

**The Experiment Report of**

***Machine Learning***

**College Software College**

**Subject Software Engineering**

**Members**   **Zhang Huakui**

**Student ID 201530613610**

**E-mail 1033185507@qq.com**

**Tutor**   **Tan Mingkui**

**Date submitted** **2017.12.7**

1. **Topic:**

Linear Regression, Linear Classification and Gradient Descent

**2. Time: 2017.12.2**

**3. Reporter: Zhang Huakui**

**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](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html#housing) in [LIBSVM Data](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/), including 506 samples and each sample has 13 features. Linear classification uses [australian](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#australian) in [LIBSVM Data](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/), including 690 samples and each sample has 14 features.

**6. Experimental steps:**

The experimental code and drawing are completed on jupyter.

**Linear Regression and Gradient Descent**

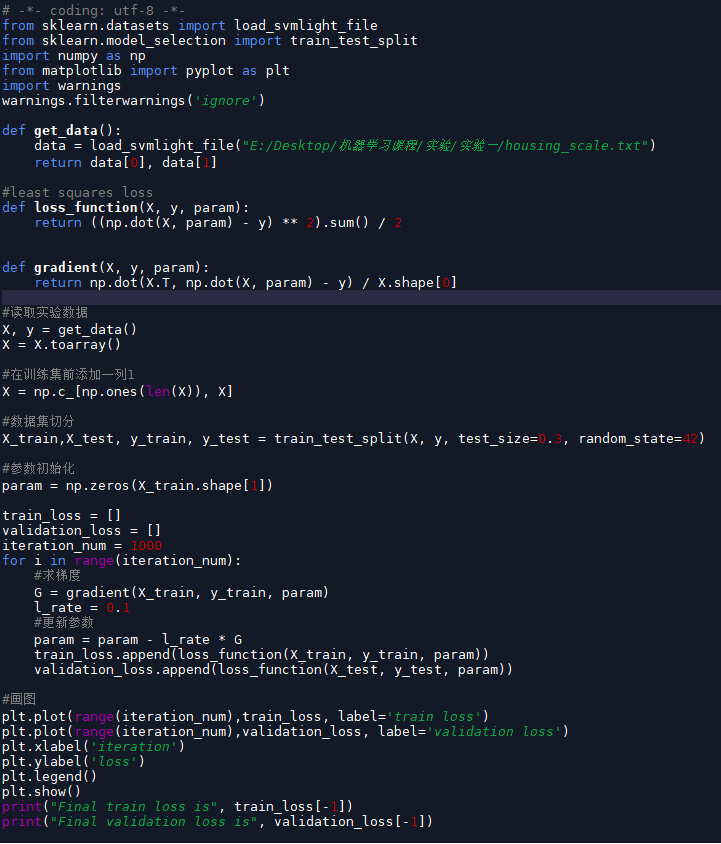
1. Load the experiment data. You can use [load\_svmlight\_file](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_svmlight_file.html) function in sklearn library.
2. Devide dataset. You should divide dataset into training set and validation set using [train\_test\_split](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html) 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: .  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. Repeat step 5 to 8 for several times, and drawing graph of   as well as  with the number of iterations.

