

**《机器学习》课程实验报告**

**学 院 软件学院**

**专 业 软件工程**

**组 员**  **王圣杰**

**学 号 201530612873**

**邮 箱** [**2256989051@qq.com**](mailto:2256989051@qq.com)

**指导教师**  **吴庆耀**

**提交日期** **2017年12月08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: 王圣杰

## 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。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:

***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。
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，Wt = Wt-1 + ηD。η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L\_train，在验证集上测试并得到Loss函数值L\_test。
9. 重复步骤5-8若干次，**画出**L\_train**和**L\_test**随迭代次数的变化图**。

**线性分类和梯度下降**

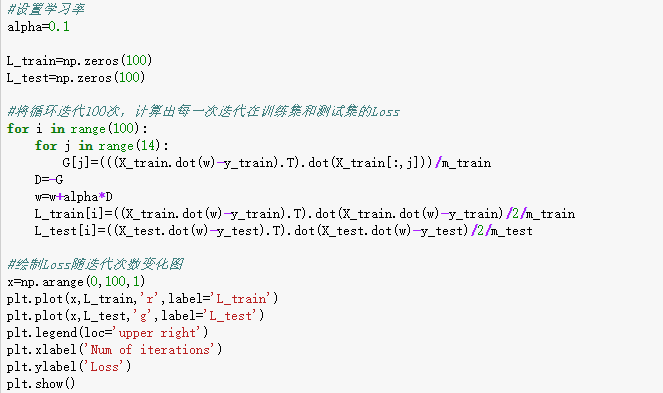
1. 读取实验数据，使用sklearn库的load\_svmlight\_file函数读取数据。
2. 将数据集切分为训练集和验证集，本次实验不切分测试集。使用train\_test\_split函数切分数据集。
3. 支持向量机模型参数初始化，可以考虑全零初始化，随机初始化或者正态分布初始化。
4. 选择Loss函数及对其求导，过程详见课件ppt。
5. 求得**所有样本**对Loss函数的梯度G。
6. 取梯度G的负方向，记为D。
7. 更新模型参数，Wt = Wt-1 + ηD。η为学习率，是人为调整的超参数。
8. 选择合适的阈值，将计算结果**大于阈值的标记为正类，反之为负类**。在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

## 7. 代码内容

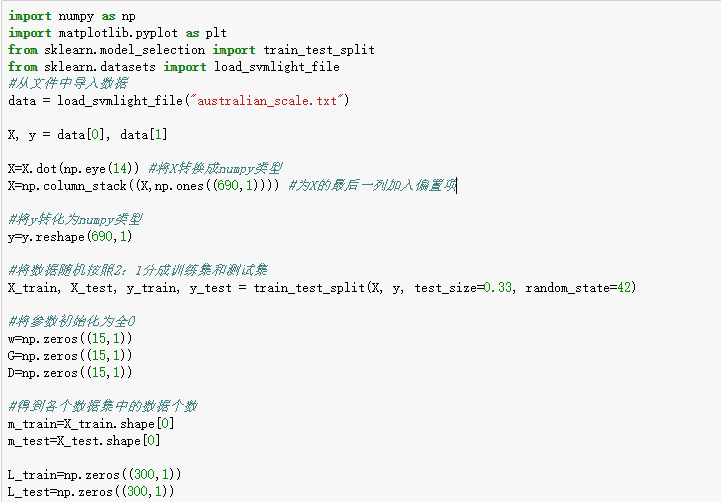
(由于代码复制过来格式被破坏，所以这里给的是截图，代码详见GitHub仓库)

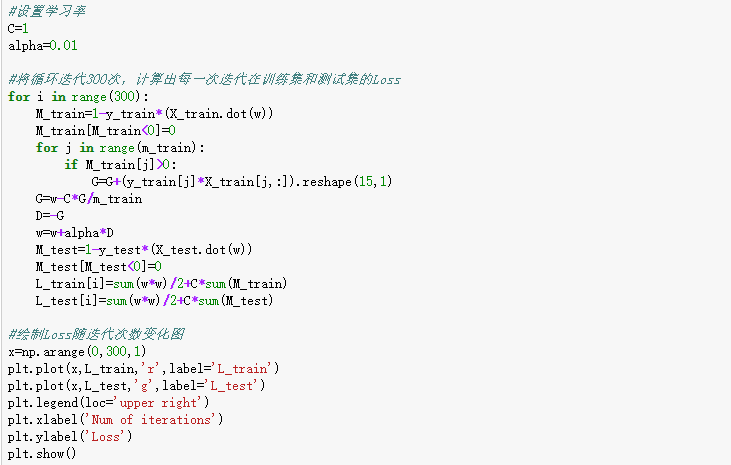
线性回归：





线性分类：





## 8. 选择的评估方法:

线性回归和线性分类采用的评估方法都是将数据集分割成训练集和测试集，在训练集上训练样本得到模型参数，最后在测试集上测试训练得到的模型

## 9. 模型参数的初始化方法:

线性回归和线性分类采用的模型参数的初始化方法都是全零初始化

## 10.选择的loss函数及其导数:

线性回归：

Loss函数：(Xw-y)T(Xw-y)/2/m

导数：X T (Xw-y)/m

线性分类：

Loss函数：||w||²/2+CTXi+b))

