实验介绍

1.实验内容

本实验通过运用线性回归方法预测鲍鱼年龄。

2.实验目标

通过本实验掌握并实现基于线性回归方法预测鲍鱼年龄。

3.实验知识点

- 线性回归

4.实验环境

- python 3.6.5

5.预备知识

- Python编程基础

数据准备

点击屏幕右上方的下载实验数据模块，选择下载linear_regression_abalone.tgz到指定目录下，然后再依次选择点击上方的File->Open->Upload,上传刚才下载的数据集压缩包，再使用如下命令解压：

In [1]:
```
!tar -zxvf linear_regression_abalone.tgz
```

```
linear_regression_abalone/
linear_regression_abalone/abalone.txt
```

本实验的数据集目录为：linear_regression_abalone，数据集文件为目录下的abalone.txt文件。

线性回归算法可以对真实数据进行预测。

该数据记录了鲍鱼的年龄，及其他的特征量。鲍鱼年龄可由鲍鱼壳的层数推算得到。

我们先来看下数据格式。由于包含多个数据特征，我们无法在二维平面下展示数据的分布。

In [1]:
```
!type linear_regression_abalone / abalone.txt
```

```
命令语法不正确。
```

我们可以看到数据集中包含多个数据特征，但我们只需要知道最后一列为鲍鱼的年龄，即我们的y值。我们将上面的数据分别应用于标准线性回归和局部加权线性回归。

【练习】代码实现

In [1]:
```
#导入实验需要的库numpy、matplotlib.pyplot
#!/usr/bin/env python
#-*- coding:utf-8 -*-
from numpy import *
```

实现文件上传函数，传入filename文件路径参数，返回模型训练需要的数据矩阵和目标值向量。

In [13]:
```
#实验文件上传函数
def loadDataSet(fileName):
    """ 打开一个用tab键分隔的文本文件

    :param fileName: -文件名
    :return: dataMat -数据矩阵 labelMat -目标值向量
    """
    #得到列数，不包括最后一列，默认最后一列值为目标值
    numFeat = len(open(fileName).readline().split('\t')) - 1
    #定义初始数据集
    dataMat = []
    labelMat = []
    #读取文件内容
    fr = open(fileName)
    #构造数据集
    for line in fr.readlines():
        date = []
        temp = line.strip().split('\t')
        for i in range(numFeat):
            date.append(float(temp[i]))
        dataMat.append(date)
        labelMat.append(float(temp[-1]))
    return dataMat, labelMat
```

代码实现计算最佳拟合直线，通过给定的参数x、y。来计算最佳的w值，代码实现最小二乘法。

In [28]:
```
def standRegres(xArr, yArr):
    """
    计算最佳拟合直线

        :param xArr: 给定的输入值
        :param yArr: 给定的输出值
        :return: 回归系数
    """
    #将数据保存到矩阵中
    xArr = asmatrix(xArr)
    yArr = asmatrix(yArr).T
    #计算x.T *x
    data = xArr.T * xArr
    #使用linalg.det()方法来判断它的行列式是否为零，即是否可逆
    if not linalg.det(data):
        raise ValueError
    #使用最小二乘法计算w值
    ws = data.I * (xArr.T * yArr)  #使用最小二乘法计算w值 矩阵.I逆矩阵
    return ws
```

代码实现计算回归系数。

In [ ]:
```
"""
计算回归系数
parameters:
    testPoint -待预测数据
    xArr -给定输入值
    yArr -给定输出值
    k -高斯核的k值，决定对附近的点赋予多大的权重
return:
    testPoint * ws -回归系数的估计值
"""


def lwlr(testPoint, xArr, yArr, k=1.0):
    ### Start Code Here ###
    #读入数据到矩阵

    #获取样本点个数

    #创建对角权重矩阵，该矩阵为方阵，阶数为样本点个数

    #遍历整个数据集
    #计算每个样本点对应的权重值，随着样本点与待预测点距离的递增，权重将以指数级衰减

    #判断矩阵是否可逆

    #计算归回归系数

    ### End Code Here ###

    return testPoint * ws
```

In [23]:
```
"""
测试函数
parameters:
    testArr -测试数据集
    xArr -给定输入值
    yArr -给定输出值
    k -高斯核的k值
return:
    yHat -预测值
"""


def lwlrTest(testArr, xArr, yArr, k=1.0):
    m = shape(xArr)[0]
    yHat = zeros(m)
    for i in range(m):
        yHat[i] = lwlr(testArr[i], xArr, yArr, k)
    return yHat


"""
计算预测误差的平方和
parameters:
    yArr -给定y值
    yHatArr -预测y值
return:
    ((yArr-yHatArr)**2).sum() -误差矩阵
"""


def rssError(yArr, yHatArr):
    return ((yArr - yHatArr) ** 2).sum()
```

In [29]:
```
if __name__ == '__main__':
    abX, abY = loadDataSet('linear_regression_abalone/abalone.txt')
    # yHat01 = lwlrTest(abX[0:99],abX[0:99],abY[0:99],0.1)
    # yHat1 = lwlrTest(abX[0:99],abX[0:99],abY[0:99],1)
    # yHat10 = lwlrTest(abX[0:99],abX[0:99],abY[0:99],10)
    # print("使用局部加权线性回归预测误差：")
    # print("核为0.1时：",rssError(abY[0:99],yHat01.T))
    # print("核为1时：",rssError(abY[0:99],yHat1.T))
    # print("核为10时：",rssError(abY[0:99],yHat10.T))
    # yHat01 = lwlrTest(abX[100:199],abX[0:99],abY[0:99],0.1)
    # yHat1 = lwlrTest(abX[100:199],abX[0:99],abY[0:99],1)
    # yHat10 = lwlrTest(abX[100:199],abX[0:99],abY[0:99],10)
    # print("使用局部加权线性回归预测误差在新数据上的表现：")
    # print("核为0.1时：",rssError(abY[100:199],yHat01.T))
    # print("核为1时：",rssError(abY[100:199],yHat1.T))
    # print("核为10时：",rssError(abY[100:199],yHat10.T))
    ws = standRegres(abX[0:99], abY[0:99])
    yHat = mat(abX[100:199]) * ws
    print("使用标准线性回归预测误差为：", rssError(abY[100:199], yHat.T.A))
```

```
使用标准线性回归预测误差为： 518.6363153248552
```

可以看到，当k=0.1时，训练集误差小，但是应用于新的数据集之后，误差反而变大了。这就是经常说道的过拟合现象。我们训练的模型，我们要保证测试集准确率高，这样训练出的模型才可以应用于新的数据，也就是要加强模型的普适性。可以看到，当k=1时，局部加权线性回归和简单的线性回归得到的效果差不多。这也表明一点，必须在未知数据上比较效果才能选取到最佳模型。那么最佳的核大小是10吗？或许是，但如果想得到更好的效果，应该用10个不同的样本集做10次测试来比较结果。

本次实验展示了如何使用局部加权线性回归来构建模型，可以得到比普通线性回归更好的效果。局部加权线性回归的问题在于，每次必须在整个数据集上运行。也就是说为了做出预测，必须保存所有的训练数据。

实验总结

通过本实验掌握并实现基于线性回归方法预测鲍鱼年龄。