

**《机器学习》课程实验报告**

**学 院 软件学院**

**专 业 软件工程**

**组 员**  **吴伟民**

**学 号 201530613061**

**邮 箱 2727709668@qq.com**

**指导教师**  **吴庆耀**

**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:吴伟民

## 4. 实验目的:

进一步理解线性回归和梯度下降的原理。

在小规模数据集上实践。

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中的Housing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中的australian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:



## 代码内容:

线性回归与梯度下降

class LinearRegressionByMyself(object):

def \_\_init\_\_(self,Learning\_rate=0.001,epoch=8):

self.Learning\_rate=Learning\_rate

self.epoch=epoch

def fit(self,x,y):

self.w=np.zeros(1+x.shape[1])

self.cost\_list=[]

self.row=x.shape[0]

for i in range(self.epoch):

output=self.Regression\_input(x)

error=(y-output)

self.w[1:]+=self.Learning\_rate\*x.T.dot(error)

self.w[0]+=self.Learning\_rate\*error.sum()

cost=(error\*\*2).sum()/2.0

self.cost\_list.append(cost)

return self

def Regression\_input(self,x):

#f=self.w[0]+self.w[1:]

#print('self.w[1:]:',self.w[1:])

#print('f:',f)

#print('self.w[0]:',self.w[0])

#print('x\*w:',x[0].dot(self.w[1:]))

return x.dot(self.w[1:])+self.w[0]

线性分类与梯度下降

class LinearClassification(object):

def \_\_init\_\_(self,Learning\_rate=0.001,threshold=0.0,epoch=8):

self.Learning\_rate=Learning\_rate

self.epoch=epoch

def fit(self,x,y,threshold,C):

self.w=np.zeros((1,1+x.shape[1]))

self.cost\_list=[]

self.threshold=threshold

self.C=C

#对每个x进行计算

for i in range(self.epoch):

cost=0

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

output=self.Classification\_input(x[j,:],self.threshold)

if(y[j]!=output):

#计算y\*x

t=self.C\*y[j]\*x[j,:]

#计算梯度

dw=self.w[0,1:]-t

#计算cost

cost+=((1-y[j]\*output)\*self.C)

#更新w

self.w[0,1:]=self.w[0,1:]+self.Learning\_rate\*dw

self.w[0,0]=self.w[0,0]+self.Learning\_rate\*(self.w[0,0]-self.C\*y[j])

self.w1=self.w.T

cost+=((self.w1\*self.w.T).sum())/2

self.cost\_list.append(cost)

return self

#计算每个f(X)

def Classification\_input(self,x,threshold):

f=x\*self.w[0,1:]+self.w[0,0]

if(f>=threshold):

return 1

else:

return -1

（针对线性回归和线性分类分别填写8-12内容）

## 选择的评估方法（留出法，交叉验证，k折交叉验证等）:

## 留一验证法

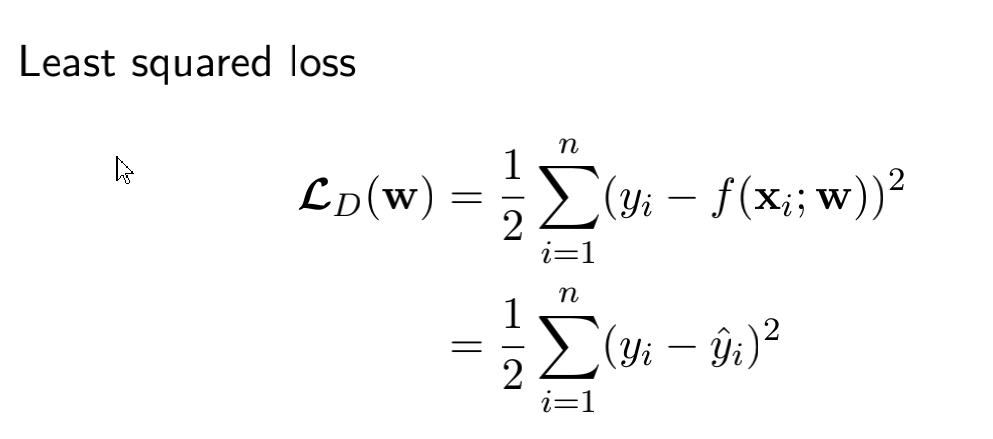
## 模型参数的初始化方法:

全零初始化

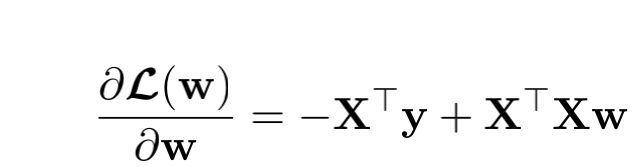
## 选择的loss函数及其导数:

线性回归与梯度下降

Loss函数

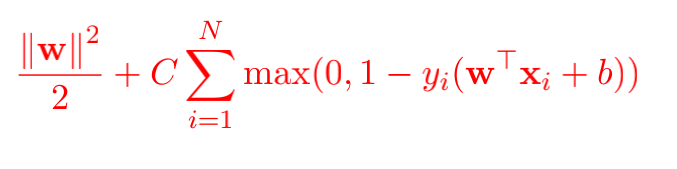


导数



线性分类与梯度下降

Loss函数



导数

2

## 11.实验结果和曲线图:

## 超参数选择（η,epoch等）：

线性回归

η=0.001

epoch=8

线性分类

η=0.001

epoch=8

C=0.9

## 评估结果（根据选择的评估方法）：

线性回归

η=0.001

epoch=8

线性分类

η=0.001

epoch=8

C=0.9

## 预测结果（最佳结果）：

线性回归

η=0.001

epoch=8

线性分类

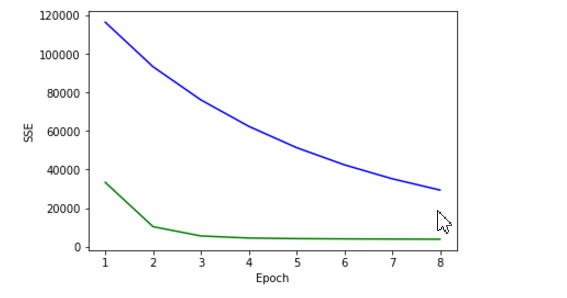
η=0.001

epoch=8

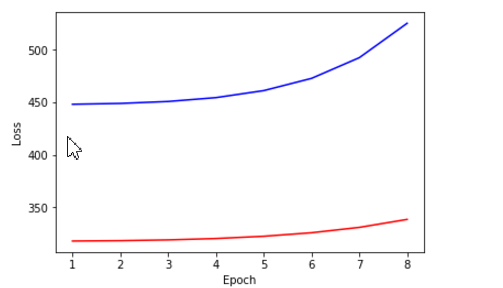
C=0.9

## 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增加，损失值变小，线性分类则是相反

## 对比线性回归和线性分类的异同点：

线性回归的幅度相对比线性分类的幅度大，而且前者会随迭代次数增加而减少损失值，后者则相反

相同点是他们模型函数都是线性的

## 14.实验总结：

通过动手操作并且自己独立分析，对这些算法有更深入的了解，但是需要花更多时间去调节好参数