD.

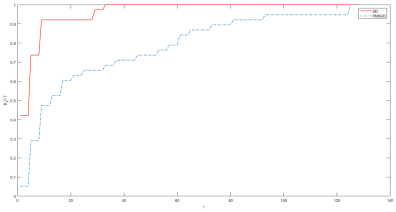

Fig. 1. Performance for CPU Time of GD and FMSGD

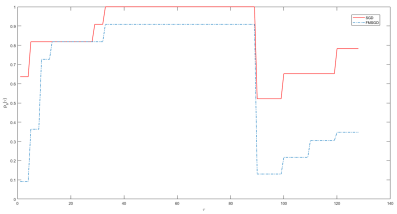

Fig. 2. Performance for CPU Time of SGD and FMSGD

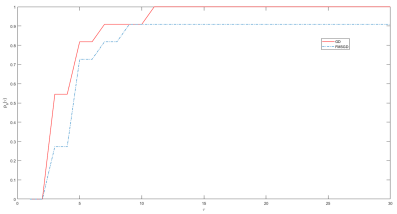

Fig. 3. Performance for NIT of GD and FMSGD

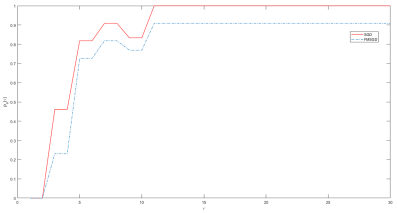

Fig. 4. Performance for NIT of SGD and FMSGD

IV. Simulation Study

The disadvantage of logistic regression is that it can only be classified into two categories. Generally, it is necessary to combine OVR, OVO and MVM with other classification strategies. The strategy chosen in this paper is OVR.

The OVR strategy is a strategy that can transform the binary classification into a multi-class classification problem. The idea is: for the classification problem, first train a binary classifier, and calibrate the prediction result of one of the classifiers to positive during the test. The prediction results of the remaining categories are categorized as anti-class. Let all the data pass through the three binary classifiers, and the result of the final prediction is positive. If the prediction is positive by multiple classifiers, then the prediction confidence is considered, and the category with the highest prediction confidence is selected. In this paper, we use the wine quality evaluation data set from UCI, we use the OVR strategy to establish three logistic regression models, and carry out Multi-Class Classification of the OVR strategy. The classification results are shown in the following figure 5, figure 6 and figure 7.

Since the wine dataset consists of three categories, the following three maps are the classification results of the three

two sorters when the three categories are classified as positive, and one of the categories is taken as a positive example.

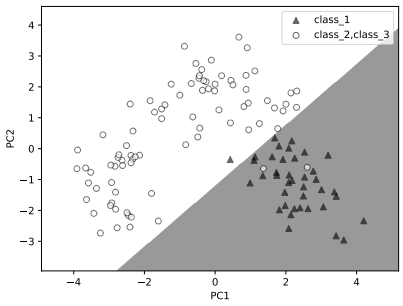

Fig. 5. Class1 vs Rest Simulation Results

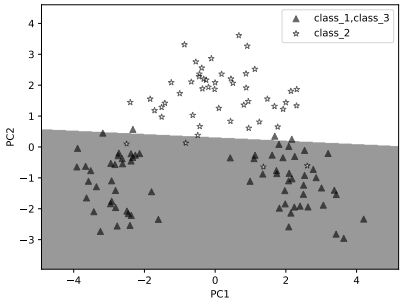

Fig. 6. Class2 vs Rest Simulation Results

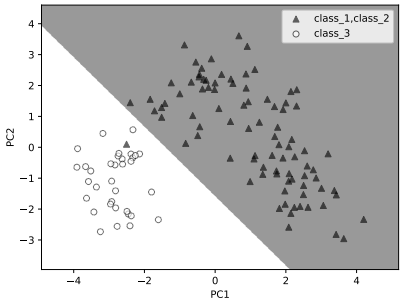

Fig. 7. Class3 vs. Rest Simulation Results

The final three-point classification results are shown in the figure 8.

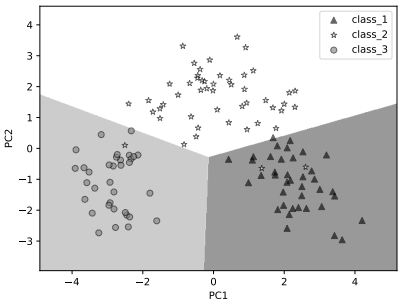

Fig. 8. Three Classification Results

V. CONCLUSION

Combined with numerical simulation and programming practice, it can be seen that fixed memory step gradient descent method has better global convergence and performance than gradient descent method and stochastic gradient descent method. The application of fixed memory step gradient descent method to multi-class classification problems also has a good effect.

ACKNOWLEDGMENT

This work is partly supported by The Seventh Student Innovation Activity Project of Tongji Zhejiang College No.0719038, 2019 Zhejiang University Students Science and Technology Innovation Activity Plan Project No.2019R437005 and The Eighth Teaching Reform Project of Tongji Zhejiang College No.0119029.

REFERENCES

[1] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, "The elements of statistical learning: data mining, inference and prediction," *The Mathematical Intelligencer*, vol. 27, no. 2, pp. 83–85, 2005.

[2] E. B. Khalil, B. Dilkina, G. L. Nemhauser, S. Ahmed, and Y. Shao, "Learning to run heuristics in tree search," in *Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017.

[3] M. Collins, R. E. Schapire, and Y. Singer, "Logistic regression, adaboost and bregman distances," *Machine Learning*, vol. 48, no. 1-3, pp. 253–285, 2002.

[4] H. He, H. Daume III, and J. M. Eisner, "Learning to search in branch and bound algorithms," in *Advances in neural information processing systems*, 2014, pp. 3293–3301.

[5] J. Yu, "State of health prediction of lithium-ion batteries: Multiscale logic regression and gaussian process regression ensemble," *Reliability Engineering & System Safety*, vol. 174, pp. 82–95, 2018.

[6] A. Lodi and G. Zarpellon, "On learning and branching: a survey," *Top*, vol. 25, no. 2, pp. 207–236, 2017.

[7] E. Khalil, H. Dai, Y. Zhang, B. Dilkina, and L. Song, "Learning combinatorial optimization algorithms over graphs," in *Advances in Neural Information Processing Systems*, 2017, pp. 6348–6358.

[8] I. Ruczinski, C. Kooperberg, and M. LeBlanc, "Logic regression," *Journal of Computational and graphical Statistics*, vol. 12, no. 3, pp. 475–511, 2003.

[9] V. Kecman, *Learning and soft computing: support vector machines, neural networks, and fuzzy logic models*. MIT press, 2001.

[10] L. Wang and X. Fu, "Data mining with computational intelligence,(2005)."

[11] Y. Wei, Y. Kang, W. Yin, and Y. Wang, "Design of generalized fractional order gradient descent method," *arXiv preprint arXiv:1901.05294*, 2018.

[12] D. Li and D. Zhu, "An affine scaling interior trust-region method combining with nonmonotone line search filter technique for linear inequality constrained minimization," *International Journal of Computer Mathematics*, vol. 95, no. 8, pp. 1494–1526, 2018.

[13] Z. Wang and D. Zhu, "An affine scaling interior point filter line-search algorithm for linear inequality constrained minimization," *Numerical functional analysis and optimization*, vol. 31, no. 8, pp. 955–973, 2010.