

**《机器学习》课程实验报告**

**学 院 软件学院**

**专 业 软件工程**

**组 员**  **谢家骏**

**学 号 201530613191**

**邮 箱 470552583@qq.com**

**指导教师**  **吴庆耀**

**提交日期** **2017年 12 月 10 日**

## 1. 实验题目: 线性回归、线性分类与梯度下降

## 2. 实验时间：2017年 12 月 2 日

## 3. 报告人: 谢家骏

## 4. 实验目的:

1. 进一步理解线性回归和梯度下降的原理。
2. 在小规模数据集上实践。
3. 体会优化和调参的过程。

## 数据集以及数据分析：

线性回归使用的是[LIBSVM Data](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/" \t "https://www.zybuluo.com/chenyaofo/note/_blank)中的[Housing](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html" \l "housing" \t "https://www.zybuluo.com/chenyaofo/note/_blank)数据，包含506个样本，每个样本有13个属性。

线性分类使用的是[LIBSVM Data](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/" \t "https://www.zybuluo.com/chenyaofo/note/_blank)中的[australian](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html" \l "australian" \t "https://www.zybuluo.com/chenyaofo/note/_blank)数据，包含690个样本，每个样本有14 个属性。

## 实验步骤:

*线性回归和梯度下降*

1. 读取实验数据，使用sklearn库的[load\_svmlight\_file](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_svmlight_file.html" \t "https://www.zybuluo.com/chenyaofo/note/_blank)函数读取数据。
2. 将数据集切分为训练集和验证集，本次实验不切分测试集。使用[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)函数切分数据集。
3. 线性模型参数初始化，可以考虑全零初始化，随机初始化或者正态分布初始化。
4. 选择Loss函数及对其求导，过程详见课件ppt。
5. 求得****所有样本****对Loss函数的梯度。
6. 取梯度的负方向，记为。
7. 更新模型参数，。为学习率，是人为调整的超参数。
8. 在训练集上测试并得到Loss函数值，在验证集上测试并得到Loss函数值。
9. 重复步骤5-8若干次，****画出和随迭代次数的变化图****。

*线性分类和梯度下降*

1. 读取实验数据，使用sklearn库的[load\_svmlight\_file](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_svmlight_file.html" \t "https://www.zybuluo.com/chenyaofo/note/_blank)函数读取数据。
2. 将数据集切分为训练集和验证集，本次实验不切分测试集。使用[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)函数切分数据集。
3. 支持向量机模型参数初始化，可以考虑全零初始化，随机初始化或者正态分布初始化。
4. 选择Loss函数及对其求导，过程详见课件ppt。
5. 求得****所有样本****对Loss函数的梯度。
6. 取梯度的负方向，记为。
7. 更新模型参数，。为学习率，是人为调整的超参数。
8. 选择合适的阈值，将计算结果****大于阈值的标记为正类，反之为负类****。在训练集上测试并得到Loss函数值，在验证集上测试并得到Loss函数值。
9. 重复步骤5-8若干次，****画出和随迭代次数的变化图****。

## 代码内容:

线性回归与梯度下降：

from sklearn import datasets as ds

from sklearn.model\_selection import train\_test\_split

import numpy as np

from numpy import random

import matplotlib.pyplot as plt

def Loss(X,y,w):

temp = y - np.dot(X,w)

loss = np.dot(temp.T,temp)/2

return loss

def Grad(X,y,w):

grad = np.dot(X.T,np.dot(X,w)-y)

return grad

def iteration(x\_train,x\_validation,y\_train,y\_validation,w):

itera = 100

lr = 0.0001

train\_loss=[]

validation\_loss=[]

for i in range(itera):

loss\_t=Loss(x\_train,y\_train,w)

tx,ty=x\_train.shape

lt=loss\_t[0,0]/tx

train\_loss.append(lt)

loss\_v=Loss(x\_validation,y\_validation,w)

vx,vy=x\_validation.shape

lv=loss\_v[0,0]/vx

validation\_loss.append(lv)

w=w-lr\*Grad(x\_train,y\_train,w)

return w,train\_loss,validation\_loss

if \_\_name\_\_ == '\_\_main\_\_':

x\_train, y\_train = ds.load\_svmlight\_file('E:/test/test.txt')

x\_train = x\_train.toarray()

yl=len(y\_train)

y\_train=y\_train.reshape(yl,1)

n\_sample,n\_feature=x\_train.shape

x\_train = np.concatenate([x\_train, np.ones(shape=[506, 1], dtype=np.float32)], axis=1)

x\_train, x\_validation, y\_train, y\_validation = train\_test\_split(x\_train, y\_train, test\_size=0.2, random\_state=1)

n\_sample,n\_feature=x\_train.shape

w0=np.zeros(shape=(n\_feature,1))

w,train\_loss,validation\_loss=iteration(x\_train, x\_validation, y\_train, y\_validation,w0)

plt.plot(train\_loss, label='train loss')

plt.plot(validation\_loss, label='validation loss')

plt.xlabel('Iteration')

plt.ylabel('Loss')

plt.legend()

plt.show()

