华南理工大学

South China University of Technology

The Experiment Report of *Machine Learning*

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Chen Han

Supervisor:
Mingkui Tan

Student ID:
201936380086

Grade:
Class 1, Grade 2019

October 7, 2021

Linear Regression and Stochastic Gradient Descent

***Abstract*—Further understand of linear regression closed-form solution and Stochastic gradient descent with experiments.**

I. Introduction

In order to further understand the conception of linear regression and stochastic gradient descent, we calculate the closed-form solution of linear regression and act stochastic gradient descent on a small dataset, conducted optimized parameter and observed the experimental effect.

II. Methods and Theory

We use linear regression and stochastic gradient descent.

Linear regression tries to learn a linear model to predict the output as best as possible. More generally, we try to learn a model like :

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

which is called multiple linear model.

A. first part

The first part of this experiment is to obtain the closed-form solution of linear regression. There are the steps:

1. Determine the loss function of this linear regression:

$$\begin{aligned} Loss(\boldsymbol{w}) &= \frac{1}{2}\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i\boldsymbol{w})^2 \\ &= \frac{1}{2}\begin{bmatrix} y_1 - \boldsymbol{x}_1\boldsymbol{w} \\ \vdots \\ y_n - \boldsymbol{x}_n\boldsymbol{w} \end{bmatrix}^T \begin{bmatrix} y_1 - \boldsymbol{x}_1\boldsymbol{w} \\ \vdots \\ y_n - \boldsymbol{x}_n\boldsymbol{w} \end{bmatrix} \\ &= \frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) \end{aligned}$$

where

$$\begin{aligned} \boldsymbol{x}_i &= (x_{11}, x_{12}, \cdots, x_{1m}, 1)\boldsymbol{w} \\ &= (w_1, w_2, \cdots, w_m, b)^T\boldsymbol{X} \\ &= (\boldsymbol{x_1}, \cdots, \boldsymbol{x_n})^T \end{aligned}$$

2. Take the derivative of the loss function:

$$\boldsymbol{a} = \boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}$$

$$\begin{aligned} \frac{\partial Loss(\boldsymbol{w})}{\partial \boldsymbol{w}} &= \frac{\partial \boldsymbol{a}}{\partial \boldsymbol{w}}\frac{\partial(\frac{1}{2}\boldsymbol{a}^T\boldsymbol{a})}{\partial \boldsymbol{a}} \\ &= \frac{1}{2}\frac{\partial \boldsymbol{a}}{\partial \boldsymbol{w}}(2\boldsymbol{a}) \\ &= \frac{\partial(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})}{\partial \boldsymbol{w}}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) \\ &= -\boldsymbol{X^T}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) \end{aligned}$$

3. Since Loss($\boldsymbol{w}$) is a convex function, we can get $\boldsymbol{w^*}$ from $\frac{\partial Loss(\boldsymbol{w})}{\partial \boldsymbol{w}} = 0$

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

B. second part

The second part of this experiment is to obtain the linear regression model through the method of stochastic gradient descent.

In machine learning algorithms, it is sometimes necessary to build a loss function for the original model, and then optimize the loss functionthrough the optimization algorithm, so as to find the optimal parameters and minimize the value of the loss function. In the optimization algorithm of machine learning parameters, the algorithm based on gradient descentGDis often used.

Stochastic gradient descent (SGD) is a simple but very effective method, which is widely used for linear classifier learning under convex loss functions such as SVMs and logistic regression.

There are the steps:

1. Find the best direct D. We can use $\boldsymbol{D} = -\frac{\partial Loss(\boldsymbol{w})}{\partial \boldsymbol{w}}$ as the direction optimization.

$$\boldsymbol{D_t} = -\boldsymbol{x_i^T}(\boldsymbol{y} - \boldsymbol{x_t}\boldsymbol{w_t})$$

where $\boldsymbol{x_t}$ is a random sample in train set.

2. Find a good step size η which is called it learning rate.
3. Set $\boldsymbol{w_{t+1}} = \boldsymbol{w_t} + \eta\boldsymbol{D_t}$
4. Repeat step 1,2,3 for 5 to 8 rounds.

