The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:

Peng Cheng

Supervisor:

Mingkui Tan

Student ID:

201830020255

Grade:

Undergraduate

2020-10-24

Linear Regression and Stochastic Gradient Descent

***Abstract*—In this paper, the linear regression problem is transformed into the optimization problem of minimizing the loss function, and parameters in linear regression are solved based on the closed form solution method and the stochastic gradient descent method respectively.**

I. INTRODUCTION

This paper aims to attain the solution of parameters in linear regression. Linear regression is the simplest and most used method in regression analysis and prediction, so it is important us to understand and study it further.

II. METHODS AND THEORY

1.1 Linear regression

Linear regression is one of the regression problems. Linear regression assumes that the target value is linearly related to the features, that is, it satisfies a multiple linear equation. By constructing the loss function, the minimum parameters w and b of the loss function are solved. In general, we can express it as follows:

$$f(\boldsymbol{x};\boldsymbol{w},b)=\boldsymbol{w}^T\boldsymbol{x}+b$$

2.2 Loss function

The least square loss is used in this paper to measure the performance for regression：

$$L_D(\boldsymbol{w},\ b)=1/n\sum_{i=1}^{n}(y_i-f(\boldsymbol{x};\boldsymbol{w},b))^2$$

And the task is the find the value of W and b when l is minimized, so the optimization formula of the core objective is：

$$\boldsymbol{w}^*,\ b^*=argmin_{w,\ b}L_D(\boldsymbol{w},\ b)$$

2.3 Further optimization

In order to simplify the model, we add a regular term to the optimization function with the weights of penalty factor C, and the optimization problem becomes:

$$min_{w,b}J(\boldsymbol{w},b)=C\frac{||\boldsymbol{w}||^2}{2}+1/n\sum_{i=1}^{n}(y_i-f(\boldsymbol{x};\boldsymbol{w},b))^2$$

2.4 Solution of parameter

The paper proposed two methods to realize the solution of linear regression parameters: the closed-form solution and Stochastic Gradient Descent (SGD).

In order to simplify our proof, we introduce augmented matrix and augmented vectors still represent $\boldsymbol{w}$ and **X**.

$$\boldsymbol{X}=(\boldsymbol{x_1},\boldsymbol{x_2},\dots,\ \boldsymbol{x_n})^T$$

$$\boldsymbol{w}=(\boldsymbol{b},\boldsymbol{w_1},\dots,\ \boldsymbol{w_n})^T$$

Then the loss function and optimization becomes:

$$L_D(\boldsymbol{w})=\frac{1}{2}\left|\left|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\right|\right|^2$$

$$J_D(\boldsymbol{w})=\frac{||\boldsymbol{w}||^2}{2}+\frac{1}{2}||\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}||^2$$

The closed-form solution directly derives the loss function and let the derivatives equals 0 to find the Extreme point :

let ($\boldsymbol{a}=\boldsymbol{y}-\boldsymbol{x}\boldsymbol{w}$), then

$$\frac{\partial L_D(\boldsymbol{w})}{\partial w}=\frac{\partial \boldsymbol{a}}{\partial \boldsymbol{w}}\frac{\partial\left(\frac{\boldsymbol{1}}{\boldsymbol{2}}\boldsymbol{a}^T\boldsymbol{a}\right)}{\partial \boldsymbol{a}}$$

$$=\frac{\boldsymbol{1}}{\boldsymbol{2}}\frac{\partial \boldsymbol{a}}{\partial \boldsymbol{w}}2\boldsymbol{a}$$

$$=\frac{\partial(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})}{\partial \boldsymbol{w}}\boldsymbol{y}-\boldsymbol{X}=0$$

Assuming $\left|\left|\boldsymbol{X}^T\boldsymbol{X}\right|\right|\neq 0$,then we get:

$$\boldsymbol{w}=(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

Another way is SGD. The core content of gradient descent is to update the independent variables constantly (seeking partial derivatives for **W** and **b**), so that the objective function is constantly approaching the minimum value.

In this paper, we use $d=-\frac{\partial J_D(\boldsymbol{w})}{\partial w}=C\boldsymbol{w}+\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}$ as the direction of optimization and update parameters with learning rate η:

$$\boldsymbol{w}\text{'}=\boldsymbol{w}-\eta\frac{\partial J_D(\boldsymbol{w})}{\partial w}$$

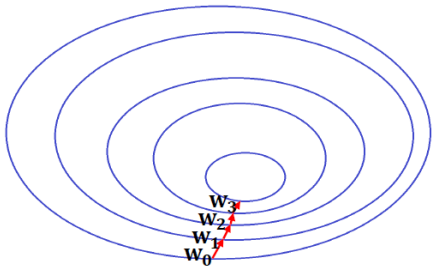

Figure. 1. Sketch map of SGD

Gradient descent uses all samples to calculate the direction, however, information is redundant amongst samples and it presents slow convergence for large data set. So SGD is used in this paper, which randomly select an example in the training set to calculate the direction and we can therefore get to a reasonable solution quickly.

Algorithm: SGD	
1	Initialize parameter **w** and learning rate **η**
2	**While** stopping condition is not achieve **do**
3	Randomly select an example i in the
4	training set
5	$\mathbf{w} = \mathbf{w} - \eta \frac{\partial J_D(\boldsymbol{w})}{\partial w}$
6	**end**

III. EXPERIMENT

A. Dataset

Linear Regression uses Housing in LIBSVM Data, including 506 samples and each sample has 13 features.

And then it is split into training set and validation set, which comprises 75% and 25% of the raw data, respectively.

B. Implementation

The initialization of the experiment is as follows:

TABLE I
INITIALIZATION OF PARAMETERS

Learning rate	η = 0.005
Max epoch number	e = 100
Penalty factor	C = 0.5

And all parameter of w is initialized randomly.

Using the closed form solution, we get the train loss of 11.2248 and validation loss of 10.6453.

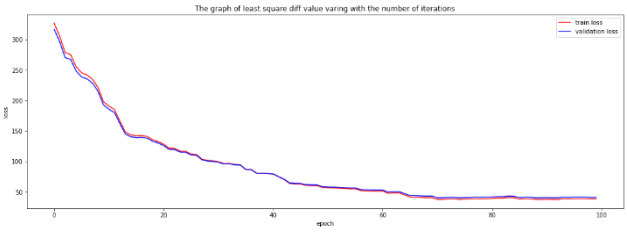

Figure. 2. The graph of least square diff value varying with the number of iterations using SGD

Using SGD, after 100 iterations, the train loss and validation loss drop to 38.1249 and 39.1076, which can be seen from the diagram above:

IV. CONCLUSION

In this experiment, the parameters of linear regression model are solved by two ways. In the closed form solution method, the loss of verification set is 10.6453, while the loss of random gradient descent is 39.1076, which indicates that the fitting degree of the model is high, but meanwhile the loss of random gradient is slightly greater than that of the closed form solution, because the calculated result is not an accurate gradient. For optimization problems, although the loss function obtained by each iteration is not towards the global optimal direction, but the direction of the large whole is towards the global optimal solution, and the final result is often near the global optimal solution.