线性分类与梯度下降：

from sklearn import datasets as ds

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

import numpy as np

import matplotlib.pyplot as plt

import os

import matplotlib.pyplot as plt

def Loss(x\_train,y\_train,w,C,b):

ls=0

i=0

for x\_sample, y\_sample in zip(x\_train,y\_train):

temp = np.ones((n\_sample,1)) - y\_sample \* np.dot(x\_sample,w)

if(0<temp[i,0]):

ls+=temp[i,0]

else:

ls+=0

i+=1

loss=np.dot(w.T,w)/2+C\*ls

return loss

def Gradw(x\_train,y\_train,w,C,b):

g=0

i=0

for x\_sample, y\_sample in zip(x\_train,y\_train):

temp = np.ones((n\_sample,1)) - y\_sample \* np.dot(x\_sample,w)

if(temp[i,0]>0):

gw=-y\_sample\*x\_sample

else:

gw=0

g+=gw

i+=1

gradw=w+C\*g

return gradw

def Gradb(x\_train,y\_train,w,C,b):

g1=0

i=0

for x\_sample, y\_sample in zip(x\_train,y\_train):

temp = np.ones((n\_sample,1)) - y\_sample \* np.dot(x\_sample,w)

if(temp[i,0]>0):

gb=-y\_sample

else:

gb=0

g1+=gb

i+=1

gradb=C\*g1

return gradb

def iteration(x\_train,x\_validation,y\_train,y\_validation,w,C,b):

itera = 1000

lr = 0.01

train\_loss=[]

validation\_loss=[]

for i in range(itera):

loss\_t=Loss(x\_train,y\_train,w,C,b)

tx,ty=x\_train.shape

lt=loss\_t[0,0]/tx

train\_loss.append(lt)

loss\_v=Loss(x\_validation,y\_validation,w,C,b)

vx,vy=x\_validation.shape

lv=loss\_v[0,0]/vx

validation\_loss.append(lv)

w=w-lr\*Gradw(x\_train,y\_train,w,C,b)

b=b-lr\*Gradb(x\_train,y\_train,w,C,b)

return w,b,train\_loss,validation\_loss

if \_\_name\_\_ == '\_\_main\_\_':

C=0.1

x\_train, y\_train = ds.load\_svmlight\_file('E:/test/2.txt')

x\_train = x\_train.toarray()

yl=len(y\_train)

y\_train=y\_train.reshape(yl,1)

n\_sample,n\_feature=x\_train.shape

x\_train, x\_validation, y\_train, y\_validation = train\_test\_split(x\_train, y\_train, test\_size=0.2, random\_state=1)

n\_sample,n\_feature=x\_train.shape

w0=np.zeros(shape=(n\_feature,1))

b=0

w,b,train\_loss,validation\_loss=iteration(x\_train, x\_validation, y\_train, y\_validation,w0,C,b)

plt.plot(train\_loss, label='train loss')

plt.plot(validation\_loss, label='validation loss')

plt.xlabel('Iteration')

plt.ylabel('Loss')

plt.legend()

plt.show()

（针对线性回归和线性分类分别填写8-12内容）

## 8. 选择的评估方法（留出法，交叉验证，k折交叉验证等）:

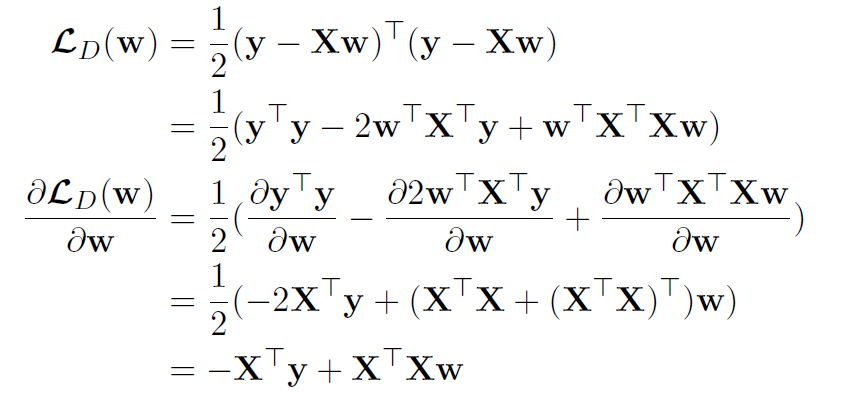
## 9. 模型参数的初始化方法:

