

**The Experiment Report of**

***Machine Learning***

**College Software College**

**Subject Software Engineering**

**Members**  陈文慧

**Student ID 201530611272**

**E-mail 1223807024@qq.com**

**Tutor**  谭明奎

**Date submitted** **2017. 12 . 8**

1. **Topic:**

**Linear regression, Linear classification and Dradient descent**

## **2. Time:** 2017.12.02

**3. Reporter:陈文慧**

**4. Purposes:**

1.Further understand of linear regression and gradient descent.

2.Conduct some experiments under small scale dataset.

3.Realize the process of optimization and adjusting parameters.

**5. Data sets and data analysis:**

Linear Regression uses Housing in LIBSVM Data, including 506 samples and each sample has 13 features. You are expected to download scaled edition. After downloading, you are supposed to divide it into training set, validation set.

Linear classification uses australian in LIBSVM Data, including 690 samples and each sample has 14 features. You are expected to download scaled edition. After downloading, you are supposed to divide it into training set, validation set.

1. **Experimental steps:**

**Linear Regression and Gradient Descent**

1.Load the experiment data. You can use load\_svmlight\_file function in sklearn library.

2.Devide dataset. You should divide dataset into training set and validation set using train\_test\_split function. Test set is not required in this experiment.

3.Initialize linear model parameters. You can choose to set all parameter into zero, initialize it randomly or with normal distribution.

4.Choose loss function and derivation: Find more detail in PPT.

5.Calculate gradient toward loss function from all samples.

6.Denote the opposite direction of gradient as .

7.Update model: . is learning rate, a hyper-parameter that we can adjust.

8.Get the loss under the training set and by validating under validation set.

9.Repeate step 5 to 8 for several times, and drawing graph of as well as with the number of iterations.

**Linear Classification and Gradient Descent**

1.Load the experiment data.

2.Divide dataset into training set and validation set.

3.Initialize SVM model parameters. You can choose to set all parameter into zero, initialize it randomly or with normal distribution.

4.Choose loss function and derivation: Find more detail in PPT.

5.Calculate gradient toward loss function from all samples.

6.Denote the opposite direction of gradient as .

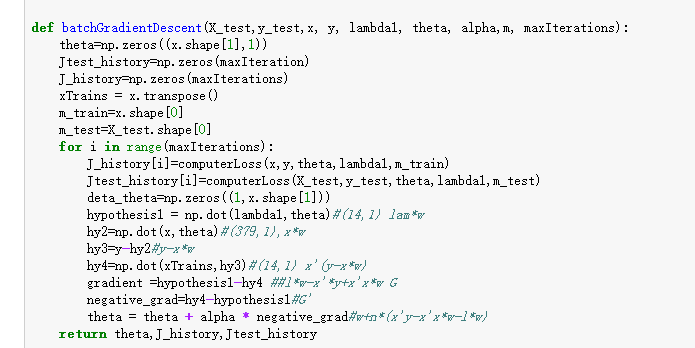
7.Update model: .  is learning rate, a hyper-parameter that we can adjust.

8.Select the appropriate threshold, mark the sample whose predict scores greater than the threshold as positive, on the contrary as negative. Get the loss under the trainin set and by validating under validation set.

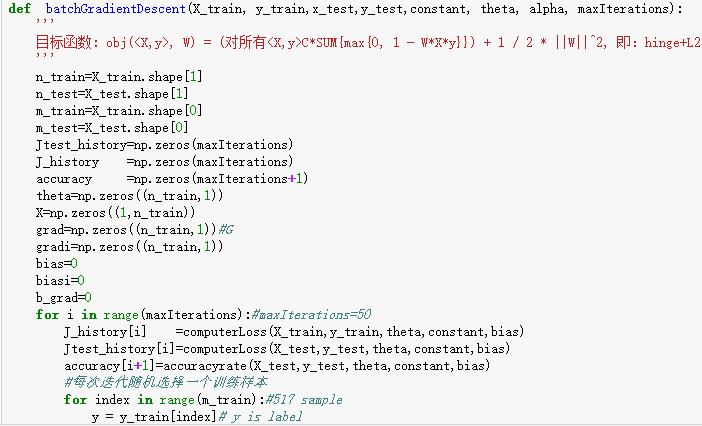
9.Repeate step 5 to 8 for several times, and drawing graph of as well as with the number of iterations.

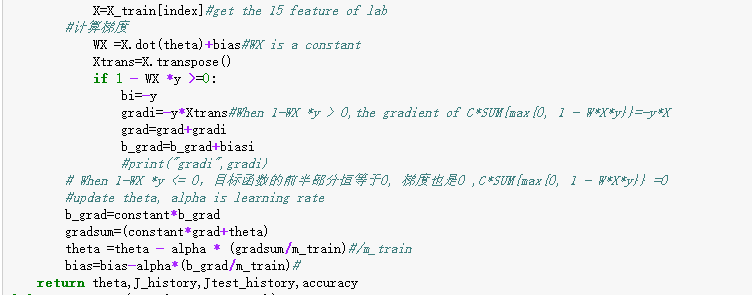
**7. Code:**

**Linear Regression and Gradient Descent**



**Linear Classification and Gradient Descent**





(Fill in the contents of 8-12 respectively for linear regression and linear classification)

1. **Selection of validation (hold-out, cross-validation, k-folds cross-validation, etc.):**

The cross validation method is adopted in both linear and linear classifications.

