

**《机器学习》课程实验报告**

**学 院 软件学院**

**专 业 软件工程**

**组 员**  **郑铭莉**

**学 号 201530613801**

**邮 箱** [**727457905@qq.com**](mailto:727457905@qq.com)

**指导教师**  **吴庆耀**

**提交日期** **2017年 12 月 2 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: 郑铭莉

## 4. 实验目的:

进一步理解线性回归和梯度下降的原理。

在小规模数据集上实践。

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。请自行下载scaled版本，并将其切分为训练集，验证集。

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。请自行下载scaled版本，并将其切分为训练集，验证集。

## 6. 实验步骤:

本次实验代码及画图均在jupyter上完成。

（1）线性回归和梯度下降

1.读取实验数据，使用sklearn库的load\_svmlight\_file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G。

6.取梯度的G负方向，记为D。

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。

（2）线性分类和梯度下降

1.读取实验数据，使用sklearn库的load\_svmlight\_file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G。

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:

线性回归梯度下降：

for i in range(iteration):

for j in range(x\_train.shape[0]):

gradient = x\_train[j].T \* (x\_train[j] \* w - y\_train[j]) + lamb \* w

w = w - learning\_rate \* gradient

线性分类梯度下降：

for i in range(iteration):

for j in range(x\_train.shape[0]):

if np.sum((1 - y\_train[j] \* (x\_train[j] \* w + b[j] ))) > 0:

w\_gradient = w + x\_train[j].T \* C \* (-1 \* y\_train[j])

b\_gradient = -1 \* C \* y\_train[j]

else:

w\_gradient = w

b\_gradient = 0

w = w - learning\_rate \* w\_gradient

b[j] = b[j] - learning\_rate \* b\_gradient

## 8. 选择的评估方法（留出法，交叉验证，k折交叉验证等）:

线性回归：留出法

线性分类：留出法

## 9. 模型参数的初始化方法:

线性回归：w = np.empty((x\_train.shape[1],1)) 随机赋值

b\_train = np.zeros(x\_train.shape[0]) 全为0

线性分类：b = np.zeros((x\_train.shape[0],1)) 全为0

w = np.empty((x\_train.shape[1],1)) 随机赋值

## 10.选择的loss函数及其导数:

线性回归：



线性分类：

## 11.实验结果和曲线图:

## 超参数选择（η,epoch等）：

线性回归：η=0.01，λ=0.0001，epoch=100

线性分类：η=0.01，C=0.01

## 评估结果（根据选择的评估方法）：

**线性回归：**

η=0.01，λ=0.0001，epoch=100, =12.5694026303，

=12.6497240934

η=0.001，λ=0.0001，epoch=100, =6255.46139184，

=4672.87627821

η=0.1，λ=0.0001，epoch=100, =13.87836326，

= 13.87836326

**线性分类：**

固定C=0.01，调节η：

η=0.9，C=0.01, =0.431602759696，

= 2.22905384465

η=0.1，C=0.01, = 4.12701841918，

= 2.26153730601

固定η=0.9，调节C：

η=0.9，C=0.001, = 0.420116027597，

= 0.227490538446

η=0.9，C=0.1, = 0.0，

= 23.863

## 预测结果（最佳结果）：

线性回归：

η=0.01，λ=0.0001，epoch=100, =12.5694026303，

=12.6497240934

线性分类：

η=0.9，C=0.001, = 0.420116027597，

= 0.227490538446

## loss曲线图

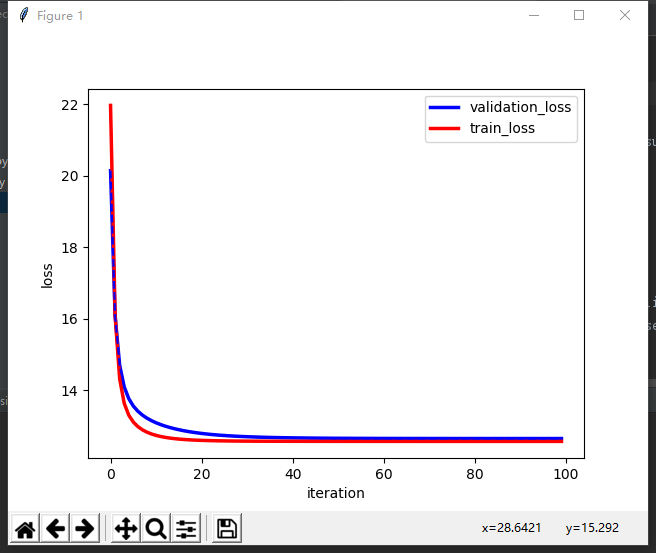


图1. 线性回归

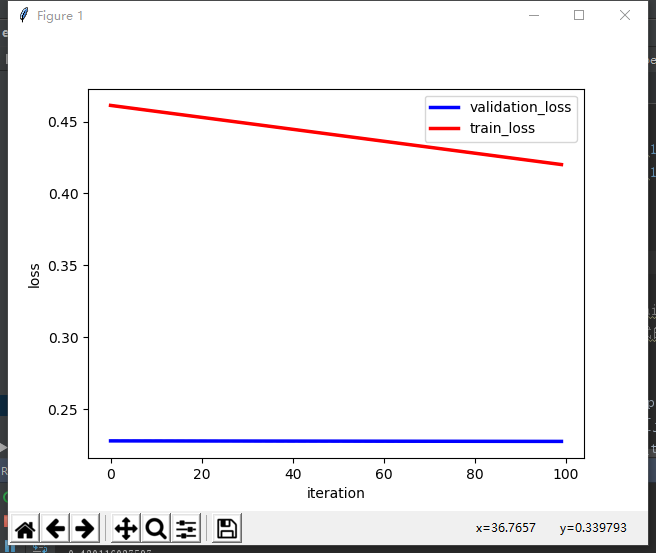


图2. 线性分类

## 12.实验结果分析:

线性回归：经过十几次迭代后收敛基本稳定，和比较接近

线性分类：收敛很快，而随迭代次数增加逐渐增大，出现过拟合现象

## 13.对比线性回归和线性分类的异同点：

   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)。

## 14.实验总结：

在实验过程中，发现调参数是一件很重要的工作，参数稍微有一点变化就会对结果产生很大影响，比如线性分类中，c越大说明模型容忍了更多的噪声，c小到一定程度会收敛，但太小了就会过拟合，学习率太小发生比小，太大收敛得很快且loss值都很小，但迭代次数增加后发生过拟合。

另外在线性分类中，发现数据集的划分对结果也有很大影响，

另外通过实验加深了算法的计算过程，平时都是直接调用函数和库，动手写计算过程增强了对算法的理解，同时提高了编程能力和应用能力。