线性回归与梯度下降：全零初始化

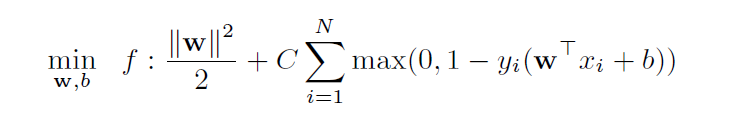
线性分类与梯度下降：全零初始化

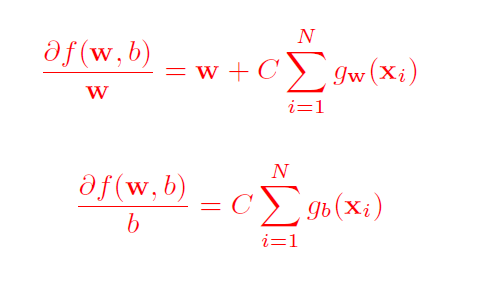
## 选择的loss函数及其导数:

线性回归与梯度下降：



线性分类与梯度下降：





## 11.实验结果和曲线图:

## 超参数选择（η,epoch等）：

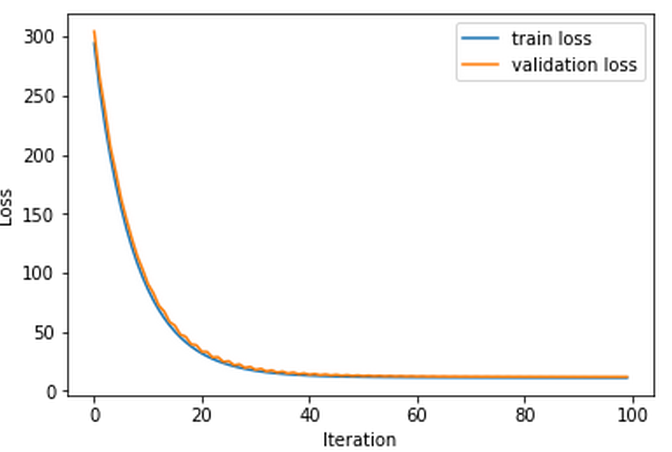
线性回归与梯度下降：itera = 100,η=0.0001

线性分类与梯度下降：itera = 1000,η=0.01

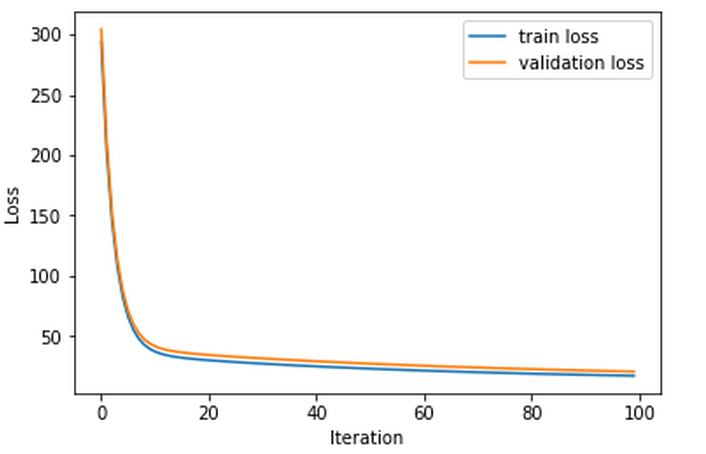
## 评估结果（根据选择的评估方法）：

线性回归与梯度下降：

itera = 100,η = 0.001

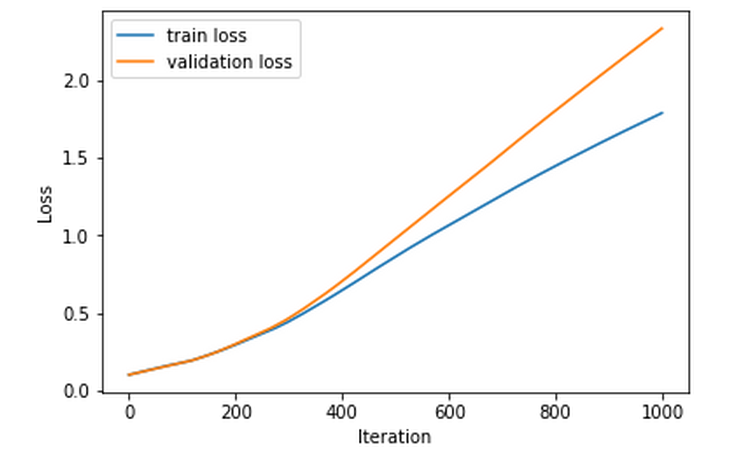


itera = 100,η = 0.0001

