

**《机器学习》课程实验报告**

**学 院 软件学院**

**专 业 软件工程**

**组 员**  **张毓嘉**

**学 号 201530613689**

**邮 箱 3290196208@qq.com**

**指导教师**  **吴庆耀**

**提交日期** **2017年 12 月 07 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08日

## 3. 报告人:张毓嘉

## 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:

*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 "_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 "_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 "_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 "_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）

## 8. 选择的评估方法（留出法，交叉验证，k折交叉验证等）:

留出法

if abs(loss - train\_new\_loss) > 0.00001:

loss = train\_new\_loss

else:

break

train\_loss.append(loss)

test\_loss.append(test\_new\_loss)

count += 1

if count>50:

break

## 9. 模型参数的初始化方法:

线性回归：

def get\_data():

data = load\_svmlight\_file("housing\_scale.txt")

return data[0], data[1]

x,y=get\_data()

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x,y, test\_size = 0.33,random\_state=42)

线性分类：

def get\_data():

data = load\_svmlight\_file("australian\_scale.txt")

return data[0], data[1]

x,y=get\_data()

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x,y, test\_size = 0.33,random\_state=42)

## 10.选择的loss函数及其导数:

线性回归：

loss = 1/(2\*\*feature) \* sum([(f(x\_train[i,:],w,b) - y\_train[i])\*\*2 for i in range(feature)])

gradient\_averange = sum([(f(x\_train[k,:],w,b)-y\_train[k])\*x\_train[k,i] for k in range (x\_train\_len)])/x\_train\_len

线性分类：

p=0.8

loss = sum([w[i] \*\* 2 for i in range(feature)]) + p / x\_train\_len\* sum([max(0, 1 - y\_train[i] \* f(x\_train[i, :], w, b)) for i in range(x\_train\_len)])

gradient\_averange = sum([((y\_train[k] < 1) \* (-y\_train[k]) \* x\_train[k, i]) for k in range(x\_train\_len)]) / x\_train\_len

