

**The Experiment Report of**

***Machine Learning***

**College Software College**

**Subject Software Engineering**

**Members**  赵子文(zhao ziwen)

**Student ID 201530613764**

**E-mail me@zhaoziwen.com.cn**

**Tutor**   **Tan mingkui**

**Date submitted** **2017.12.08**

**1. Topic:** Linear Regression, Linear Classification and Gradient Descent

**2. Time:** 2017/12/8

**3. Reporter:** Zhao Ziwen

**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 use 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:**

Linear Regression and Gradient Descent:

1、Load the experiment data. Use load\_svmlight\_file function in sklearn library.

2、Divide dataset. Divide dataset into training set and validation set using train\_test\_split function.

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

4、Choose loss function and derivation.

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 reat, a hyper-parameter that we can adjust.

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

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

Linear Classification and Gradient Descent

Linear Classification and Gradient Descent：

1、Load the experiment data:

2、Divide dataset into training set and validation set.

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

4、Choose loss function and derivation.

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、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 trainin set and Lvalidation by validating under validation set.

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

**7. Code:** For details, see attached.

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

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

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

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

Linear Regression ****

Linear Classification



**11. Experimental results and curve:**

For details, see attached.

## Hyper-parameter selection (η, epoch, etc.): 0.001

## Assessment Results (based on selected validation):

## Predicted Results (Best Results):

## Loss curve:

**12. Results analysis:** For details, see attached.

## 13. Similarities and differences between linear regression and linear classification: Overall, the two questions are essentially the same, that is, the fitting (matching) of the model. However, the y value (also known as label) of the classification problem is more discretized, and the same y value may correspond to a large number of x, which is of a certain range.Therefore, the classification problem is more (some x in a certain region) corresponds to (a y), and the model of regression problem is more inclined to (x in a very small region or x in general) to (y).

**14. Summary:** **T**he two basic types of machine learning problems, linear regression and classification.

How to choose a reasonable model (linear, or nonlinear (eg step function, Gaussian function)).

Create a "nice" error function (you can evaluate the fit, but also the convex function)

Take all possible techniques (such as derivative descent method, extreme value equation method) to find the best model parameters