

**The Experiment Report of**

***Machine Learning***

**College** Software College

**Subject** Software Engineering

**Members**

**Student ID** 201530612651

**E-mail** 835380624@qq.com

**Tutor**  mingkui Tan

**Date submitted** 2017.12. 2

**1. Topic:** LinearRegression& Linear Classification & Gradient Descent

**2. Time:** 2017-12-02 9:00-12:00 AM

**3. Reporter:** 任嘉宁

**4. Purposes:**

* + Further understand of linear regression and gradient descent.
  + Conduct some experiments under small scale dataset.
  + 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.
  + Linear classification uses australian in LIBSVM Data, including 690 samples and each sample has 14 features.

**6. Experimental steps:**

6.1 Linear Regression and Gradient Descent

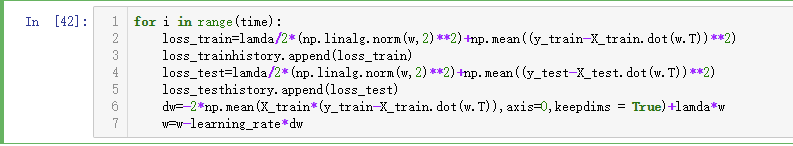
* Use load\_svmlight\_file function in sklearn library to load the experiment data.
* Use train\_test\_split function to devide dataset into training set and validation set
* Initialize linear model parameters. Set all parameter into zero, initialize it randomly or with normal distribution.
* Choose loss function and derivation
* Calculate gradient toward loss function from all samples.
* Denote the opposite direction of gradient as .
* Update model: . is learning rate, a hyper-parameter that we can adjust.
* Get the loss under the training set and by validating under validation set.
* Repeate step 5 to 8 for several times, and drawing graph of as well as with the number of iterations.

6.2 Linear Classification and Gradient Descent

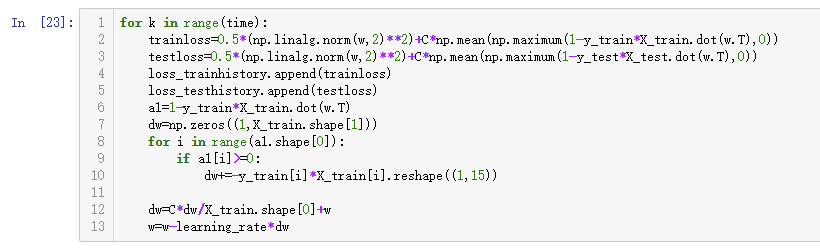
* Load the experiment data.
* Divide dataset into training set and validation set.
* Initialize SVM model parameters. Set all parameter into zero, initialize it randomly or with normal distribution.
* Choose loss function and derivation.
* Calculate gradient toward loss function from all samples.
* Denote the opposite direction of gradient as .
* Update model: . is learning rate, a hyper-parameter that we can adjust.
* 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.
* Repeate step 5 to 8 for several times, and drawing graph of as well as with the number of iterations.

**7. Code:**

Linear Regression:



Linear Classification:



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

hold-out

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

Random initialization

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

Linear Regression:

Linear Classification:

**11. Experimental results and curve:**

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

Linear Regression:

* + learning\_rate=0.001
  + lamda=1
  + time=500

Linear Classification:

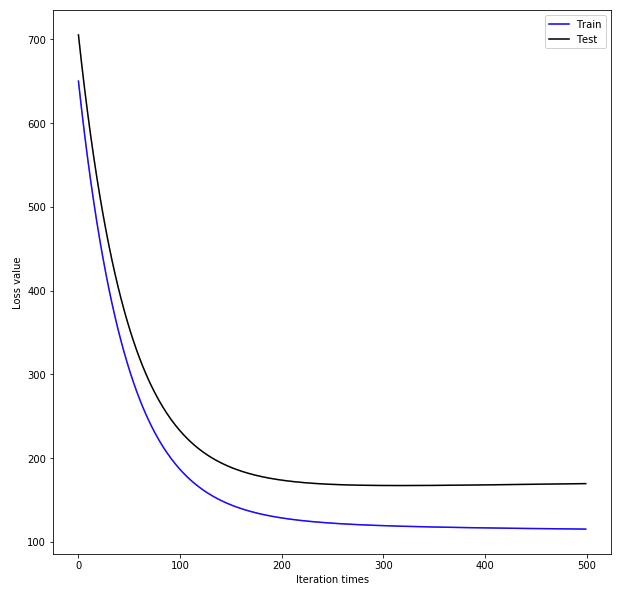
* + learning\_rate=0.005
  + time=400
  + C=5

## Assessment Results (based on selected validation):

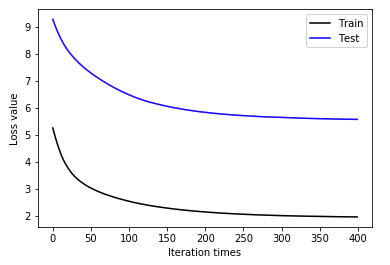
## Predicted Results (Best Results):

## Loss curve:

Linear Regression:



Linear Classification:



**12. Results analysis:**

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

**14. Summary:**