

**《机器学习》课程实验报告**

**学 院 软件学院**

**专 业 软件工程**

**组 员**  **林智远**

**学 号 201530612323**

**邮 箱 529880551@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:林智远

## 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](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/" \t "https://www.zybuluo.com/chenyaofo/note/_blank)中的[Housing](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html" \l "housing" \t "https://www.zybuluo.com/chenyaofo/note/_blank)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](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/" \t "https://www.zybuluo.com/chenyaofo/note/_blank)中的[australian](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html" \l "australian" \t "https://www.zybuluo.com/chenyaofo/note/_blank)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](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_svmlight_file.html" \t "https://www.zybuluo.com/chenyaofo/note/_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" \t "https://www.zybuluo.com/chenyaofo/note/_blank)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。
6. 取梯度的负方向，记为。
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 "https://www.zybuluo.com/chenyaofo/note/_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" \t "https://www.zybuluo.com/chenyaofo/note/_blank)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。
6. 取梯度的负方向，记为。
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:

（针对线性回归和线性分类分别填写8-12内容）

线性回归：

for i in range(0, maxIteration):

y\_predict1 = np.dot(W.transpose(),X\_train.transpose())

loss\_train = y\_predict1-y\_train

Ltrain.append(np.dot(loss\_train,loss\_train.transpose())/(2\*m\_train))

y\_predict2=np.dot(W.transpose(),X\_test.transpose())

loss\_test= y\_predict2-y\_test

Lvalidation.append(np.dot(loss\_test.transpose(),loss\_test)/(2\*m\_test))

gradient = np.dot(X\_train.transpose(), loss\_train.transpose()) /m\_train

W = W - alpha \* gradient

线性分类：

for i in range(0, maxIteration):

ltrain=0

ltest=0

g=0

for j in range(m\_train):

y\_predict1= np.dot(W.transpose(),X\_train[j].transpose())

y\_predict11= np.dot(y\_predict1,y\_train[j])

ltrain+=max(0,1-y\_predict11)

Ltrain.append(ltrain)

for j in range(m\_test):

y\_predict2=np.dot(W.transpose(),X\_test[j].transpose())

y\_predict22= np.dot(y\_predict2,y\_test[j])

ltest+=max(0,1-y\_predict22)

Lvalidation.append(ltest)

for j in range(m\_train):

y\_predict111=np.dot(W.transpose(),X\_train[j].transpose())

if np.dot(y\_train[j],y\_predict111)<=1:

g -= y\_train[j]\*X\_train[j]

gradient=W+0.9\*g;

W = W - alpha \* gradient

## 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：交叉验证

线性分类：交叉验证

## 模型参数的初始化方法:

线性回归：全为0

线性分类：全为0

## 选择的loss函数及其导数:

线性回归：Least Squared Loss

线性分类：Hinge Loss

## 11.实验结果和曲线图:

## 超参数选择（η,epoch等）：

线性回归：η=0.1，epoch=100

线性回归：η=0.001，epoch=100

## 评估结果（根据选择的评估方法）：不断调整超参数η，找出最优解

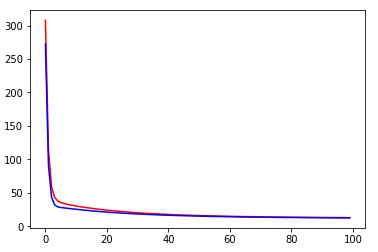
## 预测结果（最佳结果）：

线性回归：η=0.1

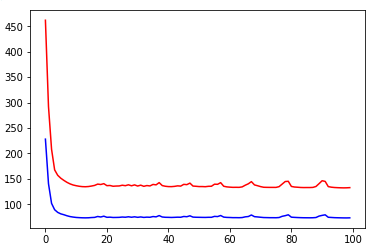
线性回归：η=0.001

## loss曲线图：

线性回归：



线性分类：



## 实验结果分析:

线性回归：随着迭代次数的增加，训练集和验证集的loss都减少，最后趋于稳定。

线性分类：随着迭代次数的增加，训练集和验证集的loss都减少，最后趋于稳定。但训练集的Hinge Loss始终大于验证集的Hinge Loss，因为训练集的样本数量比较大。

## 对比线性回归和线性分类的异同点：

同：随着迭代次数的增加，训练集和验证集的loss都减少，，最后趋于稳定。

异：线性分类中训练集的Hinge Loss始终大于验证集的Hinge Loss，因为训练集的样本数量比较大。

## 实验总结：

通过这个实验，我进一步理解线性回归和梯度下降的原理，也体会到了优化和调参的过程。