

**《机器学习》课程实验报告**

**学 院 软件学院**

**专 业 软件工程**

**组 员**

**学 号 201530613641**

**邮 箱** [**1648321269@qq.com**](mailto:1648321269@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:张雄帅

## 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。

2. 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:

1. 线性回归和梯度下降

读取实验数据，使用sklearn库的load\_svmlight\_file函数读取数据。

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。

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。

选择Loss函数及对其求导，过程详见课件ppt。

求得所有样本对Loss函数的梯度。

取梯度的负方向，记为。

更新模型参数，。为学习率，是人为调整的超参数。

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。

重复步骤5-8若干次，画出和随迭代次数的变化图。

1. 线性分类和梯度下降

读取实验数据，使用sklearn库的load\_svmlight\_file函数读取数据。

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。

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。

选择Loss函数及对其求导，过程详见课件ppt。

求得所有样本对Loss函数的梯度。

取梯度的负方向，记为。

更新模型参数，。为学习率，是人为调整的超参数。

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。

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：

初始化：

n = X\_train.shape[0]

m = X\_train.shape[1]

W = np.zeros([m,1])

Loss function：

def loss\_func(X,y,W):

n\_ = y.shape[0]

X = X.reshape((n\_,m))

y = y.reshape((n\_,1))

W = W.reshape((m,1))

M = y - X.dot(W)

loss\_ = (M.T.dot(M)/2/n\_)[0][0]

return loss\_

倒数：

def gradient(X,y,W):

X = X.reshape((n,m))

y = y.reshape((n,1))

W = W.reshape((m,1))

return (-X.T.dot(y)+X.T.dot(X).dot(W))

逻辑回归：

n = X\_train.shape[0]

m = X\_train.shape[1]

W = np.zeros([m,1])

def loss\_func(X,y,W,C):

loss\_ = 0.0

n,m = np.shape(X)

for i in range(n):

loss\_ += max(0,1-y[i]\*W.T.dot(X[i]))

loss\_ = loss\_/n\*C

loss\_ += (W.T.dot(W)[0][0]/n/2)

return loss\_

def gradient(X,y,W,C):

n,m = X.shape

M = np.zeros([n,1])

for i in range(n):

if 1-y[i]\*(W.T.dot(X[i])) >= 0:

M[i][0] = 1

y\_ = M\*y

return (W-C\*X.T.dot(y\_))/n

## 8. 选择的评估方法（留出法，交叉验证，k折交叉验证等）:

交叉验证。

## 9. 模型参数的初始化方法:

全零初始化。

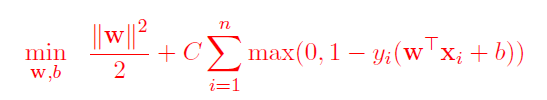
## 10.选择的loss函数及其导数:

线性回归：

Loss函数： L(w) =1/2 \* (y - Xw)T(y - Xw)

倒数： -X T y + X T Xw

线性分类：

Loss函数： 

## 11.实验结果和曲线图:

## 超参数选择（η,epoch等）：

线性回归使用η=0.0001 epoch = 100

线性分类使用η=0.0008 epoch = 1000

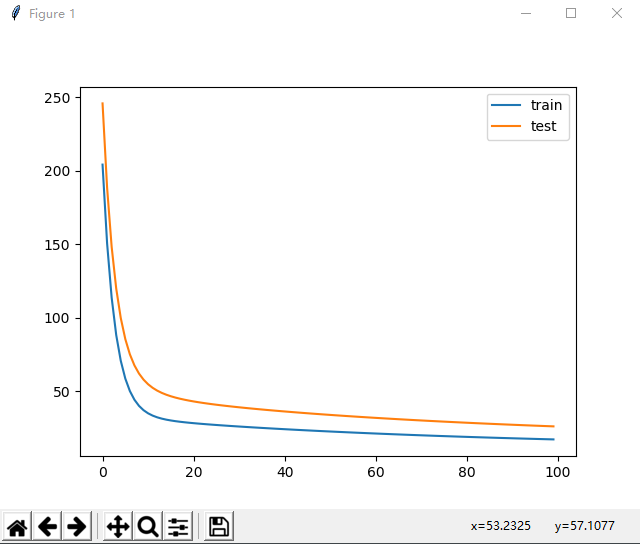
## 评估结果（根据选择的评估方法）：

## 预测结果（最佳结果）：

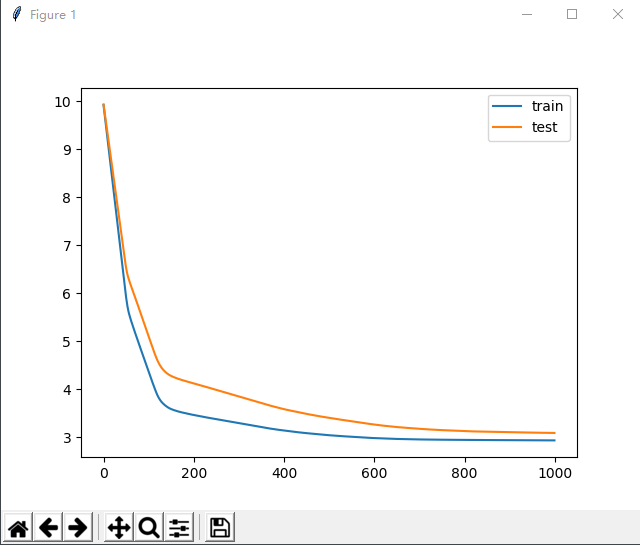
预测loss的值会收敛，而train的值将会小于test的值。

## loss曲线图：

线性回归：



线性分类



## 12.实验结果分析:

实验结果和预期一样收敛，并且train稍大于test（虽然这不是100%可能的）。线性回归由于取得学习率较大，所以收敛速度非常快。而线性分类则收敛速度较慢，并且有一定的抖动。

## 13.对比线性回归和线性分类的异同点：

同：都是求出线性参数W， 数据都可以用y = W\*x来表示。

异：线性回归求出参数使得真实值y离直线的距离尽量近，而线性分类求出参数使得真实值y在直线两侧，并且大于一个值||W||。

## 14.实验总结：

这次实验是我首次将书本上的知识变成了真正的实践，不仅仅加深了对知识的理解，还使得提升了自己的动手能力。虽然实验期间遇到了诸多了困难，但还是一一解决了，并且这些困难会提升我的知识库。