

**《机器学习》课程实验报告**

**学 院 软件学院**

**专 业 软件工程**

**组 员**  **蔡天鑫**

**学 号 201530611607**

**邮 箱 280087602@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:蔡天鑫

## 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。

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:

线性回归和梯度下降

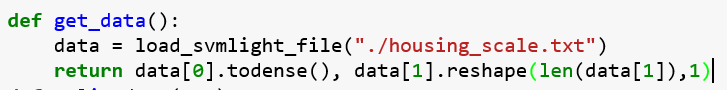
1. 读取实验数据，使用sklearn库的load\_svmlight\_file函数读取数据。
2. 将数据集切分为训练集和验证集，本次实验不切分测试集。使用train\_test\_split函数切分数据集。
3. 线性模型参数初始化，可以考虑全零初始化，随机初始化或者正态分布初始化。
4. 选择Loss函数及对其求导，过程详见课件ppt。
5. 求得所有样本对Loss函数的梯度G。
6. 取梯度G的负方向，记为D
7. 更新模型参数，，
8. 在训练集上测试并得到Loss函数值，在验证集上测试并得到Loss函数值。
9. 重复步骤5-8若干次，画出和随迭代次数的变化图。

线性分类和梯度下降

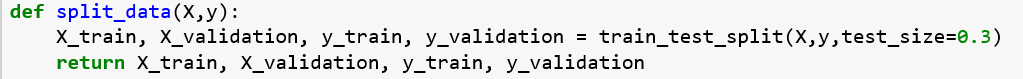
1. 读取实验数据，使用sklearn库的load\_svmlight\_file函数读取数据。
2. 将数据集切分为训练集和验证集，本次实验不切分测试集。使用train\_test\_split函数切分数据集。
3. 线性模型参数初始化，可以考虑全零初始化，随机初始化或者正态分布初始化。
4. 选择Loss函数及对其求导，过程详见课件ppt。
5. 求得所有样本对Loss函数的梯度G。
6. 取梯度G的负方向，记为D
7. 更新模型参数，，
8. 选择合适的阈值，将计算结果大于阈值的标记为正类，反之为负类。在训练集上测试并得到Loss函数值，在验证集上测试并得到Loss函数值。
9. 重复步骤5-8若干次，画出和随迭代次数的变化图。

## 7. 代码内容:

获取数据集：



切分数据集：

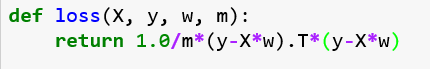


RegressionExperiment:

计算梯度：

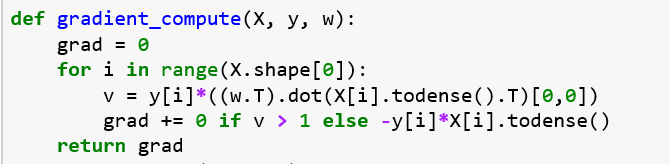


计算Loss值：

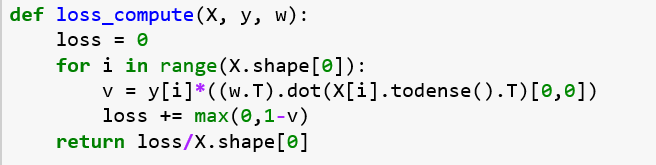


ClassificationExperiment:

计算梯度：



计算Loss值：



## 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

## 9. 模型参数的初始化方法:

Regression和Classification我采用的都是全0初始化：



## 10.选择的loss函数及其导数:

Regression:

Loss函数：

导数：

Classification:

Loss函数：

导数：

## 11.实验结果和曲线图:

## 超参数选择（η,epoch等）：

Regression:

η = 0.25,

迭代次数：1000

Classification:

η = 0.0005

迭代次数：100

## 评估结果（根据选择的评估方法）：

Linear Regression:

η = 0.25,loss\_train = 23.993,loss\_validation = 24.918

η = 0.2,loss\_train = 23.735,loss\_validation = 25.522

η = 0.1,loss\_train = 25.989,loss\_validation = 20.270

η = 0.05,loss\_train = 22.236,loss\_validation = 29.034

Linear Classification:

