

**《机器学习》课程实验报告**

**学 院 软件学院**

**专 业 软件工程**

**组 员 李焕童**

**学 号 201530611944**

**邮 箱 1779414343@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:李焕童

## 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/)中的[Housing](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html#housing)数据scaled版本，包含506个样本，每个样本有13个属性，无数据缺失重复。

线性分类使用的是[LIBSVM Data](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/)中的[australian](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#australian)数据scaled版本，包含690个样本，每个样本有14 个属性，无数据缺失重复。

## 6. 实验步骤:

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)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.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。

## 7. 代码内容:

线性回归:

Ltrain=[]

Lvalidation=[]

for i in range(0, epoch):

hypothesis = np.dot(X\_train,W)

loss\_train = hypothesis - y\_train

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

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

loss\_test= y\_predict-y\_test

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

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

W = W - η \* gradient

线性分类:

def g(x,y,i):

return 1-np.dot(y[i],np.dot(W.transpose(),x[i].transpose()))

for i in range(0,epoch):

loss\_train = 0

for j in range(m\_train):

loss\_train += C\*max(0,g(X\_train,y\_train,j))

loss\_train /=m\_train

loss\_train += np.dot(W.transpose(),W)/2

Ltrain.append(loss\_train)

loss\_test= 0

for k in range(m\_test):

loss\_test += C\*max(0,g(X\_test,y\_test,k))

loss\_test /=m\_test

loss\_test += np.dot(W.transpose(),W)/2

Lvalidation.append(loss\_test)

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

y\_predict[y\_predict>0]=1

y\_predict[y\_predict<0]=-1

count=0

for l in range(len(y\_predict)):

if y\_predict[l]==y\_test[l]:

count +=1

accurancy.append(count/m\_test)

gradient=0

for m in range(m\_train):

if g(X\_train,y\_train,m)>=0:

gradient -=C\*np.dot(y\_train[m], X\_train[m].transpose())

gradient /=m\_train

gradient +=W

W -=η\*gradient

（针对线性回归和线性分类分别填写8-12内容）

## 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: hold -out cross validation简单交叉验证

线性分类: hold -out cross validation简单交叉验证

## 9. 模型参数的初始化方法:

线性回归: 全零初始化

线性分类: 全零初始化

## 10.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)))/m

∂J(w)/∂w=w−(C\*yixi)/m if 1−yi(wTxi+b)>=0

∂J(w)/∂w=w if 1−yi(wTxi+b)<0

## 11.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）：

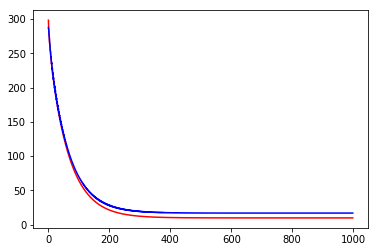
η**=0.4，loss函数数次迭代后可收敛，在第**999次迭代时已达到最小值16.7252723626

η**=0.41，loss函数数次迭代后可收敛，在第**703次迭代时已达到最小值16.720937745

η>**=0.42，loss函数不可收敛**

## 预测结果（最佳结果）：η=0.41,迭代1000次验证集loss的最小值为16.720937745

## loss曲线图：



线性分类:

## 超参数选择（η,epoch等）：η=0.112 epoch=1000 C=0.9

## 评估结果（根据选择的评估方法）：

η>**=0.2，loss函数不能平滑收敛**

η**=0.113，loss函数数次迭代后可收敛，**在第297次迭代时已达到最小值0.57542742442准确率最大值： 0.8840579710144928

η**=0.112，loss函数数次迭代后可收敛，**在第286

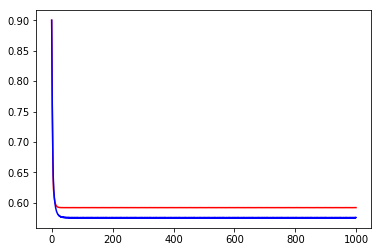
次迭代时已达到最小值0.575420683566准确率最大值： 0.8840579710144928

η**=0.111，loss函数数次迭代后可收敛，**在第372

次迭代时已达到最小值0.575425172036准确率最大值： 0.8840579710144928

## 预测结果（最佳结果）：η=0.112，1000次迭代验证集loss的最小值为0.575420683566准确率最大值： 0.8840579710144928

## loss曲线图：



## 12.实验结果分析:

线性回归：验证集和训练集的平均loss函数值都在收敛，验证集的平均loss函数值在703次迭代后达到最小值16.720937745

线性分类：验证集和训练集的平均loss函数值都在收敛，验证集的平均loss函数值在286次迭代后达到最小值0.575420683566，准确率在19次迭代后达到最大值0.8840579710144928，说明该数据集线性可分，分类效果良好。

## 13.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

## 14.实验总结：

通过这次实验，我清楚地掌握了线性回归、线性分类的原理和公式推导，初步使用了梯度下降算法，并对调参有了一定的认识。