

**《机器学习》课程实验报告**

**学 院 软件学院**

**专 业 软件工程**

**组 员**  **邢正颖**

**学 号 201530613238**

**邮 箱 137339291@qq.com**

**指导教师** 谭明奎

**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:邢正颖

## 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](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/" \t "_blank)中的[Housing](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html" \l "housing" \t "_blank)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](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/" \t "_blank)中的[australian](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html" \l "australian" \t "_blank)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](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G。
6. 取梯度G的负方向，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**。

*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G。
6. 取梯度G的负方向，记为D。
7. 更新模型参数，。为学习率，是人为调整的超参数。
8. 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：

def grad(X,y,w):

    return -X.T\*y+X.T\*X\*w

while iter\_cnt <= iter\_time:

    iter\_cnt += 1

    w -= nita\*grad(X\_train,y\_train,w)

线性分类：

def gw(X,y,i):

    if 1-y[i]\*(ele(w.T\*matrix(X[i]).T)+b) >= 0:

        return -y[i]\*matrix(X[i]).T

    else:

        return mat(zeros((n,1)))

def gb(X,y,i):

    if 1-y[i]\*(ele(w.T\*matrix(X[i]).T)+b) >= 0:

        return -y[i]

    else :

        return 0

def gradw(X,y):

    t = mat(zeros((n,1)))

    for i in range(n):

        t += gw(X,y,i)

    t \*= c

    return w+t

def gradb(X,y):

    t = 0

    for i in range(n):

        t += gb(X,y,i)

    t \*= c

    return t

while iter\_cnt < iter\_time:

    iter\_cnt += 1

    w += nita\*(-gradw(X\_train,y\_train)/n);

    b += nita\*(-gradb(X\_train,y\_train)/n);

## 8. 选择的评估方法（留出法，交叉验证，k折交叉验证等）:

线性回归和线性分类都使用的留出法

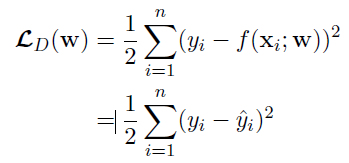
## 9. 模型参数的初始化方法:

均使用的全零初始化

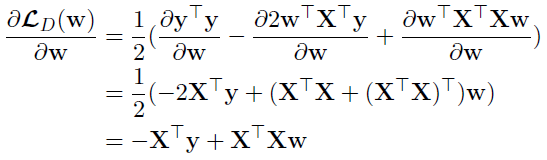
## 选择的loss函数及其导数:

线性回归：

使用的是Least squared loss

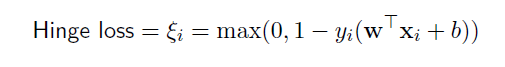


求导：

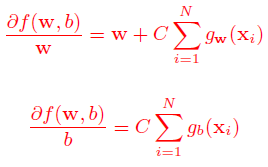


线性分类：

使用的是Hinge Loss



求导：



## 11.实验结果和曲线图:

## 超参数选择（η,epoch等）：

线性回归：η=0.0002，迭代次数为100

线性分类：η=0.14，迭代次数为100

## 评估结果（根据选择的评估方法）：

线性回归：

η=0.005 iter\_cnt = 200

函数图像和预期不符

η=0.001 iter\_cnt = 200

函数在75次迭代后收敛

η=0.0001 iter\_cnt = 200

函数在125次迭代后收敛

η=0.0003 iter\_cnt = 200

函数在75次迭代后收敛

η=0.0002 iter\_cnt = 200

函数在75次迭代后收敛

线性分类：

η=0.05 iter\_cnt = 1000

validation准确率:0.85714

图像在200多次迭代之后收敛

η = 0.45 iter\_cnt = 1000

validation准确率0.7857142857142857

图像剧烈波动，无法收敛

η=0.1 iter\_cnt = 200

validation准确率0.8571428571428571

图像收敛

η=0.1 iter\_cnt = 100

validation准确率0.8571428571428571

函数图像没有收敛，100次迭代后仍有波动

η=0.16及以上开始准确度多次迭代后仍有波动，无法收敛

η=0.14 iter\_cnt = 100

validation准确率0.8571428571428571

合适，可以收敛而且迭代次数少

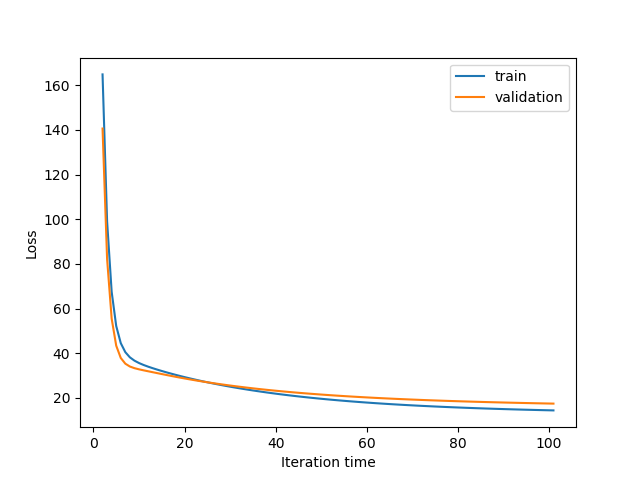
## 预测结果（最佳结果）：

线性回归：loss函数值收敛至11.34

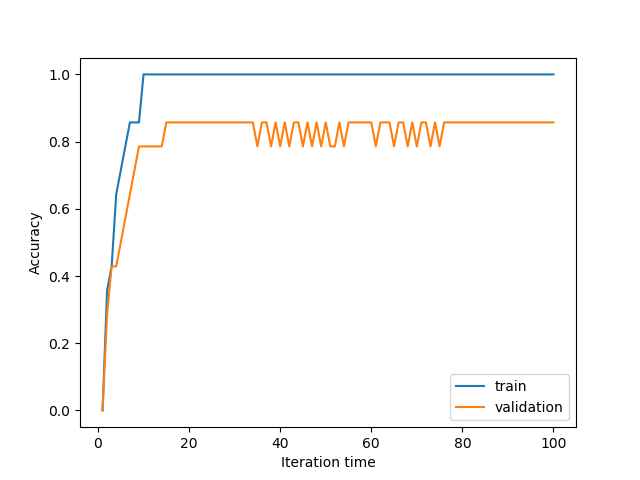
线性分类：validation数据集正确率在85.7%

## loss曲线图：

线性回归：



线性分类：



## 12.实验结果分析:

线性回归实验中，在调整到了合适的学习度之后，多次迭代之后loss函数的值收敛到一个较小的值，并且训练集和验证集的loss函数值最后非常相近，证明实验结果是正确的

线性分类实验中，多次迭代之后训练集的准确率收敛至1，验证集的准确率收敛到0.85左右，是一个较合理的值。

## 13.对比线性回归和线性分类的异同点：

相同点：都是通过线性模型去拟合训练集中的数据，并通过梯度下降等方式迭代的求其中的参数。

不同点：线性回归是想要尽量缩短训练集中的点和线性模型的距离，而线性分类则是想要用线性模型将平面内的点分为两类，并使它们和线性模型的距离最大化。两者的loss函数和参数都不相同。

## 14.实验总结：

本次实验我们通过Python模拟了线性回归与线性分类的过程，并将结果可视化，看到了loss函数随迭代次数的变化以及它在训练集和验证集上的区别，体验到了调参的过程，同时了解了Python相关库的使用方法，增强了代码能力。本次实验是相对基础的一次实验，希望可以为之后的实验打下好的基础。