

**《机器学习》课程实验报告**

**学 院 软件学院**

**专 业 软件工程**

**组 员**  **符王朝**

**学 号 20153061170**

**邮 箱 2262458345@qq.com**

**指导教师**  **吴庆耀**

**提交日期** **2017年 12 月 7 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:符王朝

## 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。

## 5.数据集以及数据分析：

线性回归使用的是LIBSVM Data中的Housing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中的australian数据，包含690个样本，每个样本有14 个属性。

## 6.实验步骤:

本次实验代码及画图均在jupyter上完成。

*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 "_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)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 "_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)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**。

## 7. 代码内容:

（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）:

留出法

from numpy import \*

import numpy as np

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.33, random\_state=42)

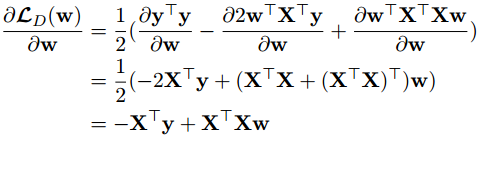
## 模型参数的初始化方法:

全零初始化

## 选择的loss函数及其导数:

线性回归：



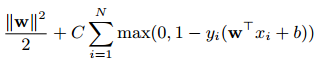


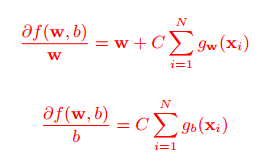
w = w - n \* ( 2 \* X2.T \*(X2 \* w - yy.T) )

loss = 1/2 \* (( yy.T - X2\*w ).T ) \* ( yy.T - X2 \* w)

lossTest = 1/2 \* (( yyy.T - X3\*w ).T ) \* ( yyy.T - X3 \* w)

线性分类





def gw(i):

if 1 - yy[0,i] \* ( dot(w.T,X\_train[i].T)[0,0] + b) >= 0:

return -dot(yy[0,i],X\_train[i].T)

else:

return 0

def gb(i):

if 1 - yy[0,i] \* (dot(w.T,X\_train[i].T)[0,0] + b) >= 0:

return -yy[0,i]

else:

return 0

def max(i):

if 1 - yy[0,i] \* (dot(w.T,X\_train[i].T)[0,0] + b) >= 0:

return 1 - yy[0,i] \* (dot(w.T,X\_train[i].T)[0,0] + b)

else:

return 0

def maxt(i):

if 1 - yyy[0,i] \* (dot(w.T,X\_test[i].T)[0,0] + b) >= 0:

return 1 - yyy[0,i] \* (dot(w.T,X\_test[i].T)[0,0] + b)

else:

return 0

def mul(w):

sum = 0

for i in range(14):

sum += w[i,0]\*\*2

return sum

## 11.实验结果和曲线图:

## 超参数选择（η,epoch等）：

线性分类：

n = 0.00001

线性分类：

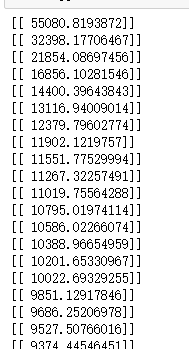
n = 0.001 #rate

C = 0.1

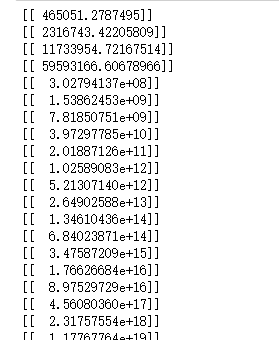
## 评估结果（根据选择的评估方法）：

线性回归：

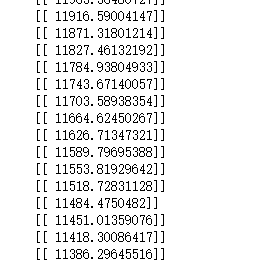
N = 0.0001



N = 0.001

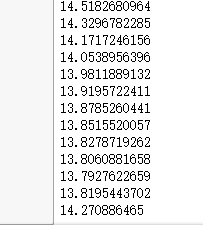


N=0.00001

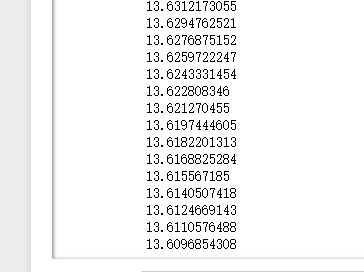


线性分类：

N = 0.01 C = 0.1

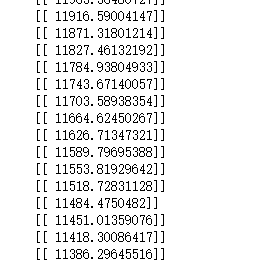


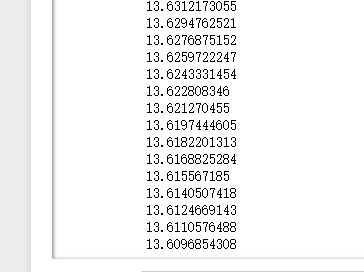
N=0.001 c= 0.1



## 预测结果（最佳结果）：

线性回归：

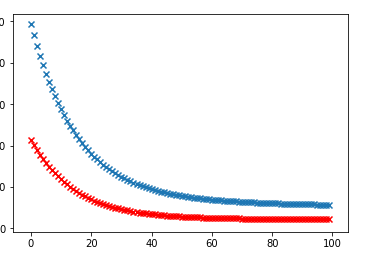
线性分类



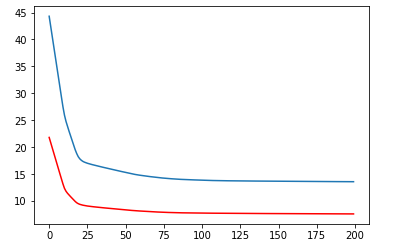
## loss曲线图：

线性回归

红：验证集



线性分类

红：验证集

## 实验结果分析:

N c 的数值很影响最后的图示

## 13.对比线性回归和线性分类的异同点：

总的来说两个问题本质上都是一致的，就是模型的拟合（匹配）。 但是分类问题的y值(也称为label), 更离散化一些. 而且， 同一个y值可能对应着一大批的x,  这些x是具有一定范围的。

所以分类问题更多的是 (一定区域的一些x) 对应 着 (一个y).   而回归问题的模型更倾向于 (很小区域内的x，或者一般是一个x)  对应着  (一个y).

在把一个问题建模的时候一定要考虑好需求，让你的模型更好的与现实问题相对应。

## 14.实验总结：

加深对线性分类和回归的了解，获益匪浅