η = 0.05.Loss曲线震荡严重

η = 0.005.Loss曲线震荡严重

η = 0.0005.Loss曲线正常，loss\_train = 0.307,loss\_validatioin = 0.276

## 预测结果（最佳结果）：

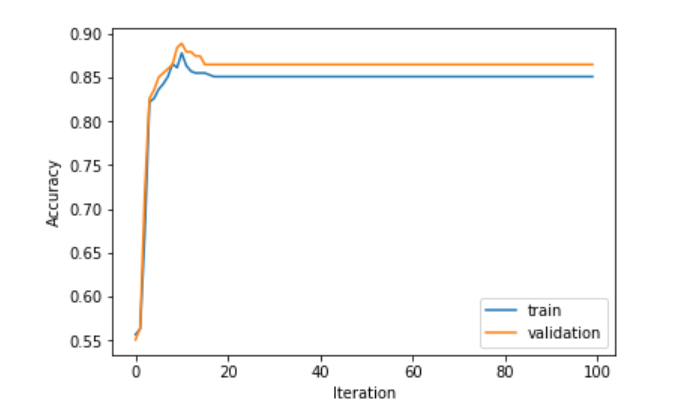
Linear Regression:

η = 0.25,loss\_train = 23.993,loss\_validation = 24.918

Linear Classification:

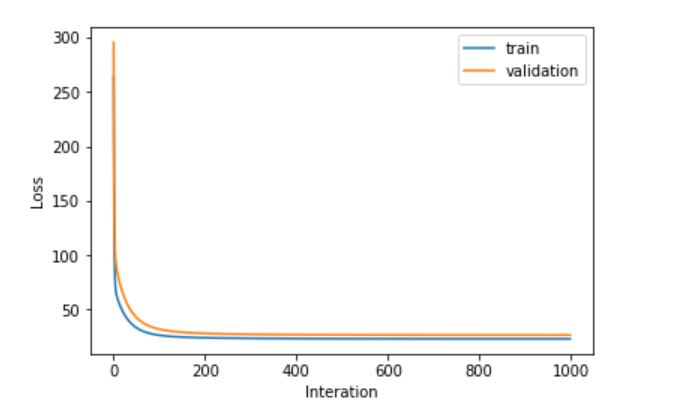
η = 0.0005,loss\_train = 0.307,loss\_validatioin = 0.276

此时的Accuracy随迭代次数的曲线：

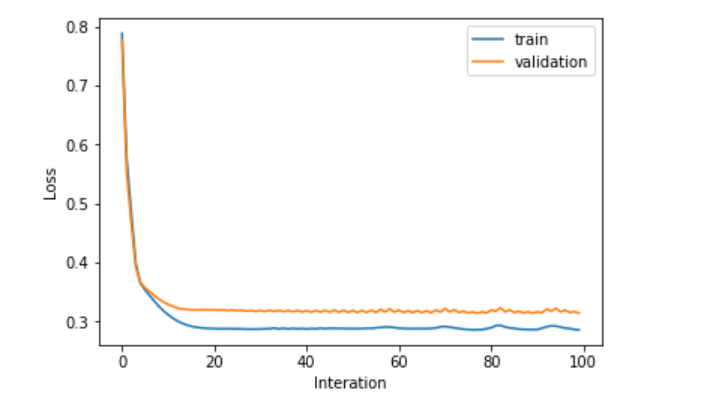


## loss曲线图：

Linear Regression:



Linear Classification:



## 12.实验结果分析:

Linear Regression:

随着迭代次数的增加，Loss值一开始剧烈减小，然后趋于平稳，并且train和validation集的Loss变化情况基本一致且相差不大，实验结果正常。

Linear Classification:

随着迭代次数的增加，Loss值一开始剧烈减小，然后趋于平稳，但是会出现细小的锯齿形震荡，而且当学习率设置大的时候，震荡更加明显。train和validation集的Loss变化情况基本一致，实验结果基本正常。

## 13.对比线性回归和线性分类的异同点：

线性回归和线性分类的function采用的都是线性函数y=wx+b,

不同的是，线性回归的关注点在于function返回的数值，通过梯度下降更新参数去拟合超平面

线性分类将function的值变成了离散的{0，1}点，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：

通过本次实验，我加深了对线性回归和线性分类（SVM）的理解，自己亲手实现了一遍代码，感觉中途遇到了很多问题：

首先是将数学公式转化为代码：通过python进行了很繁琐的矩阵、向量运算，手打这些真的要调很久bug。

其次，在调参方面，有时候会看到奇奇怪怪的Loss图还怎么都改不对。

最后，本次实验还教会了我们用很常用的jupyter 写python代码，图文并茂，功能很强大。

总之，本次实验是对我书本知识的强化，也是动手能力的培养，收获良多。