

**The Experiment Report of**

***Machine Learning***

**College Software College**

**Subject Software Engineering**

**Members**  刘晓涛，蔡建聪，王跃

**Student ID 201530612378**

**E-mail 1416101719@qq.com**

**Tutor**   **谭明奎**

**Date submitted** **2017.12.07**

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

**2. Time: 2017.12.02**

**3. Reporter:刘晓涛**

**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 uses [Housing](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html" \l "housing" \t "https://www.zybuluo.com/chenyaofo/note/_blank) in [LIBSVM Data](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/" \t "https://www.zybuluo.com/chenyaofo/note/_blank), including 506 samples and each sample has 13 features. The data I use had been scaled.So,it is convenient to run the data without more scaling operation

2）Linear classification uses [australian](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html" \l "australian" \t "https://www.zybuluo.com/chenyaofo/note/_blank) in [LIBSVM Data](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/" \t "https://www.zybuluo.com/chenyaofo/note/_blank), including 690 samples and each sample has 14 features. The data I use had been scaled.So,it is convenient to run the data without more scaling operation

**6. Experimental steps:**

**1)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" \t "https://www.zybuluo.com/chenyaofo/note/_blank) 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" \t "https://www.zybuluo.com/chenyaofo/note/_blank) 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

（5）.Calculate gradient  G toward loss function from all samples.

（6）.Denote the opposite direction of gradient  G as D .

（7）Update model: .η 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.

**2）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  .

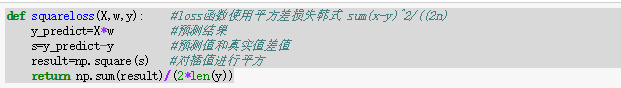
（7）.Update model:.η is learning rate, a hyper-parameter that we can adjust.

（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   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.

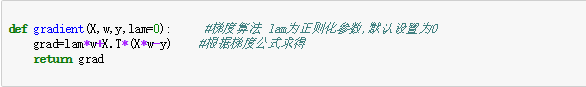
**7. Code:**

**For linear regression:**

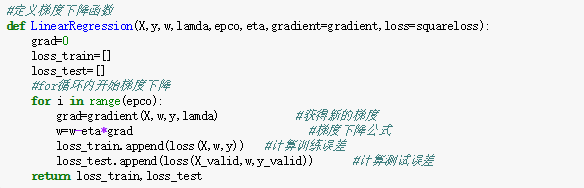
Square Loss function:



Gradient:

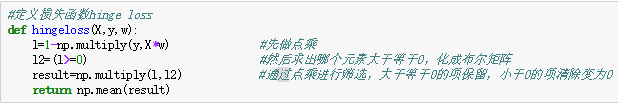


Linear Regression:

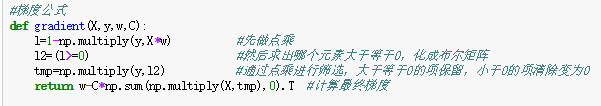


**For linear classification:**

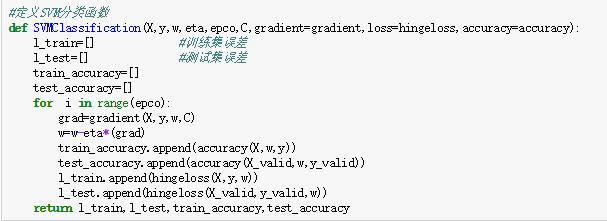
Hinge loss function:



Gradient:



Linear Classification:



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

For Linear Regression, the validation I used is Hold-Out Method. By using train\_test\_split(), I randomly split the dataset into train set and validation set with the ratio as 7:3.

For Linear Classification,the validation I used is Hold-Out Method. By using train\_test\_split(), I randomly split the dataset into train set and validation set with the ratio as 7:3.

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

For Linear Regression, I set all parameters into zero.

For Linear Classification,I set all parameters into normal distribution.

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

**For Linear Regression,my loss function is square loss function**

****

**, its derivatives is**

****

****

**For Linear Classification,my loss function is hinge loss function**

****

**, its derivatives is**

****

**Where  if ,else **

1. **Experimental results and curve:**

**For Linear Regression,**

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

η=0.001,epoch=1000

## Assessment Results (based on selected validation):

Based on hold out method,

1. If I set η=0.0001 and run the model by 1000 times,the training line and the test line have just started to separate, but the rear is immediately re-closing.The final loss of validset is 11.273617068

2. If I set η=0.0005 and run it by 1000 times,the training line and the test line have just startedly close , but the rear is gradually separate.The final loss of validset is 10.79105387

3. If I set η=0.001 and run it by 1000 times,the training line and the test line are nearly close all the running time in the iteration. The final loss of valid set is 10.7634155621

4. If I set η=0.0015 and run it by 1000 times,I find the result are nan both training sets and valid sets.If I turn the η larger,the result are always nan

5. If I set η=0.00005 and run it by 1000 times,the training line and the test line have just started to separate, but the rear is immediately re-closing.The final loss of validset is 11.8479129284

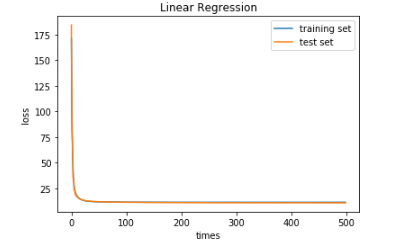
## Predicted Results (Best Results):