导数：w－CXi (for each (Xi,yi) has 1-yi(wTXi+b)>0)

## 11.实验结果和曲线图:

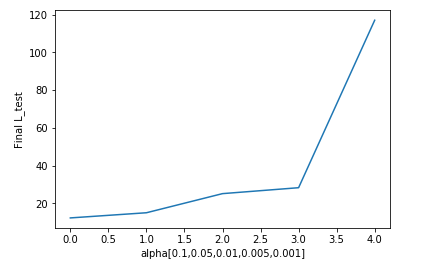
## 超参数选择：

线性回归：学习率α=0.1

线性分类：学习率α=0.01，C=1

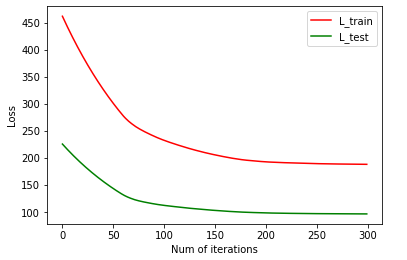
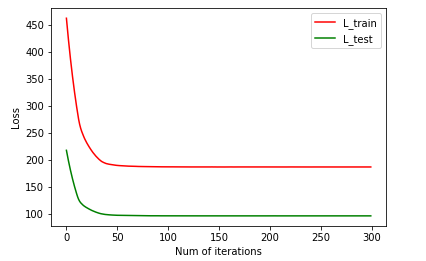
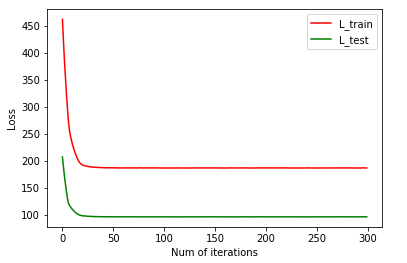
## 评估结果（根据选择的评估方法）：

线性回归：图中的横坐标从左到右为alpha分别取（0.1，0.05，0.001，0.005，0.001），由图可见当alpha=0.1的时候，最终的L\_test最小



线性分类：有下面三个图，我认为，当alpha取0.01的时候既能使下降的速度适中，也能保持算法稳定

alpha=0.1 alpha=0.05 alpha=0.01



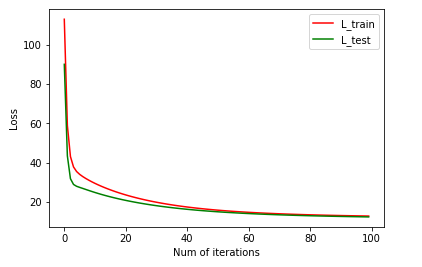
## 预测结果（最佳结果）：

线性回归：alpha=0.1，并最后在测试集上的Loss为L\_test=12.33486608，继续增大迭代次数能再稍微降低L\_test

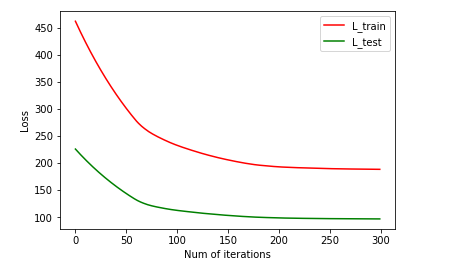
线性分类：C=1，alpha=0.01，并最后在测试集上的Loss为L\_test= 96.82902398，继续增大迭代次数能再稍微降低L\_test

## loss曲线图：

线性回归：alpha=0.1，在此模型上若再加大迭代次数还能继续使L\_test下降



线性分类：C=1，alpha=0.01，在此模型上若再加大迭代次数还能继续使L\_test下降在此模型上若再加大迭代次数还能继续使L\_test下降



## 12.实验结果分析:

线性回归：模型在迭代的时候下降的速率很快，在前几次迭代后就已经下降到一个比较小的值，后面的迭代中下降的幅度越来越小，L\_test降低到了一定值后增加迭代次数也改变基本不大。下降的速率和学习率有很大关系，过大的学习率会导致模型Loss出现增长的情况，学习率过小又会导致模型收敛速率过低。

线性分类：因为这里我的学习率取的是一个比较小的值，目的就是为了让模型能够稳定收敛。最后训练出来的模型在测试集上有85%左右的准确率，增加迭代次数，准确率也基本不变

## 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，就是模型的匹配。但是分类问题的y值更离散化一些. 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：

通过本次实验，我深入的学习了线性回归以及采用SVM实现的线性分类模型的实现和参数调试，学习了如何利用Jupyter Notebook来编写模型代码，了解了sklearn，numpy，jupyter，matplotlib包的一些基本操作。通过自己手写代码，让我进一步理解了线性回归、线性分类和梯度下降的原理，并且在小规模的数据集上实践，通过绘制图像来可视化优化和调参的过程，让这种过程给自己更深的体会和收获，在试验过程总，我有几次打错了，算法的代码，都是通过绘制出的Loss-Iterations图观察出来的，还有在调参的过程中通过更改参数来观察图形的变化让我直观地体会到了调参的过程。通过这次试验，让我对于机器学习有了真实的接触，为我之后的实验打下了必要的的基础，并在过程中收获了对于不同问题的处理方法，积累了宝贵的经验。