Project1 Report

Xueying Sui 1001682442

Problem

We need to train a linear regression model, and use this model to make prediction. We have a data set, this data set contains 150 data items. We should divide it into two parts: the training set and the test set.

We use the training data train our model, then we can get a set of parameters of estimate function. We use this estimate function and test data to make prediction and calculate accuracy.

Data

We have 150 data items, these items have three tags: Iris-setosa, Iris-versicolor and Iris-virginica. Each of the tags corresponds to 50 data items. To train linear regression model, we should assign values to these tags, like (0, 1, 2) or (100, 101, 102), etc. When we dividing the data set, we can try five or five separate or four or six separate.

Method 1

$$\theta = \left(A^T A\right)^{-1} A^T Y$$

We can see the input data points of training data as a matrix A, see the label data as a matrix Y, and see parameters as a matrix Θ. So the sum of the distances between the model and each point of the training data can be expressed as:

$$\begin{aligned} distance &= (A\theta - Y)^T (A\theta - Y) \\ &= \left(\theta^T A^T - Y^T\right)(A\theta - Y) \\ &= \theta^T A^T A\theta - \theta^T A^T Y - Y^T A\theta + Y^T Y \\ &= \theta^T A^T A\theta - 2Y^T A\theta + Y^T Y \end{aligned}$$

Now our goal is to get the minimum value of distance:

$$\frac{\partial distance}{\partial \theta} = 2\theta^T A^T A - 2Y^T A$$

We make the result of above function equals 0, we can get:

$$\theta^T A^T A = Y^T A \Rightarrow A^T A\theta = A^T Y \Rightarrow \left(A^T A\right)^{-1} A^T A\theta = \left(A^T A\right)^{-1} A^T Y$$

$$\Rightarrow \theta = \left(A^T A\right)^{-1} A^T Y$$

Now we can directly use this function to calculate parameters. And we can use these parameters get estimate function, like this:

$$y = \theta_0 + \theta_1 * x_1 + \theta_2 * x_2 + \theta_3 * x_3 + \theta_4 * x_4$$

For more convenient matrix calculation, we can make x0=1, and the estimate function like this:

$$y = \theta_0 * x_0 + \theta_1 * x_1 + \theta_2 * x_2 + \theta_3 * x_3 + \theta_4 * x_4$$

(1) We use 90 data items as training data(including 3 groups, each group has 30 data items), 60 data item as testing data(including 3 groups, each group has 20 data items).

(i)We assign values to the tags. Replace Iris-setosa with 0, replace Iris-versicolor with 1 and replace Iris-virginica with 2

Exp: for one of the data items (5.1,3.5,1.4,0.2,Iris-setosa), we can write like this:

$$\theta_0 * 1 + \theta_1 * 5.1 + \theta_2 * 3.5 + \theta_3 * 1.4 + \theta_4 * 0.2 = 0$$

for one of the data items (7.0,3.2,4.7,1.4,Iris-versicolor), we can write like this:

$$\theta_0 * 1 + \theta_1 * 7.0 + \theta_2 * 3.2 + \theta_3 * 4.7 + \theta_4 * 1.4 = 1$$

for one of the data items (6.3,3.3,6.0,2.5,Iris-virginica), we can write like this:

$$\theta_0 * 1 + \theta_1 * 6.3 + \theta_2 * 3.3 + \theta_3 * 6.0 + \theta_4 * 2.5 = 2$$

We use training data calculate the values of these parameters. After we get the values of parameters, we can use testing data to test the accuracy of this linear regression model.

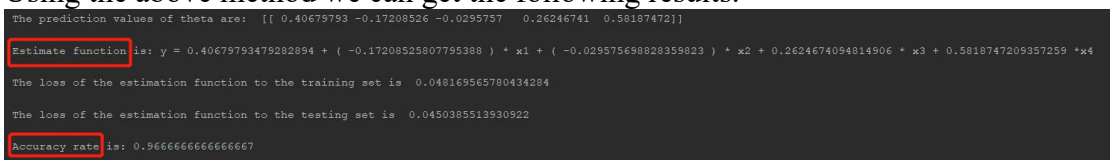
Firstly, I use MSE(Mean Squared Error) to calculate the loss of training data and estimate the loss of testing data.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Then I use testing data to calculate the accuracy, the method is to separately calculate distances between the estimated value of y and the three tag values, the tag with the smallest distance is used as the estimated tag of this data item, then compared with the real tag. If the estimated tag is the same as real tag, the prediction is correct, if not, the prediction is wrong.