1. **The initialization method of model parameters:**

Linear regression and gradient descent: all zero initialization

Linear classification and gradient descent: all zero initialization

1. **The selected loss function and its derivatives:**

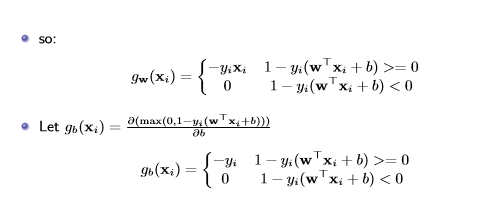
**Linear regression and gradient descent:**





**Linear classification and gradient descent:**





1. **Experimental results and curve:**

**Linear regression and gradient descent:**

## Hyper-parameter selection (η, epoch, etc.):

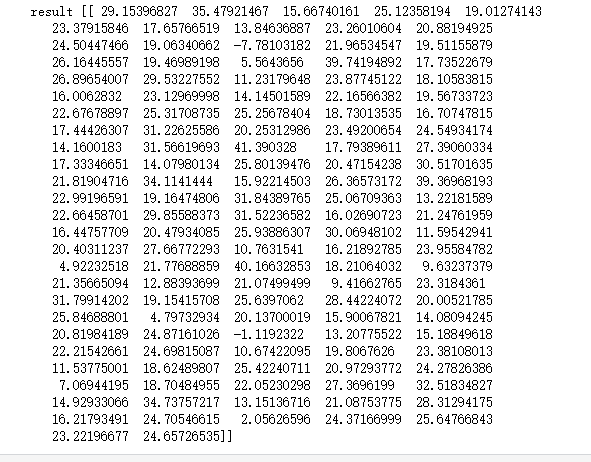
1. lambda=3,learning rate=0.0001
2. lambda=1,learning rate=0.0001
3. lambda=1,learning rate=0.00001
4. lambda=0.5,learning rate=0.0001
5. lambda=0.1,learning rate=0.0001

## lambda=0.01,learning rate=0.01

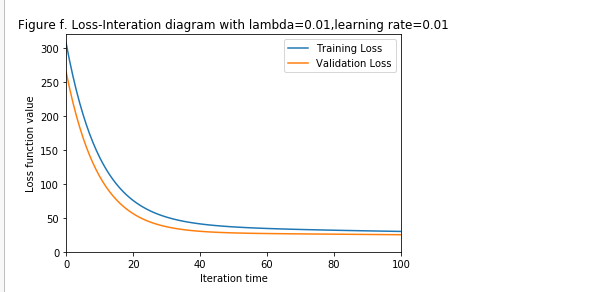
## Assessment Results (based on selected validation):



## Predicted Results (Best Results):



## Loss curve:

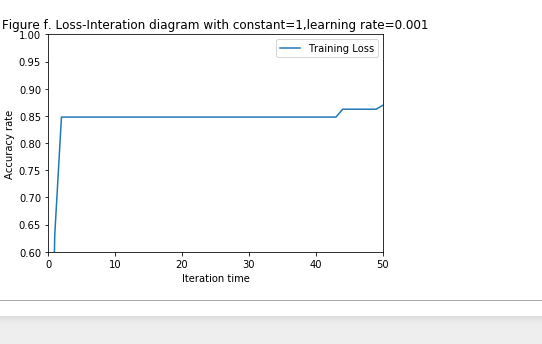


**Linear classification and gradient descent:**

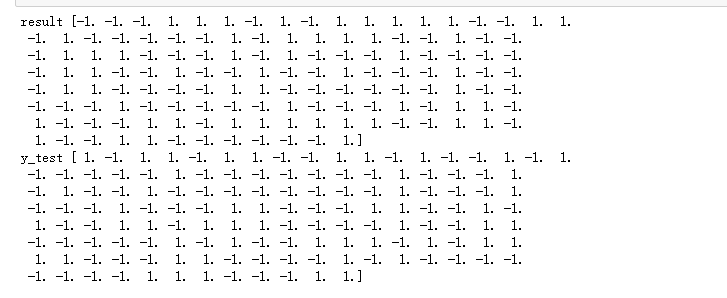
## Hyper-parameter selection (η, epoch, etc.):

## constant=0.5,learning rate=0.00001

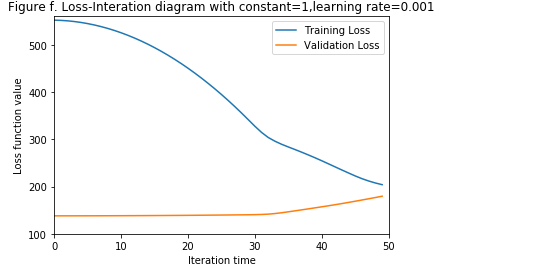
## Assessment Results (based on selected validation):



## Predicted Results (Best Results):



## Loss curve:



1. **Results analysis:**

Linear regression and gradient descent: the experimental results basically meet the expectations, and the adjustment parameters have achieved good results.

Linear classification and gradient descent: experimental results are not consistent with expectations, and convergence is achieved in the first step. Therefore, the results obtained by adjusting parameters are not ideal, and need further optimization.

1. **Similarities and differences between linear regression and linear classification:**

The same point: there are loss functions, model, and data, all of which need to be adjusted to get the best results.

Similarities and differences: linear regression is relatively simple, linear classification needs to be divided into positive and negative classes, it is possible that the threshold is not good results will result in bad results.

1. **Summary:**

This experiment makes a linear regression and linear classification of two experiments, using a gradient descent method to achieve linear regression is easy, the results are very good, linear classification is more difficult, the results are not satisfactory, at the beginning of the derivative for the wrong, the result is illogical, after a some of the debugging search, to correct the problem, but the correct result was not expected, looking for the wrong place, or to initialize the parameters of the loss function, the derivative are checked, but still can not find an error. Further improvement is needed.