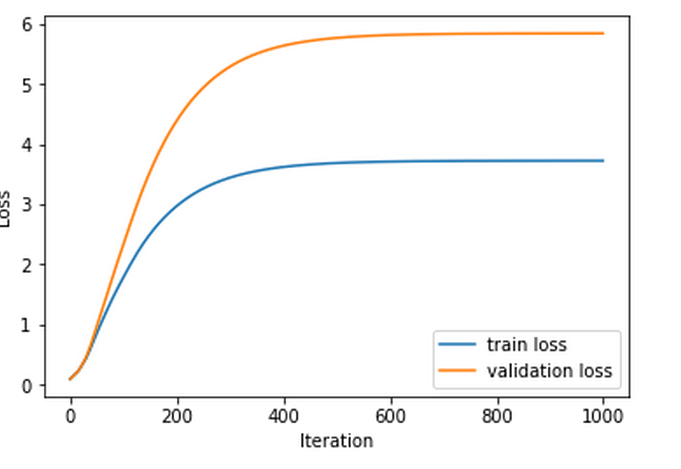


线性分类与梯度下降：

itera = 1000,η=0.001



itera = 1000,η=0.01



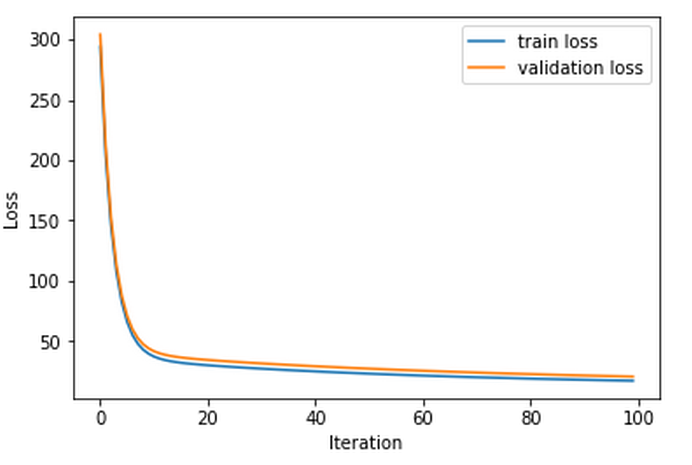
## 预测结果（最佳结果）：

线性回归与梯度下降：η=0.0001

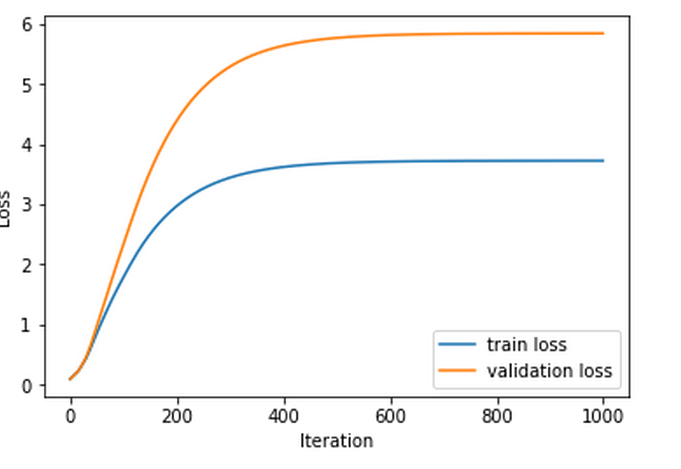
线性分类与梯度下降：η=0.01

## loss曲线图：

线性回归与梯度下降：



线性分类与梯度下降：



## 实验结果分析:

线性回归与梯度下降：Loss值会随着迭代次数的增加趋于一个稳定值

线性分类与梯度下降：Loss值会随着迭代次数的增加趋于一个稳定值

## 对比线性回归和线性分类的异同点：

相同点：本质上是一致的，就是模型的拟合

不同点：Loss函数及其导数、梯度下降函数的实现都不同

## 实验总结：

通过这次实验，我进一步了解了线性回归、线性分类与梯度下降的原理，并可以在小规模数据集上实践，同时也体会了优化和调参的过程。