

**《机器学习》课程实验报告**

**学 院 软件学院**

**专 业 软件工程**

**组 员**  **鲍淞琳**

**学 号 201530611029**

**邮 箱 2329287917@qq.com**

**指导教师** **谭明奎**

**提交日期** **2017年 12月8 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: 鲍淞琳

## 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。

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。

## 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，。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。

## 代码内容:

（线性回归）

# 预测训练集

y\_pred = x\_train\*w + b

train\_loss = 0.5\*np.mean(np.square(y\_pred - y\_train))

# 预测验证集

y\_test\_pred = x\_test\*w + b

test\_loss = 0.5\*np.mean(np.square(y\_test\_pred - y\_test))

train\_loss\_list.append(train\_loss)

test\_loss\_list.append(test\_loss)

# 参数更新

w = w - alpha\*(y\_pred - y\_train)\*x\_train/x\_train.shape[0]

b = b - alpha\*np.mean(y\_pred - y\_train)

（线性分类）

hinge = 1 - y\_train\*(x\_train\*w + b)

hinge[hinge < 0] = 0

train\_loss = np.sum(hinge) + 0.5\*p\*np.linalg.norm(w)

hinge\_temp = 1 - y\_test\*(x\_test\*w + b)

hinge\_temp[hinge\_temp < 0] = 0

test\_loss = np.sum(hinge\_temp) + 0.5\*p\*np.linalg.norm(w)

train\_loss\_list.append(train\_loss)

test\_loss\_list.append(test\_loss)

# 参数更新

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

if y\_train[j]\*(x\_train[j]\*w + b) < 1:

w = w + (alpha\*y\_train[j]\*x\_train[j]).toarray()[0]

b = b + alpha\*y\_train[j]

## **-----线性回归**

## 8. 选择的评估方法：交叉验证

## 9. 模型参数的初始化方法: 随机初始化

## 10.选择的loss函数及其导数:

## J(θ)=(1/2m)∑(i=1-m)(hθ(x(i))−y(i))^2，m代表有m组样本

dw=-x^T\*y+x^T\*x\*w

## 11.实验结果和曲线图:

## 超参数选择（η,epoch等）：η=0.001 epoch=100

## 评估结果（根据选择的评估方法）：0.78

## 预测结果（最佳结果）：

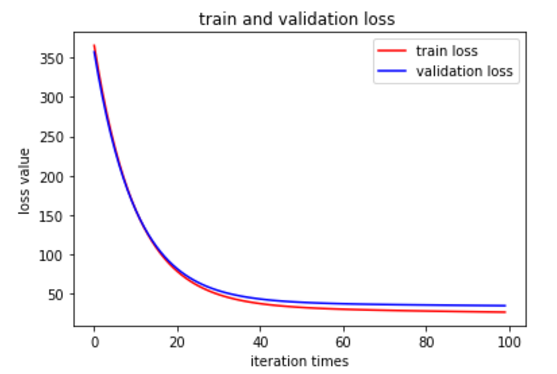
w：[-4.21320551 -2.27989647 -3.64070165 -2.01913963 -2.93504447

0.9824022 0.64606172 -4.63304395 -0.46101914 -2.898746

0.52743024 4.31217891 -4.52020341]

b：4.12919326904

## loss曲线图：



## 实验结果分析:

如果特征值之间存在线性关系就可以使用线性回归建模对其预测结果。本次测试是对一个房屋售价的数据集进行探索，并找出售价与特征值之间的关系建立回归模型，来通过此特征值来预测房价。从结果可以看出不同特征对房价的影响是不同的，有正面影响也有负面影响。随着迭代次数的增加，准确率也越来越高，第40次迭代后，结果趋于平稳。

## **-----线性分类**

## 8. 选择的评估方法：交叉验证

## 9. 模型参数的初始化方法: 随机初始化

## 10.选择的loss函数及其导数:

## J(θ)=1/2||m||^2 + ∑(i=1-m)alpha\*(y(i)(w^T\*x(i)+b)-1)，m代表有m组样本

dw=w-∑(i=1-m)alpha\*y(i)\*x(i)

db=-∑(i=1-m)alpha\*y(i)

## 11.实验结果和曲线图:

## 超参数选择（η,epoch等）：η=0.001 epoch=100

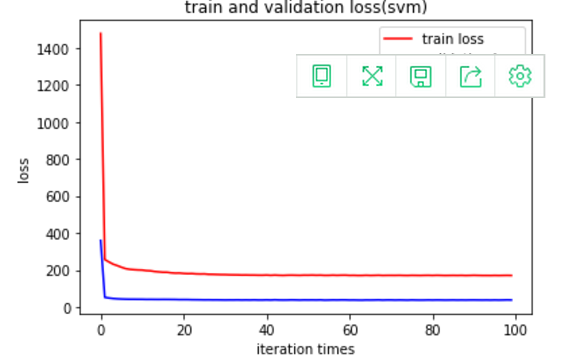
## 评估结果（根据选择的评估方法）：0.8

## 预测结果（最佳结果）：

w：[ 0.03830907 -0.13616394 -0.03070217 0.12356748 0.3145475 0.05275241 0.203288 1.10428736 0.07205143 0.2899311 -0.00376342 1.12668741 -0.50525518 0.86262353]

b：0.606541878679

## loss曲线图：



## **实验结果分析:**

使用SVM模型时loss函数下降的比较快，说明SVM训练模型效果更好一点，SVM是在拟合一个超平面，关注点集中于使得距离此分界面最近的几个点据平面的距离最大化。

## 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，就是模型的拟合。通过迭代找到最能拟合样本的参数，再进行预测。

不同点事线性分类将线性回归输出的很大范围的数，例如从负无穷到正无穷，压缩到0和1之间，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.

## 实验总结：

本次实验通过样本数据集进行模型的构建和预测。通过此次的实验更深入的了解了线性回归与线性分类，学习率、迭代次数等都会对最后的结果产生影响，选择合适的参数很重要，这样才能提高准确率。