

**《机器学习》课程实验报告**

**学 院 软件学院**

**专 业 软件工程**

**组 员**  黄景浩

**学 号 201530611739**

**邮 箱** [**394876962@qq.com**](mailto:394876962@qq.com)

**指导教师**  **吴庆耀**

**提交日期** **2017年12月8日**

## 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: 黄景浩

## 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。

## 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，。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Ltrain，在验证集上测试并得到Loss函数值Lvalidation。
9. 重复步骤5-8若干次，画出Ltrain和Lvalidation随迭代次数的变化图。

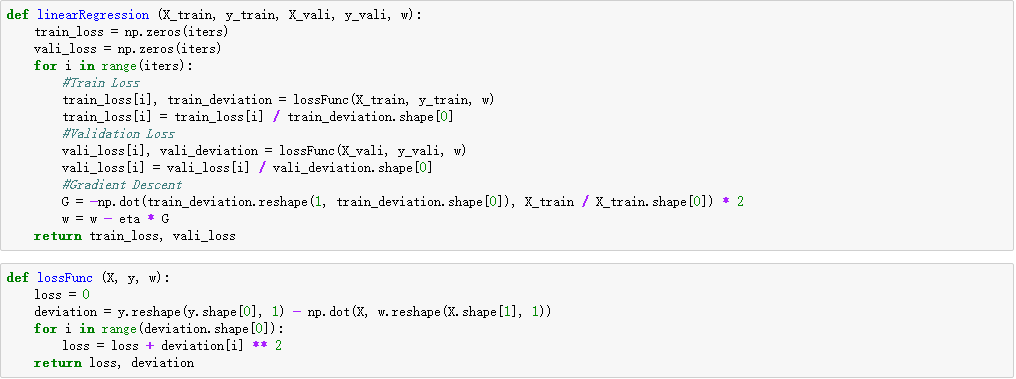
**线性分类和梯度下降：**

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. 取梯度的负方向G，记为。D
7. 更新模型参数，。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Ltrain，在验证集上测试并得到Loss函数值Lvalidation。
9. 重复步骤5-8若干次，画出Ltrain和Lvalidation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）

线性回归：



线性分类：



## 选择的评估方法: 留出法

## 9. 模型参数的初始化方法: 正态分布初始化

## 10.选择的loss函数及其导数:

## 11.实验结果和曲线图:

## 超参数选择（η,epoch等）：

线性回归：

η= 0.1

线性分类:

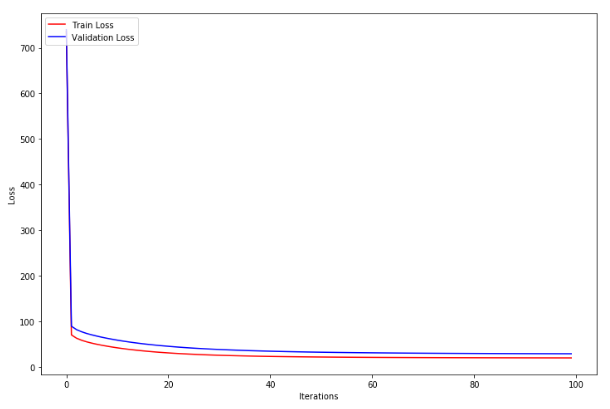
η= 0.0005

## 评估结果（根据选择的评估方法）：

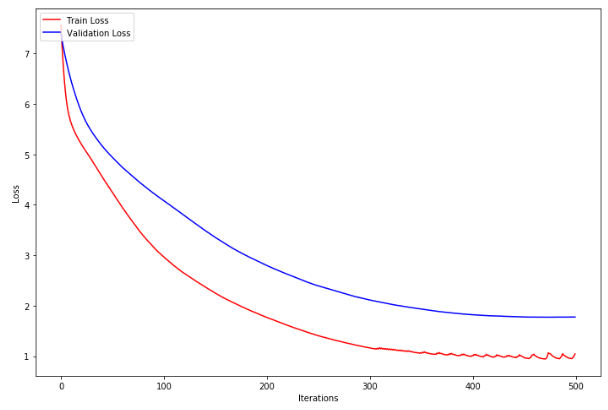
## 预测结果（最佳结果）：

## loss曲线图：

线性回归：



线性分类：



## 12.实验结果分析:

线性回归：

训练集与验证集表现相近，体现出数据分布的一致性。第1次迭代loss快速下降，随着迭代次数增长，loss下降速率快速减缓。实验中发现100次迭代以后即使减小步长η，loss也基本不再下降，维持在25以上（使用L2 loss）。这应该是由数据本身分布所导致的。

线性分类：

训练集与验证集表现相近，体现出数据分布的一致性。前100次迭代中loss快速下降，随着迭代次数增加，loss减小速度减缓。在调节参数适当后，线性分类loss可降低至1.0左右（L1 hinge loss），实验中发现若以0为阈值，最高约有85%的分类准确率。但在迭代300次以后，即使减小步长η也无法再提高准确率，减小loss。这应该是由数据本身分布所导致的。

## 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均使用梯度下降算法迭代逼近最优解

线性回归使用线性回归算法，线性分类使用SVM hinge loss算法。

## 14.实验总结：

通过本次实验，我更深入地理解了线性回归，SVM算法，同时也学习了Python语言及相关库函数的调用，提高了我解决实际问题的能力，如求解梯度时如果完全按照公式计算会得到过于大的数字导致溢出，通过进行适当的等比例缩小保证正常计算。