

**《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G。
6. 取梯度的负方向，记为D。
7. 更新模型参数，W（t)=W(t-1)+nD。n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G。
6. 取梯度的负方向，记为D。
7. 更新模型参数，W（t)=W(t-1)+nD。n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：

alpha = 0.41

maxIteration = 1000

Iteration=range(0,maxIteration)

Ltrain=[]

Lvalidation=[]

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：

alpha = 0.02

maxIteration = 1000

Iteration=range(0,maxIteration)

Ltrain=[]

Lvalidation=[]

for i in range(0, maxIteration):

ltrain=0

ltest=0

g=0

count=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)

lltrain=np.dot(W.transpose(),W)/2+(0.9\*ltrain)/m\_train

Ltrain.append(lltrain)

for j in range(m\_test):

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

if y\_pre>=0 and y\_test[j]>=0:

count+=1

if y\_pre<0 and y\_test[j]<0:

count+=1

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)

accurancy=count/m\_test

lltest=np.dot(W.transpose(),W)/2+(0.9\*ltest)/m\_test

Lvalidation.append(lltest)

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)/m\_train;

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:J=(y-Xw)T\*(y-Xw)/2\*m

∂J(w)/∂w=XT\*(Xw-y)/m

线性分类:J=||w||^2/2+C\*Sum 1\_m max(0,1−yi(wTxi +b))

∂J(w)/∂w=w−C\*yxi if 1−yi(wTxi+b)>=0

∂J(w)/∂w=w if 1−yi(wTxi+b)<0

## 实验结果和曲线图:

线性回归：

## 超参数选择（η,epoch等）：

η=0.41，epoch=1000

评估结果（根据选择的评估方法）：

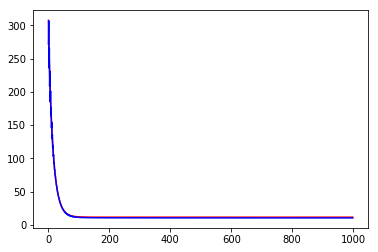
η=0.4，loss函数数次迭代后趋于平缓，最小值为10.3670281084

η>=0.42，loss函数呈飙升趋势

η=0.41，loss函数数次迭代后趋于平缓，最小值10.3666312135，且整体的loss比η=0.4时要低

## 预测结果（最佳结果）：η=0.41时验证集loss的最小值为10.3666312135

## loss曲线图：



线性分类：

## 超参数选择（η,epoch等）：

η=0.02，epoch=1000

评估结果（根据选择的评估方法）

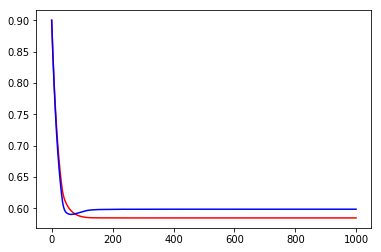
η=0.01，loss函数数次迭代后平缓，最小值0.59033306069

η=0.02，loss函数数次迭代后平缓，最小值为0.59028698303

η=0.03，loss函数数次迭代后平缓，最小值0.590291649028

预测结果（最佳结果）：η=0.02，验证集loss的最小值为0.59028698303，最后一次迭代准确率为0.8508771929824561

## loss曲线图：



## 实验结果分析:

线性回归：验证集和训练集的平均loss函数值最后都平缓，验证集的平均loss函数值最小值10.3666312135

线性分类：验证集和训练集的平均loss函数值最后都平缓，验证集的平均loss函数值最小值0.59028698303，最后一次迭代准确率为0.8464912280701754

，说明该数据集线性可分，分类效果良好。

## 对比线性回归和线性分类的异同点：

相同：都使用线性函数，小规模数据集都可以用梯度下降法求解

不同：损失函数不同，数据集标签不同

## 实验总结：

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