華南理工大學

South China University of Technology

The Experiment Report of *Machine Learning*

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Lizhao Liu

Supervisor:
Mingkui Tan

Student ID:
201730683109

Grade:
Undergraduate

November 9, 2019

Linear Regression, Ridge Regression and Gradient Descent

***Abstract*—In this report, we solve the house price prediction problem in Housing Dataset by using Linear Regression, Ridge Regression by closed form solution, and even Linear Regression optimized by using Gradient Descent and its variants e.g. Stochastic Gradient Descent and Mini Batch Gradient Descent. We perform experiments on four aspects:**
1. Comparing the magnitude of W between linear regression and ridge regression optimized by closed form solution.
2. Comparision between Gradient Descent and its variants in terms of convergence and time-cost.
3. Tuning learning rate of Mini-Batch Gradient Descent.
4. Comparing the magnitude of W between linear regression optimized by closed form solution, ridge regression and linear regression optimized by mini-batch Gradient Descent.

I. Introduction

LINEAR Regression is the core of machine learning. Based on Linear Regression, many machine learning algorithm, such as Ridge Regression, Logistics Regression, SVM are developed. Even in many deep learning model, such as Covolutional Neural Network(CNN), Long Short Term Memory(LSTM), Gated Recurrent Unit(GRU), linear regression(or Linear Layer) is the basic component. So it is very important to fully exploit the insight of Linear Regression. To achieve that goal, In this experiment, we conduct many experiments on linear regression optimized by closed form solution and gradient descent, ridge regression by solving the house price predicting problem.

II. Methods and Theory

In this part, we first define the Linear Regression, Mean Squared Error(MES) Loss function. Then we solve the closed form solution of Linear Regression and its variant Ridge Regression. At last we define gradient descent algorithm, and its variants schocastic gradient descent and mini batch gradient descent.

A. Linear Regression

Given dataset $D = \{x_i, y_i\}_{i=1}^m$, where $x_i \in \mathbf{R}^n$ and $y_i \in \mathbf{R}$. Liner Regression Model is parameterized by (W, b), where $W \in \mathbf{R}^n$ and $b \in \mathbf{R}$ and the form of it is:

$$y_i = x_i^T W + b \tag{1}$$

We can write it in vertorized form:

$$y = X^T W \tag{2}$$

where

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}, \quad X = \begin{pmatrix} 1 & \mathbf{x}_1^\mathsf{T} \\ 1 & \mathbf{x}_2^\mathsf{T} \\ \vdots & \vdots \\ 1 & \mathbf{x}_m^\mathsf{T} \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1n} \\ 1 & x_{21} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{m1} & \cdots & x_{mn} \end{pmatrix},$$

$$\boldsymbol{W} = \begin{pmatrix} b \\ W_1 \\ W_2 \\ \vdots \\ W_n \end{pmatrix}.$$

B. Mean Squared Error

Given the prediction of Linear Model $\hat{y} = X^T W$ and the Ground truth y, the MSE loss function is:

$$L(\hat{y}, y) = \frac{1}{2m}\Sigma_{i=1}^m(\hat{y}_i, y_i) = \frac{1}{2m}(\hat{y} - y)^T(\hat{y} - y) \tag{3}$$

C. Closed form Solution

The best parameter W^* can be obtained by

$$W^* = \arg\min_W L(X^T W, y) \tag{4}$$

So, we set

$$\frac{\partial L(\hat{y}, y)}{\partial W} = 0 \tag{5}$$

We can get

$$W^* = (X^T X)^{-1} X^T y \tag{6}$$

D. Ridge Regression

In equation 6, when $m > n$, $X^T X$ is not invertible. Ridge Regression tackle this problem by adding a small term. So the closed form solution for Ridge Regression is

$$W^*_{ridge} = (X^T X + \lambda \mathbf{I})^{-1} X^T y \tag{7}$$

In this way, the loss function for Ridge Regression is

$$L_{ridge}(\hat{y}, y) = \frac{1}{2m}(\hat{y} - y)^T(\hat{y} - y) + \frac{1}{2}W^T W \tag{8}$$

In another point of view, Ridge Regression can be seen as a regularized form of Linear Regression.

E. Gradient Descent

Not all problem can be derived a closed form solution, which is often the case in machine learning. So instead of getting the best parameters W^* immedeatly, we can update it incremently until W gets closed enough to W^*. That is the basic idea of Gradient Descent Algorithm. Gradient Descent Algorithm can be found in Algorithm 1.