Results

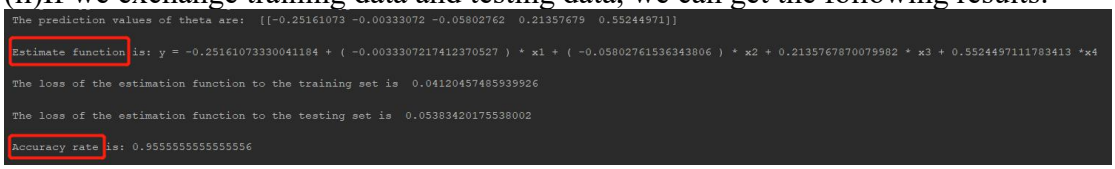
Using the above method we can get the following results.



The result of the loss for training data is about 0.048, and the result of the loss for testing data is about 0.045. These two loss results are similar.

Then I use the method mentioned above to calculate the accuracy, the result is about 96.7%.

(ii)If we exchange training data and testing data, we can get the following results:

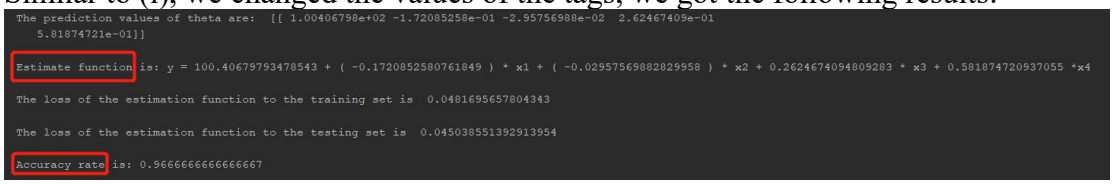


Different from before results, the result of the loss for this time's training data is about 0.041, and the result of the loss for this time's testing data is about 0.054.

Then I use the method mentioned above to calculate the accuracy, the result is about 95.6%.

(iii)We assign values to the tags, Replace Iris-setosa with 100, replace Iris-versicolor with 101 and replace Iris-virginica with 102

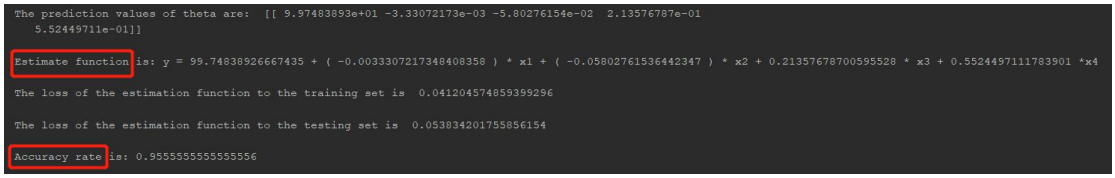
Similar to (i), we changed the values of the tags, we got the following results:



The result of the loss for training data is about 0.048, and the result of the loss for testing data is about 0.045. These two loss results are similar.

Then I use the method mentioned above to calculate the accuracy, the result is about 96.7%.

(iv)If we exchange training data and testing data, we can get the following results:



Different from (iii) results, but similar to (ii) results, the result of the loss for this time's training data is about 0.041, and the result of the loss for this time's testing data is about 0.054.
Then I use the method mentioned above to calculate the accuracy, the result is about 95.6%.

