

**The Experiment Report of**

***Machine Learning***

**College Software College**

**Subject Software Engineering**

**Members**   **李恺哲**

**Student ID 201530611982**

**E-mail 623332032@qq.com**

**Tutor**   **吴庆耀**

**Date submitted** **2017. 12. 4**

**1. Topic:** linear regression and linear classification

**2. Time: 2017.12.02**

**3. Reporter:李恺哲**

**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:**

***Linear Regression and Gradient Descent***

1. **Load the experiment data.**
2. **Cut dataset.**
3. **Initialize linear model parameters.**
4. **Choose loss function and derivation**
5. **Calculate gradient toward loss function from all samples.**
6. **Denote the opposite direction of gradient as.**
7. **Update model**
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.**
4. **Choose loss function and derivation**
5. **Calculate gradient toward loss function from all samples.**
6. **Denote the opposite direction of gradient**
7. **Update model**
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 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.**

**7. Code: regression**

from sklearn.datasets import load\_svmlight\_file

import numpy as np

from sklearn.cross\_validation import train\_test\_split

import matplotlib.pyplot as plt

t\_X,t\_y=load\_svmlight\_file('housing\_scale') #读取数据

X\_train,X\_test, y\_train, y\_test =train\_test\_split(t\_X,t\_y,test\_size=0.4, random\_state=1)#分割数据

lossList=[]

def optimizer(X\_train,w,b):

learning\_rate=0.01

loss=0

for i in range(X\_train.shape[0]):

loss+=w\*X\_train[i,:].transpose()+b-y\_train[i]

G=-loss/X\_train.shape[0]

lossList.append(G)

print (G)

w=w-learning\_rate\*G

b=b-learning\_rate\*G

return w,b

w=np.zeros((13))

b=0

iter=500

for i in range(0,iter):

w,b=optimizer(X\_train,w,b)

iterList=list(range(500))

plt.plot(iterList,lossList)

plt.show()

svm：

**from** numpy **import** \*  
**import** matplotlib.pyplot **as** plt  
**import** math  
**from** sklearn.datasets **import** load\_svmlight\_file  
**import** numpy **as** np  
**from** sklearn.cross\_validation **import** train\_test\_split  
  
t\_X,t\_y=load\_svmlight\_file(**'australian\_scale'**) *#读取数据*X\_train,X\_test, y\_train, y\_test =train\_test\_split(t\_X,t\_y,test\_size=0.2, random\_state=1)*#分割数据***def** fit( x, y, c=1, lr=0.01, batch\_size=128,epoch=1000):  
 batch\_size = min(batch\_size, len(y))  
 w = np.zeros(x.shape[1])  
 b = 0.  
 **for** \_ **in** range(epoch):  
 w \*= 1 - lr  
 *# 随机选取 batch\_size 个样本* batch = np.random.choice(x.shape[0], batch\_size)  
 x\_batch, y\_batch = x[batch], y[batch]  
 err = 1 - y\_batch \* predict(x\_batch,w,b, True)  
 **if** np.max(err) <= 0:  
 **continue** mask = err > 0  
 delta = lr \* c \* y\_batch[mask]  
 w += np.mean(delta[..., None].transpose()\* x\_batch[mask], axis=0)  
 b += np.mean(delta)  
 **return** w,b  
  
**def** predict( x,w,b, raw=False):  
 y\_pred = x.dot(w) + b  
 **if** raw:  
 **return** y\_pred  
 **return** np.sign(y\_pred).astype(np.float32)  
  
w,b=fit(X\_train,y\_train)  
**print**(**"准确率：{:8.6} %"**.format((predict(X\_test,w,b) == y\_test).mean() \* 100))

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

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

All parameters are set to zero

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

(y1^2+y2^2+…)/(2\*m)

*Li*=∑*j*≠*yi*max(0,*wTjxi*−*wTyixi*+Δ)

**11. Experimental results and curve:**

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

## regression:η=0.01,iter=500

classification: η=0.01,iter=1000

## Assessment Results (based on selected validation):

Regression: loss: low

Classification: accuracy: 84%

## Predicted Results (Best Results):

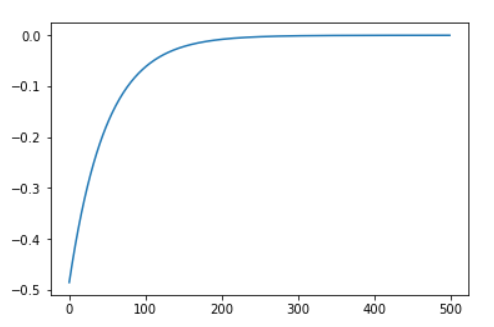
Regression: loss: low

Classification: accuracy: 92% on small dataset

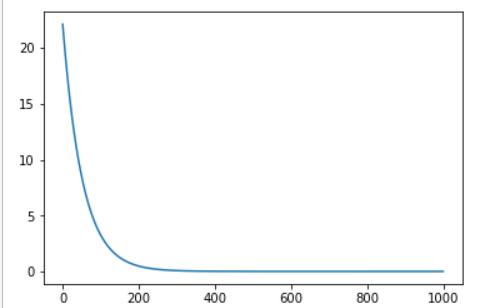
## Loss curve:

Regression:

![C:\Users\lkz\AppData\Roaming\Tencent\Users\623332032\TIM\WinTemp\RichOle\01TC4](8QUHXSS6RGP3UE`O.png](data:image/png;base64,)

Test: 

Classification:



**12. Results analysis:**

Regression: The linear regression works well on the small dataset, and it’s easy to train.

Classification: The linearsvm has a higher accuracy on the small dataset, also on a lower learning rate.

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

Similarity: They both build a model that fit the input x and y, and they are both linear model, every feature only has only one power.

Difference: The difference is that to regression, there has one separated predicted y to every one of the inputs x, which means the y is dispersed , while to classification, there are only limited outputs y to matching to the inputs x, which means y in a limited area.

**14. Summary:**

Through the lab, I get familiar with the basic machine learning coding skills, and learn deeper about the linear regression and the linear classification. For example, I didn’t realized that svm is one kind of linear classification method before, after the lab, I know it, and find out that there are only loss function’s difference between linearsvm and other kind of linear methods.