Algorithm 1: Gradient Descent

```
Input: Training Dataset $D = \{x_i, y_i\}_{i=1}^m$, learning rate
       $\eta$, max iteration $N$, parameter $W$
Output: Optimized parameter $\hat{W}$
1 $t \leftarrow 1$
2 for $t < N$ do
3     $\hat{y} \leftarrow X^T W$
4     $L(\hat{y}, y) \leftarrow \frac{1}{2m}(\hat{y} - y)^T(\hat{y} - y)$
5     $\Delta W \leftarrow \frac{\partial L}{\partial W}$
6     $W \leftarrow W - \eta * \Delta W$
7 $\hat{W} = W$
8 return $\hat{W}$
```

F. Stochastic Gradient Descent

Different from gradient descent which updates after it computes the gradient of parameter over all example, stochastic gradient descent updates immediately once it computes parameter's gradient from only one example. Stochastic Gradient Descent Algorithm can be found in Algorithm 2. It can be seen in the algorithm that, there is another for loop inside the for loop of gradient descent algorithm.

Algorithm 2: Stochastic Gradient Descent

```
Input: Training Dataset $D = \{x_i, y_i\}_{i=1}^m$, learning rate
       $\eta$, max iteration $N$, parameter $W$
Output: Optimized parameter $\hat{W}$
1  $t \leftarrow 1$
2  for $t < N$ do
3      $i \leftarrow 0$
4      for $i < m$ do
5          $\hat{y}_i \leftarrow X_i^T W$
6          $L(\hat{y}_i, y_i) \leftarrow \frac{1}{2}(\hat{y}_i - y_i)^T(\hat{y}_i - y_i)$
7          $\Delta W \leftarrow \frac{\partial L}{\partial W}$
8          $W \leftarrow W - \eta * \Delta W$
9  $\hat{W} = W$
10 return $\hat{W}$
```

G. Mini Batch Gradient Descent

Due to the fact that stochastic gradient descent updates the parameter based on only one examples, which it is time-comsuming, mini batch gradient descent updates the parameter based on a batch of examples, typically the batch size is bigger than 1 and smaller than the number of whole examples, e.g. 8, 16, 32... Mini Batch Gradient Descent Algorithm can be found in Algorithm 3.

III. Experiments

A. Dataset

We conduct all the experiments on housing dataset, which has total 506 examples and each example's form is (x, y), where $x \in \mathbf{R}^{13}$ and $y \in \mathbf{R}$. We split the dataset into three

Algorithm 3: Mini Batch Gradient Descent

```
Input: Training Dataset $D = \{x_i, y_i\}_{i=1}^m$, learning rate
       $\eta$, max iteration $N$, parameter $W$
Output: Optimized parameter $\hat{W}$
1  $t \leftarrow 1$
2  for $t < N$ do
3      $i \leftarrow 0$
4      for $X_{batch}, y_{batch}$ in $D$ do
5          $\hat{y}_{batch} \leftarrow X_{batch}^T W$
6          $L(\hat{y}_{batch}, y_{batch}) \leftarrow$
             $\frac{1}{2}(\hat{y}_{batch} - y_{batch})^T(\hat{y}_{batch} - y_{batch})$
7          $\Delta W \leftarrow \frac{\partial L}{\partial W}$
8          $W \leftarrow W - \eta * \Delta W$
9  $\hat{W} = W$
10 return $\hat{W}$
```

part: 272 training examples, 67 validation examples and 167 test examples. X is already scaled between -1 and 1. And we normalized the $X_{train}, X_{validation}, X_{test}$ by using the mean and variance of X_{train}.

B. Implementation

We implement the Linear Regression, Ridge Regression and Gradien Descent Algorithm using python and mainly rely on the numpy package.

C. Linear Regression and Ridge Regression

In this section, we conduct the experiment of serveral magtitude of λ, which leads to serveral W_{ridge}. We use L_2norm to represent the magtitude of a vector. And we evaluate the performance on housing data's test set. Given $v \in \mathbf{R}^n$

$$L_2norm(v) = \sqrt{\Sigma_{i=1}^n v_i} \tag{9}$$

From Table I, we can see that:

First, as the magtitude of lambda becomes bigger, both MSE_{train} and MSE_{val} become bigger.

Second, as the magtitude of lambda becomes bigger, the magtitude of W_{ridge} become smaller.

The first observation is not often the cases. But the second observation can be explained that as the regularizer λ become bigger, the W is regularized, so that it is smaller.

By comparing Table I and Table II, we can see that, the performance and the weight magtitude is almost the same as the λ is small, but the difference become larger as λ grows.

TABLE I
Different lambdas' impact in terms of error and W magtitude.

log_{10}^{λ}	-2	-1	0	1	2	3
MSE_{train}	10.23	10.23	10.24	10.63	30.37	181.35
MSE_{val}	17.08	17.09	17.16	18.20	43.92	214.55
$L_2norm(W_{ridge})$	0.83	0.82	0.81	0.72	0.50	0.38

TABLE II
LINEAR REGRESSION ERROR AND W MAGTITUDE.

MSE_{train}	10.23
MSE_{val}	17.08
$L_2norm(W)$	0.83

D. Gradient Descent, Stochastic Gradient Descent and Mini Batch Gradient Descent

In this section, we conduct experiments on optimizing the linear regression model by using three gradient descent algorrithms, and compare them in terms of convergence and time cost.

IV. CONCLUSION

This section summarizes the paper. In our experiments, you can also write your gains and inspirations in here.