(2) We use 75 data items as training data, 75 data item as testing data(just cut from middle).
(i)We assign values to the tags. Replace Iris-setosa with 0, replace Iris-versicolor with 1 and replace Iris-virginica with 2
Exp: for one of the data items (5.1,3.5,1.4,0.2,Iris-setosa), we can write like this:

$$\theta_0 * 1 + \theta_1 * 5.1 + \theta_2 * 3.5 + \theta_3 * 1.4 + \theta_4 * 0.2 = 0$$

for one of the data items (7.0,3.2,4.7,1.4,Iris-versicolor), we can write like this:

$$\theta_0 * 1 + \theta_1 * 7.0 + \theta_2 * 3.2 + \theta_3 * 4.7 + \theta_4 * 1.4 = 1$$

for one of the data items (6.3,3.3,6.0,2.5,Iris-virginica), we can write like this:

$$\theta_0 * 1 + \theta_1 * 6.3 + \theta_2 * 3.3 + \theta_3 * 6.0 + \theta_4 * 2.5 = 2$$

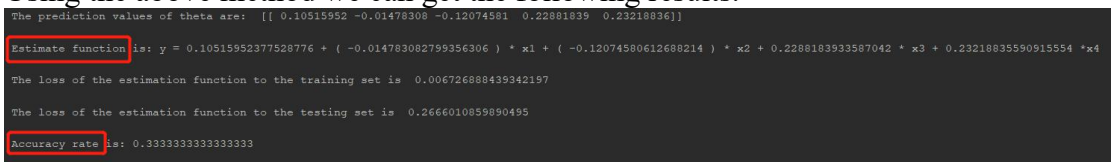
But thus time, our training data have 75 data items, including all data items that tag equals 0 and a half of data items that tag equals 1, our testing data also have 75 data items, including a half of data items that tag equals 1 and all data items that tag equals 2.
We use training data calculate the values of these parameters. After we get the values of parameters, we can use testing data to test the accuracy of this linear regression model.
Firstly, I use MSE(Mean Squared Error) to calculate the loss of training data and estimate the loss of testing data.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Then I use testing data to calculate the accuracy, the method is to separately calculate distances between the estimated value of y and the three tag values, the tag with the smallest distance is used as the estimated tag of this data item, then compared with the real tag. If the estimated tag is the same as real tag, the prediction is correct, if not, the prediction is wrong.

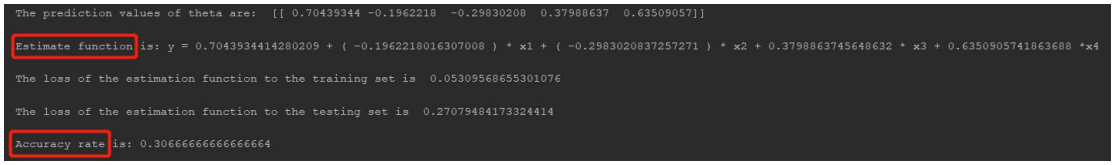
Results

Using the above method we can get the following results.



The result of the loss for training data is about 0.007, and the result of the loss for testing data is about 0.267. These two loss results are similar.
Then I use the method mentioned above to calculate the accuracy, the result is about 33.3%.

(ii)If we exchange training data and testing data, we can get the following results:



The result of the loss for this time's training data is about 0.053, and the result of the loss for this time's testing data is about 0.271.
Then I use the method mentioned above to calculate the accuracy, the result is about 30.67%.

Method 2

Gradient Descent
First we can make x0=1 and define a cost function:

$$J(\theta_0, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta \left(x^{(i)} \right) - y^{(i)} \right)^2$$

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \ldots + \theta_n x_n$$

We use the following formula to update the values of parameters:

$$\theta_i = \theta_i - \alpha \frac{\partial J}{\partial \theta_i}$$

In this formula, α is learning rate. So next we set the value of the learning rate, give parameters(Θ) a random set of initial values , and then set the number of iterations.

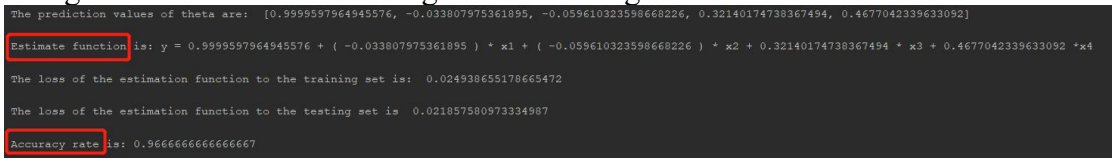
Then we need to process the data, we want most x values to be between -1 and 1, so we make standardization use the following formula:

$$x_i = \frac{x_i - \mu}{\sigma}$$

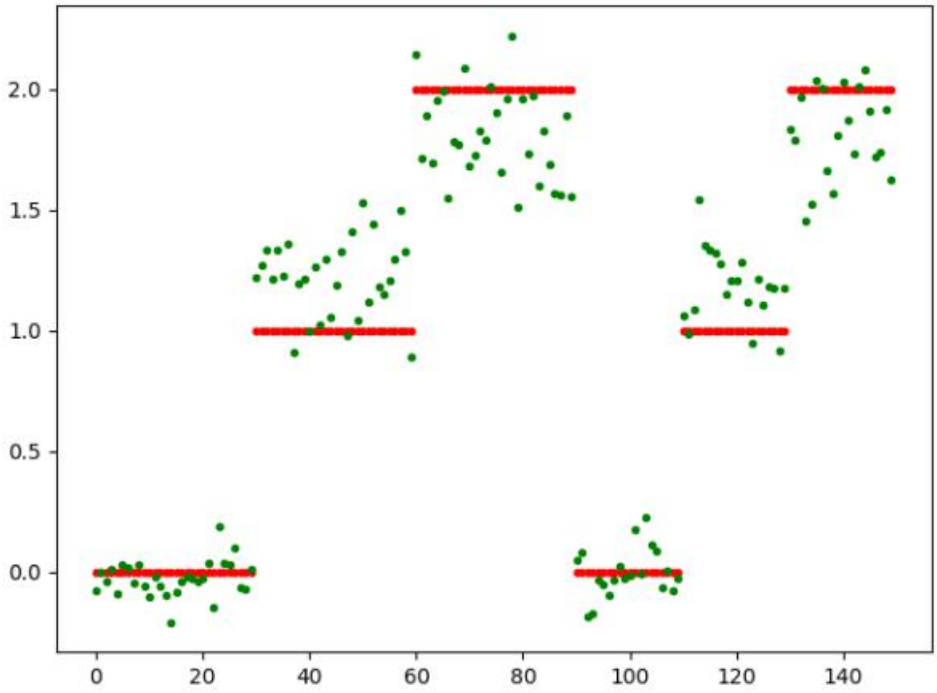
We use 90 data items as training data(including 3 groups, each group has 30 data items), 60 data item as testing data(including 3 groups, each group has 20 data items).
(i)We assign values to the tags. Replace Iris-setosa with 0, replace Iris-versicolor with 1 and replace Iris-virginica with 2
We used the above mentioned method to update the values of parameters, after we get the values of parameters, we can use testing data to test the accuracy of this linear regression model.
Firstly, I used cost function to calculate the loss of training data and estimate the loss of testing data.
Then I use testing data to calculate the accuracy, the method is to separately calculate distances between the estimated value of y and the three tag values, the tag with the smallest distance is used as the estimated tag of this data item, then compared with the real tag. If the estimated tag is the same as real tag, the prediction is correct, if not, the prediction is wrong.

Results

Using the above method we can get the following results.



The result of the loss for training data is about 0.025, and the result of the loss for testing data is about 0.022. These two loss results are similar.
Then I use the method mentioned above to calculate the accuracy, the result is about 96.7%.
This is the distribution of the tags:



The three on the left are the results of the training set, and the three on the right are the results of the test set.We can see that the distribution of the fitting results is evenly distributed.
(ii)If we exchange training data and testing data, we can get the following results:

```
The prediction values of theta are:  [0.9995597964945 57, 0.07795772771460194, -0.0499987310159661, 0.32616004426867373, 0.38978022150264713]
Estimate function is: y = 0.9995597964945 57 + ( 0.07795772771460194 ) * x1 + ( -0.0499987310159661 ) * x2 + 0.32616004426867373 * x3 + 0.38978022150264713 *x4
The loss of the estimation function to the training set is:  0.02119064510757971
The loss of the estimation function to the testing set is  0.029303708816517806
Accuracy rate is: 0.9666666666666667
```

