

**《机器学习》课程实验报告**

**学 院 软件学院**

**专 业 软件工程**

**组 员**  **李彩利**

**学 号 201530611920**

**邮 箱 2561170837@qq.com**

**指导教师**  **吴庆耀**

**提交日期** **2017年 12月 9 日**

## 1. 实验题目: 线性回归和梯度下降

## 2. 实验时间：2017年 12 月 2 日

## 3. 报告人:李彩利

## 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

## 5. 数据集以及数据分析：线性回归使用的是LIBSVMData中的Housing数据，包含506个样本，每个样本有13个属性。将其切分为训练集和验证集。

## 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" \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)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。
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:

（针对线性回归和线性分类分别填写8-12内容）

# -\*- coding=utf8 -\*-

import math;

from sklearn.externals.joblib import Memory

from sklearn.datasets import load\_svmlight\_file

from sklearn.model\_selection import train\_test\_split

from numpy.core.fromnumeric import shape

mem = Memory("./mycache")

@mem.cache

def get\_data():

data = load\_svmlight\_file("F:\BaiduNetdiskDownload\machinelearning\housing\_scale.txt")

return data[0], data[1]

def sum\_of\_gradient(x, y, thetas):

"""计算梯度向量，参数分别是x和y轴点坐标数据以及方程参数"""

m = 338;

grad0 = 1.0 / m \* sum([(thetas[0] + thetas[1] \* x[i] - y[i]) for i in range(m)])

grad1 = 1.0 / m \* sum([(thetas[0] + thetas[1] \* x[i] - y[i]) \* x[i] for i in range(m)])

#grad0 = 1.0 / m \* (thetas[0] + thetas[1] \* x - y)

#grad1 = 1.0 / m \* (thetas[0] + thetas[1] \* x - y) \* x

return [grad0, grad1];

def step(thetas, direction, step\_size):

"""move step\_size in the direction from thetas"""

return [thetas\_i + step\_size \* direction\_i

for thetas\_i, direction\_i in zip(thetas, direction)]

def distance(v, w):

"""两点的距离"""

return math.sqrt(squared\_distance(v, w))

def squared\_distance(v, w):

vector\_subtract = [v\_i - w\_i for v\_i, w\_i in zip(v, w)]

return sum(vector\_subtract\_i \* vector\_subtract\_i for vector\_subtract\_i, vector\_subtract\_i

in zip(vector\_subtract, vector\_subtract))

def gradient\_descent(stepSize, x, y, tolerance=0.000000001, max\_iter=10000):

"""梯度下降"""

iter = 0

# initial theta

thetas = [0, 0];

# Iterate Loop

while True:

gradient = sum\_of\_gradient(x, y, thetas);

next\_thetas = step(thetas, gradient, stepSize);

if distance(next\_thetas, thetas) < tolerance: # stop if we're converging

break

thetas = next\_thetas # continue if we're not

iter += 1 # update iter

if iter == max\_iter:

print 'Max iteractions exceeded!'

break;

return thetas

x, y = get\_data()

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

stepSize = 0.001;

t0, t1 = gradient\_descent(-stepSize, X\_train, y\_train);

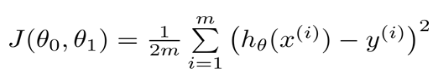
print t0, " ", t1;

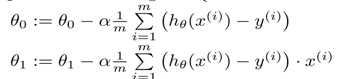
print X\_train;

## 8. 选择的评估方法（留出法，交叉验证，k折交叉验证等）:

## 9. 模型参数的初始化方法:全零初始化

## 10.选择的loss函数及其导数:





## 11.实验结果和曲线图:

## 超参数选择（η,epoch等）：0.001

## 评估结果（根据选择的评估方法）：

## 预测结果（最佳结果）：

## loss曲线图：

## 12.实验结果分析:成曲线收敛

## 13.对比线性回归和线性分类的异同点：线性回归和线性分类大同小异，在梯度下降的前面过程都是差不多的，但是线性回归返回的是连续值，而线性分类返回的是离散值

## 14.实验总结：了解了线性回归的原理和代码书写，根据实验结果调参后得到最佳的曲线