When η=0.001,I run the model by 1000 times and find that the model get a predicted results.The training line and the test line are

nearly close all the running time in the iteration. The final loss of

Valid set is 10.7634155621

## Loss curve:



**For Linear Classification,**

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

η=0.001,epoch=5000,C=0.9

## Assessment Results (based on selected validation):

Base on hold-out methos:

1.If I set η=0.001,C=0.9 and run it by 5000 times,I find the training curve and testing curve are close at begining,but setparate after several iteration and have some threshold.The final testing accuracy 0.850241545894

2.If I set η=0.0005,C=0.9 and run it by 5000 times,I find the training curve and testing curve are close at begining,but setparate after several iteration and have some threshold.The final testing accuracy 0.850241545894

3.If I set η=0.0001,C=0.9 and run it by 5000, there is some threshold on the training ans testing curves. The final testing accuracy 0.850241545894

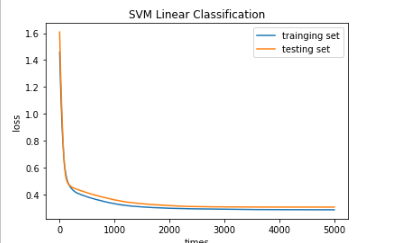
4.If I set η=0.00005,C=0.9 and run it by 5000 times,I find the training curve and testing curve are separated after several iterations.And there is no threshold anymore. But the final testing accuracy 0.850241545894. And If I turn the η smaller,the final testing accuracy may be more smaller.

5.If I set η=0.0015,C=0.9 and run it by 5000 times,I find the training curve and testing curve are close at begining,but setparate after several iteration and have many threshold.The final testing accuracy 0.855072463768.If I turn the ηlarger,there will be more threshold

## Predicted Results (Best Results):

When I set η=0.00005,C=0.9 and run it by 5000 times,I find the training curve and testing curve are nearly close all the time.And there is no threshold anymore. The final testing accuracy 0.850241545894

## Loss curve:



1. **Results analysis:**

**1)For Linear Regression:**

when eta is too large, loss may not fall back and finally be nan .For example,if I set η=0.0015,the result is nan .At this time, we must reduce eta.

If eta is too small, then it will lead to a very slow convergence.For example,when I set η<0.001,there is a phenomenon of slow convergence.

If eta is appropriate, the explanation is quickly convergent after a certain number of iterations is reached. For example, when η=0.001,the model get a predicted results.The training line and the test line are nearly close all the running time in the iteration. The final loss of validset is 10.7634155621.

**2)For Linear Classification:**

If eta is too large, the magnitude of the gradient is too large, and the global minimum is skipped.And thus,it will shake at the best value.For example,when I set η>0.0005, there are always some thresholds on the curve.

If eta is too small, then it will lead to a very slow convergence.For example,when I set η<0.0005, the speed of the convergence is obviously slow than η>0.0005

If eta is appropriate, the explanation is quickly convergent after a certain number of iterations is reached.For example,when I set η=0.0005,it gets a predicted results.The training line and the test line are nearly close all the running time in the iteration.What’s more, there is no threshold anymore.

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

**Similarities:**

1. The tlinear regression and linear classification are essentially the same, that is, the fitting (matching) of the model.
2. They are both solving Linear problem,using the same learning form,tha is .They both can use gradient descent to solve the problem.

**differences**

1. The classification problem is more (x in some areas) corresponding to (a Y). And the regression problem model is more inclined to (x in the small area, or generally a x) corresponds to (a Y).
2. The y value of the classification problem (also known as label) is more discrete while y value of the regression problem is always continuous.
3. They using different loss function and GD formula and other standards to measure the result.
4. **Summary:**

1.Through this experiment, I have realized linear regression and linear classification model。I found that these two kinds of problems have the following steps：

How to select a reasonable model (linear, or nonlinear (e.g. step function, Gauss function)).

Making a "nice" error function (it can evaluate the degree of fitting, and it's still a convex function).

The best model parameters are obtained by all possible techniques (such as gradient descent, close-formed)

2..Linear regression and linear classification are bothn linear models.So, they can use the same learning form,tha is .They both can use gradient descent to solve the problem.

3.By using the gradient descent algorithm, we iterate the weight matrix w step by step at every iteration, and point to point fitting until the loss value of the algorithm is stable, that is, the loss curve converges. And I understand that linear regression is an error analysis model based on Gauss distribution

4.When adjusting the superparameters, I observed the effect of the learning rate eta on the model prediction results.If eta is too large, the magnitude of the gradient is too large, and the global minimum is skipped.And thus,it will shake at the best value。If eta is too small, then it will lead to a very slow convergence。

5.When writing linear classification (SVM), I re understood the meaning of the soft interval, that is, hingeloss is the soft interval we require. Soft spacing is not defined as defined, but is learned by learning. Each sample has a soft interval. I have also reunderstood the relationship between the threshold and the soft interval, which is used to restrict the soft spaced.

6.There may be a problem of over fitting in the experiment. We can add regularization parameters or reduce the characteristics to prevent over fitting.