III. Experiments

A. Dataset

Linear Regression uses Housing in LIBSVM Data, including 506 samples and each sample has 13 features. We split the dataset by 75% for training and 25% for validation.

B. Implementation

1. Load the experiment data. You can use load_svmlight_file function in sklearn library.
2. Devide dataset by 0.25.
3. write Loss-calculating function.

```
def calc_loss(X, y, W):
    loss = 0.5*(np.linalg.norm(y-X.dot(W))**2)
    return loss
```

4. Get the closed-form solution by formula:

$$W^* = (X^T X)^{-1}X^T Y$$

```
def solve_W_closed_form(X, Y):
    Y = np.array(Y)
    W = np.linalg.inv(X.T.dot(X)).dot(
        X.T).dot(Y)
    return W
```

5. write stochastic gradient descent function.

```
def
    solve_W_stochastic_gradient_descent
    (X, Y, learning_rate, epoch, batch,
     X_test, Y_test):
    train_loss = []
    test_loss = []
    Y = Y.reshape(Y.shape[0], 1)
    W = np.random.rand(X.shape[1], 1)
    for i in range(epoch):
        bat = np.random.choice(X.shape
            [0], batch)
        X_batch = X[bat]
        Y_batch = Y[bat]
        gradient = -X_batch.T.dot(
            Y_batch) + X_batch.T.dot(
            X_batch).dot(W)
        gradient = gradient * (1/batch
            )
        # normnazie
        W = W - learning_rate *
            gradient
        epoch_train_loss = calc_loss(X
            , Y, W)
        epoch_test_loss = calc_loss(
            X_test, Y_test, W)
        train_loss.append(
            epoch_train_loss)
        test_loss.append(
            epoch_test_loss)
    #print(loss)
    return W, train_loss, test_loss
```

6. Get the solution.

TABLE I
VALUE OF *loss* IN LINEAR REGRESSION

Loss value	5636.7082077152245
Loss_train value	4233.440988955715
Loss_val value	1403.2672187595103

TABLE II
VALUE OF *loss* IN STOCHASTIC GRADIENT DESCENT

Loss_train value	5049.16020020202
Loss_val value	1780.4511281009281

IV. Conclusion

In this experiment, I learnt to use linear regression and stochastic gradient descent. Also, I review coding with python and realize the advantage of LaTeX.

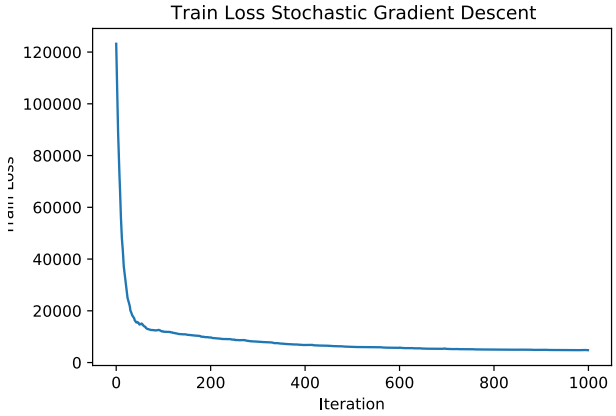

Fig. 1. Train Loss Stochastic Gradient Descent.

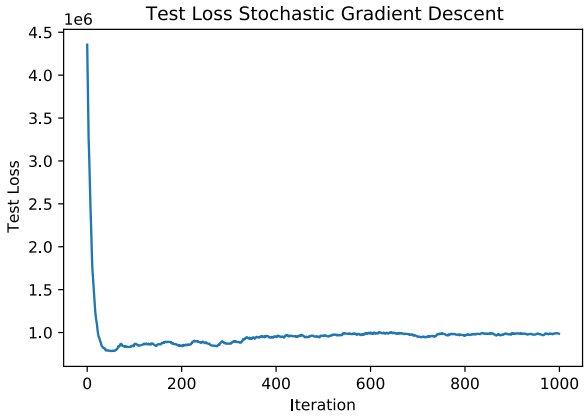

Fig. 2. validation Loss Stochastic Gradient Descent.