**Linear Classification and Gradient Descent**

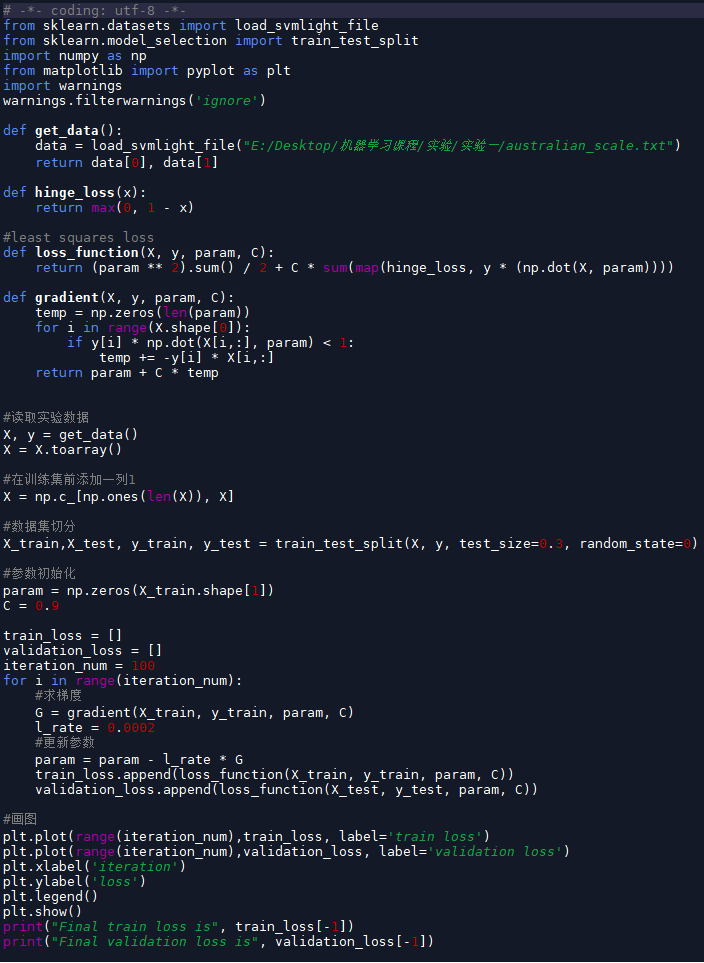
1. Load the experiment data.
2. Divide dataset into training set and validation set.
3. Initialize SVM model parameters. You can choose to set all parameter into zero, initialize it randomly or with normal distribution.
4. Choose loss function and derivation: Find more detail in PPT.
5. Calculate gradient *G* toward loss function from all samples.
6. Denote the opposite direction of gradient  *G* as *D*.
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. Repeat 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**



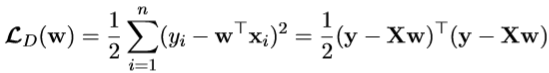
**8. Selection of validation (hold-out, cross-validation, k-folds cross-validation, etc.): hold-out**

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

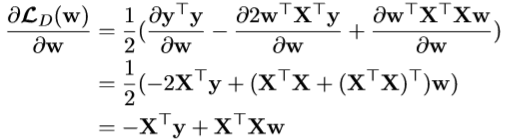
**10. The selected loss function and its derivatives:**

**Linear Regression and Gradient Descent**

Loss function:

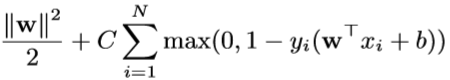


Gradient:



**Linear Classification and Gradient Descent**

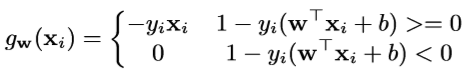
Loss function:

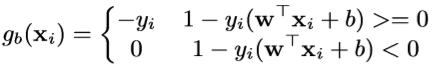


Gradient:







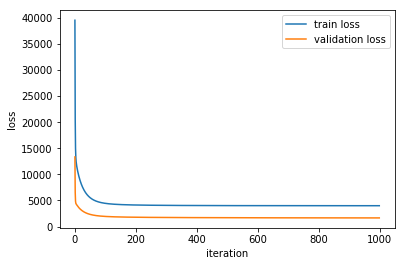


**11. Experimental results and curve:**

**Linear Regression and Gradient Descent**

## Hyper-parameter selection (η, epoch, etc.):η= 0.1

## Loss curve:

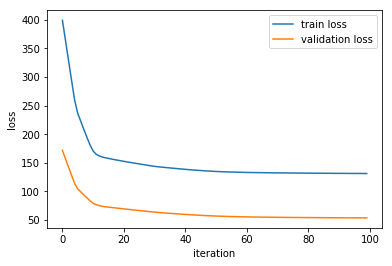


**Linear Classification and Gradient Descent**

## Hyper-parameter selection (η, epoch, etc.):η= 0.002, C = 0.9

## Assessment Results (based on selected validation): accuracy: 86%

## Loss curve:



**12. Results analysis: Both the linear regression and linear classification experiment have the good loss curves which keep descending as the iteration number grows and tend to be smooth at the end.**

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

**Similarities:**

Both linear regression and linear classification use the linear model Xw + b.

**Differences:**

Linear regression is to predict the discrete values, but linear classification is to predict continuous values.

**14. Summary:**

Through this experiment, I have further understood the linear regression, linear classification and the gradient descent. And also, it made me better realize the process of the optimization and adjusting parameters.