w[i] -= gradient\_averange \* lr

## 11.实验结果和曲线图:

## 超参数选择（η,epoch等）：

线性回归η为0.02 epoch为 50

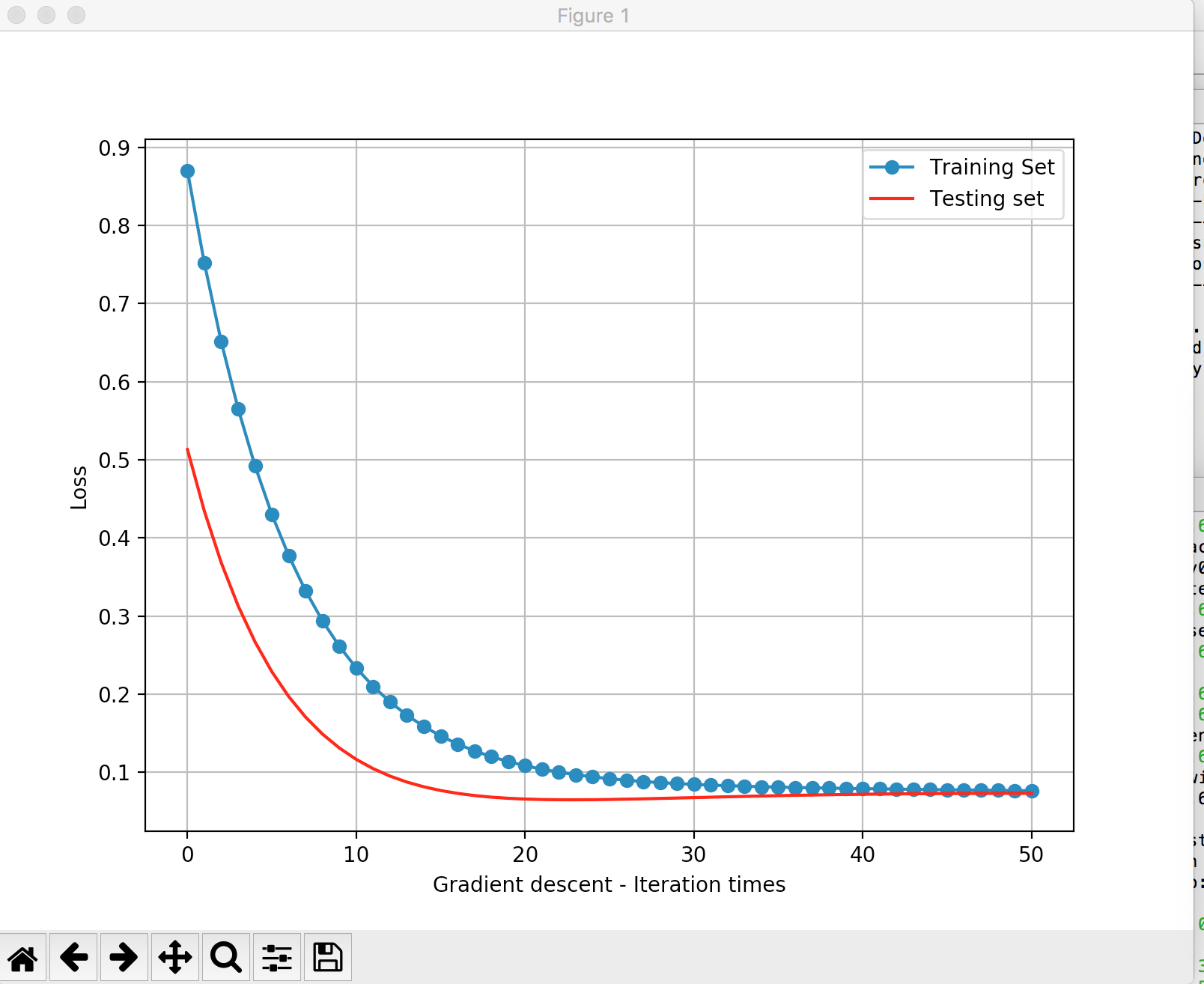
线性分类η为0.01 epoch为60

## 评估结果（根据选择的评估方法）：

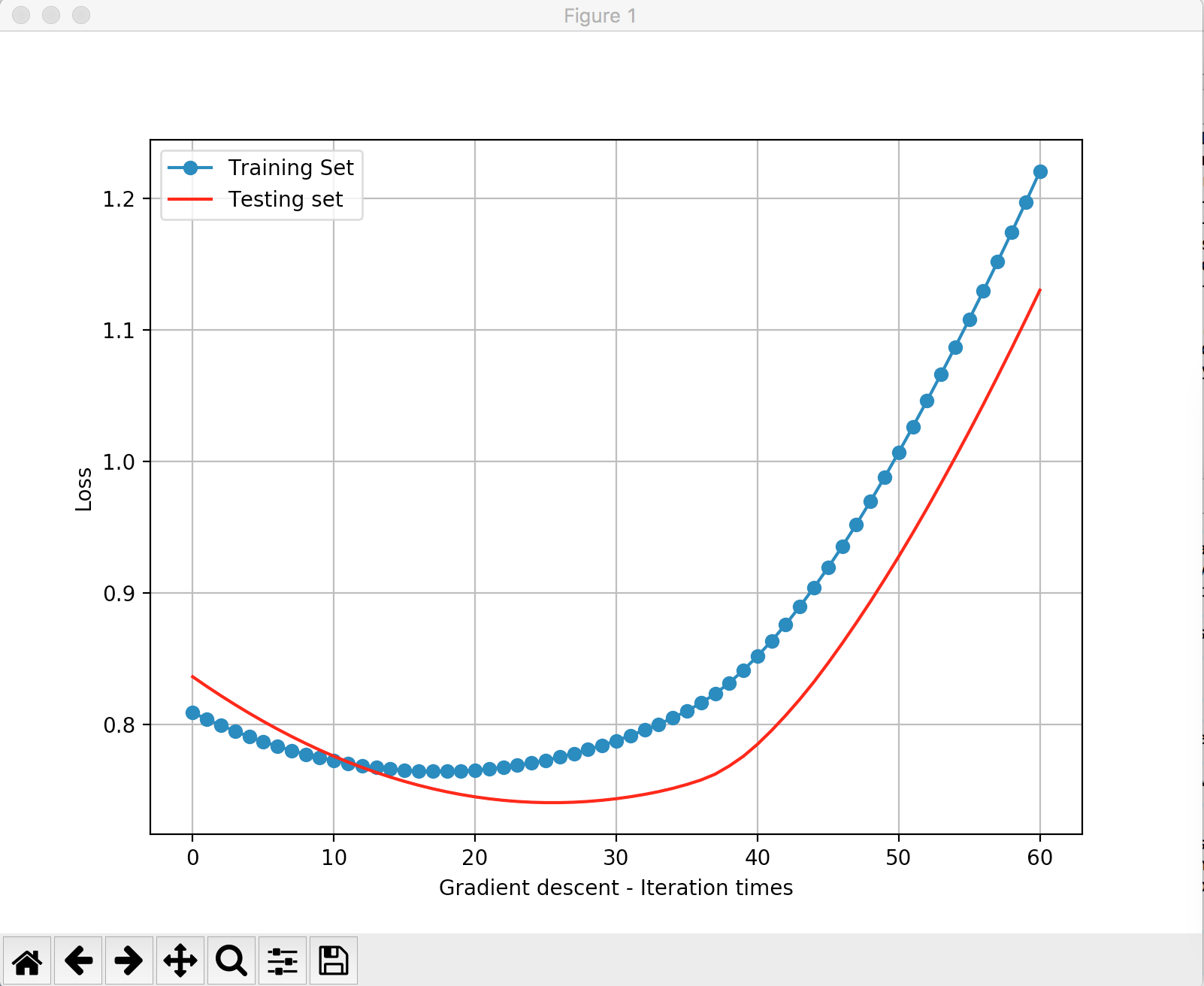
评估选择loss－testing的最低点

## 预测结果（最佳结果）：

## loss曲线图：

线性回归：

线性分类：



## 12.实验结果分析:

## 13.对比线性回归和线性分类的异同点：

说两个问题本质上都是一致的，就是模型的拟合（匹配）。 但是分类问题的 y值(也称为label), 更离散化一些. 而且， 同一个y值可能对应着一大批的x, 这些x是具有一定范围的。

所以分类问题更多的是 (一定区域的一些x) 对应 着 (一个y). 而回归问题的模型更倾向于 (很小区域内的x，或者一般是一个x) 对应着 (一个y).

## 14.实验总结：

在整个实验过程中，我们需要：

第一步：选取合理的线性模型

第二步：计算loss函数

第三部：进行梯度下降

在计算loss函数的时因为线性回归和线性分类在原理上有不同， 所以采用不同的loss函数。