

**The Experiment Report of**

***Machine Learning***

**College Software College**

**Subject Software Engineering**

**Members**   **Zhiwei Chen**

**Student ID 201530611340**

**E-mail 530108178@qq.com**

**Tutor**   **Mr.Tan**

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

1. **Topic:**

Linear Regression, Linear Classification and Gradient Descent

1. **Time:** 2017/12/02 —— 2017/12/07
2. **Reporter:**Zhiwei Chen

**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:**

（1）Linear Regression:

Source: UCI / **Housing** (Boston)

# of data: 506

# of features: 13

Link:https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html#housing

（2）Linear Classification:

Source: Statlog / **Australian**

# of classes: 2

# of data: 690

# of features: 14

Link:https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary/australian\_scale

1. **Experimental steps:**

**Linear Regression and Gradient Descent**

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

② Divide 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 G toward loss function from all samples.

⑥ Denote the opposite direction of gradient G as D.

⑦ Update model: Wt=Wt−1+ηD. η is learning rate, a hyper-parameter that we can adjust.

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

⑨ Repeat step 5 to 8 for several times, and drawing graph of Ltrain as well as Lvalidation 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.

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

⑤ Calculate gradient G toward loss function from all samples.

1. Denote the opposite direction of gradient G as D.

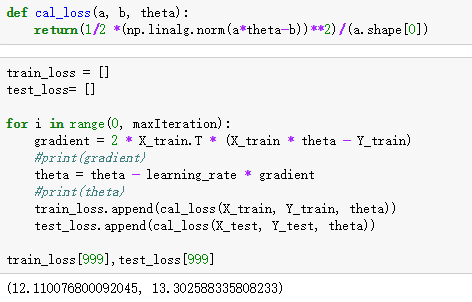
⑦ Update model: Wt=Wt−1+ηD. η is learning rate, a hyper-parameter that we can adjust.

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 Ltrain under the train set and Lvalidation by validating under validation set.

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

1. **Code:**

Linear Regression and Gradient Descent

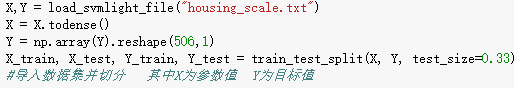


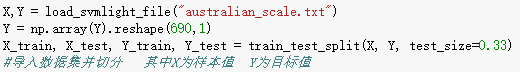
Linear Classification and Gradient Descent



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

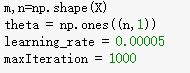
Hold-out is selected to use in both experiment.



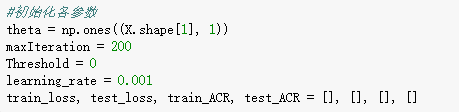


1. **The initialization method of model parameters:**

Linear Regression and Gradient Descent



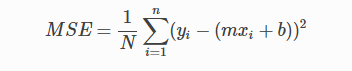
Linear Classification and Gradient Descent



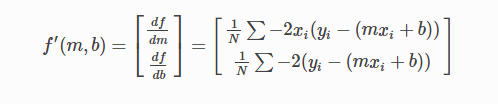
1. **The selected loss function and its derivatives:**

Linear Regression and Gradient Descent

Loss function:

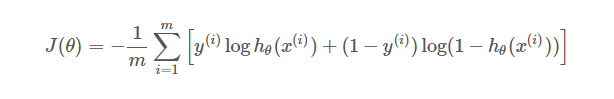


It’s derivatives:

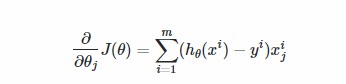


Linear Classification and Gradient Descent

Loss function:



It’s derivatives:



**11. Experimental results and curve:**

**Linear Regression and Gradient Descent**

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

η = 0.00005

epoch = 1000

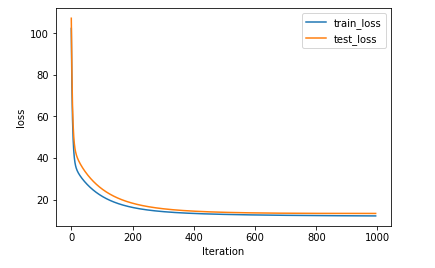
## Assessment Results (based on selected validation):

Min\_loss = 1.3026

## Predicted Results (Best Results):

Min\_loss = 12.1101

## Loss curve:



**Linear Classification and Gradient Descent**

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

η= 0.001

epoch = 200

Threshold = 0

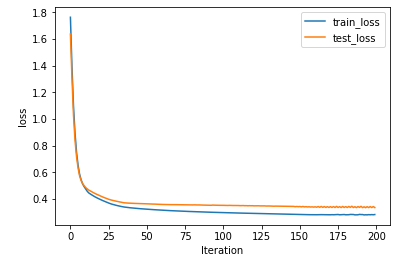
## Assessment Results (based on selected validation):

Min\_loss = 0.2157

## Predicted Results (Best Results):

Min\_loss = 0.3061

## Loss curve:



1. **Results analysis:**

The gradient descent is very smooth, which means that the descent process meets the requirements of this lab. In addition, the descent process of the loss of train and the loss of evaluation are similar, which means that the hyper-parameters chosen are suitable.

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

both use linear model

both use gradient descent

Difference:

Linear Regression is to predicted values.

Linear Classification is to predicted the classification

1. **Summary:**

It’s my first time to implement Linear Regression and Linear Classification by code.And it shows me straightly by figures in jupyter. I’m so excited that it’s due to use machine learning to solve many questions.I’ll learn it more hard.