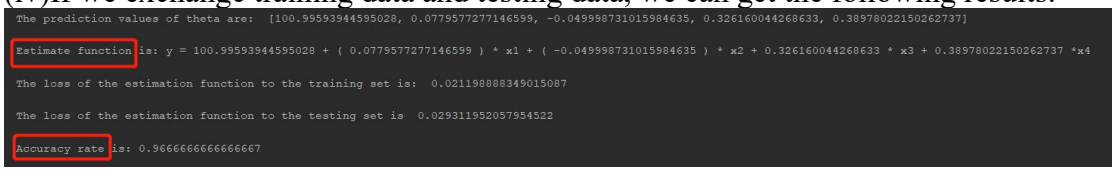
The result of the loss for this time's training data is about 0.021, and the result of the loss for this time's testing data is about 0.029.
Then I use the method mentioned above to calculate the accuracy, the result is about 96.7%, the accuracy rate is same as (i).

(iii)We assign values to the tags, Replace Iris-setosa with 100, replace Iris-versicolor with 101 and replace Iris-virginica with 102
Similar to (i), we changed the values of the tags, we got the following results:

```
The prediction values of theta are:  [100.99593944595028, -0.033807975361879544, -0.05961032359867521, 0.3214017473836348, 0.46770423396333244]
Estimate function is: y = 100.99593944595028 + ( -0.033807975361879544 ) * x1 + ( -0.05961032359867521 ) * x2 + 0.3214017473836348 * x3 + 0.46770423396333244 *x4
The loss of the estimation function to the training set is:  0.024946898420099957
The loss of the estimation function to the testing set is  0.021865824214770218
Accuracy rate is: 0.9666666666666667
```

The result of the loss for training data is about 0.025, and the result of the loss for testing data is about 0.022. These two loss results are similar.
Then I use the method mentioned above to calculate the accuracy, the result is about 96.7%.

(iv)If we exchange training data and testing data, we can get the following results:

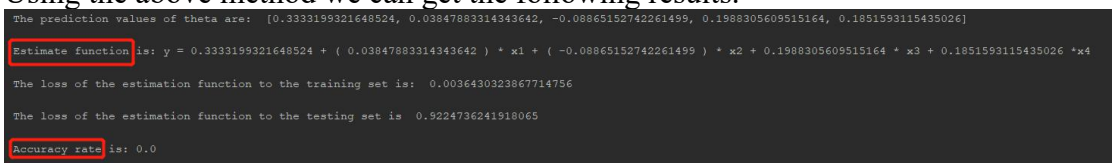


Different from (iii) results, but similar to (ii) results, the result of the loss for this time's training data is about 0.021, and the result of the loss for this time's testing data is about 0.029.
Then I use the method mentioned above to calculate the accuracy, the result is about 96.7%.

(2)We use 75 data items as training data, 75 data item as testing data(just cut from middle).
(i)We assign values to the tags. Replace Iris-setosa with 0, replace Iris-versicolor with 1 and replace Iris-virginica with 2
We used the above mentioned method to update the values of parameters, after we get the values of parameters, we can use testing data to test the accuracy of this linear regression model.
Firstly, I used cost function to calculate the loss of training data and estimate the loss of testing data.
Then I use testing data to calculate the accuracy, the method is to separately calculate distances between the estimated value of y and the three tag values, the tag with the smallest distance is used as the estimated tag of this data item, then compared with the real tag. If the estimated tag is the same as real tag, the prediction is correct, if not, the prediction is wrong.

Results

Using the above method we can get the following results.



The result of the loss for training data is about 0.004, and the result of the loss for testing data is about 0.922. These two loss results are so different.
Then I use the method mentioned above to calculate the accuracy, the result is about 0.0%.

(ii)If we exchange training data and testing data, we can get the following results:

The result of the loss for this time's training data is about 0.027, and the result of the loss for this time's testing data is about 0.898. These two loss results are so different.
Then I use the method mentioned above to calculate the accuracy, the result is about 0.0%.

Summary

Now we know the ways to train linear regression model. If we select use the matrix function, we can get the parameters directly, but if we have many features, the

computational cost of this method will be complicated and only applies to linear models.
If we select use gradient descent, we must select learning rate and steps, and we need to iterate continuously to update the values of the parameters.
We can also know that our training set must be comprehensive enough to make accurate predictions.