厦门大学
XIAMEN UNIVERSITY

信息学院

机器学习
——作业 3

姓　　名: 黄静

年　　级: 2020 级

学　　号: 23020201153756

完成日期: 2021 年 4 月 7 日

一、实验要求

在第一次作业 ML = lambda 的代码框架上编程，给出详细的实验报告，并附关键代码解析。第一题要求在 Polynomial Curve Fitting 中 M=3 时，在算法中加入 L2 约束条件，以$\frac{||w_\lambda||}{||w_\infty||}$为横坐标，$w_\lambda$为纵坐标画出 Regularization Path 曲线；第二题要求对给定图片拟合图片中的形状。

二、实验环境

本次实验采用 Visual Studio Code+numpy1.18.1 版本，使用 Python 作为实现语言。

三、实验过程

- **第一题：Regularization Path 曲线**

【1】数据集的产生

本次实验的数据集中于均匀分布在区间[0,1]之间，且每个数据随机加上服从高斯分布的噪声，实际拟合的函数为 sin(2πx)，即每个数据的实际值为在该点的 sin(2πx)的值加上随机噪声值。生成数据的代码如下：

【2】初始化模型参数

本实验中定义多项式系数为 3，即 W 系数共有四个值；初始化λ=1，同时对输入数据进行处理，使其变为多项式参数数据，代码如下：

```
x_train, y_train = create_toy_data(func, 10, 0.25)
x_test = np.linspace(0, 1, 100)
y_test = func(x_test)
alpha = 1
degree = 3
feature = PolynomialFeature(degree=degree)
X_train = feature.transform(x_train)
X_test = feature.transform(x_test)
# print(X_train,x_train)
```

【3】计算没有正则化时的 W 及其 2-范数

```
# 计算没有正则化时的w=(X^TX)^-1X^TY及其2-范数
w_infinite = inv(np.dot(np.transpose(X_train),X_train))
w_infinite = np.dot(w_infinite,np.transpose(X_train))
w_infinite = np.dot(w_infinite,y_train)
w_norm = np.linalg.norm(w_infinite, ord=2)
print(w_infinite)
```

计算结果如下：

W = [-0.0921 13.0616 -38.1135 25.4817]，W 的 2-范数为 47.6715。

```
[ -0.09212505  13.06165192 -38.11355372  25.48172699]
47.671549435436816
```

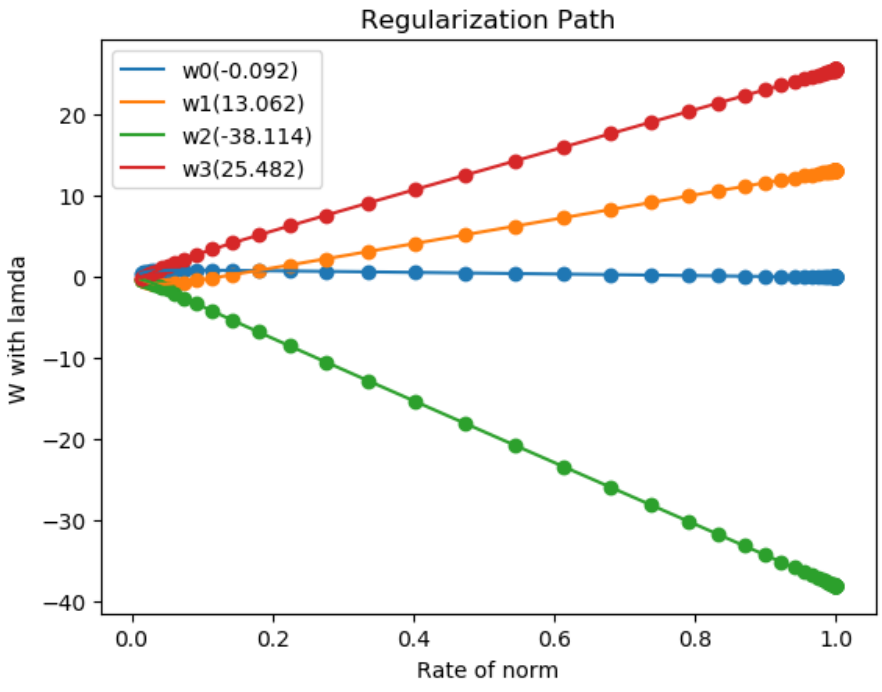
【4】求解在 lamda 值不同的情况下 w 的值

使用 rate 表示两个范数的比值，定义λ从大到小一共计算 100 次，初始值为 1，每次减小为上一次的 0.75，经过计算后得到w_λ结果的矩阵，大小为 100*4，以及范数比值 rate 矩阵，大小为 100*1。代码如下：

```python
# 求解在lamda值不同的情况下w的值
w = []
rate = []                # rate为横坐标
learning_time = 100      # lamda的变化次数，即坐标个数
for i in range(learning_time):
    model = RidgeRegression(alpha=alpha)
    model.fit(X_train, y_train)
    y = model.predict(X_test)
    error = np.mean(np.square(y - y_test))
    # print("alpha=" + str(round(alpha,6)) + ":mse=" + str(round(error,6)))
    # print(model.w)
    # 求解横坐标，即矩阵的范数
    w.append(model.w)
    lamda_norm = np.linalg.norm(model.w, 2)
    rate.append(lamda_norm / w_norm)
    alpha = alpha * 0.75
w = np.array(w)          # w为100*4的矩阵
rate = np.array(rate)    # rate为100*1的列向量
# print(rate)
```

【5】画图

实验最后得到的 Regularization Path 曲线如下所示：



根据曲线可以看出，初始情况下，即比值无穷下时，W 的每一个值的初始值也为 0，随后，随着w_λ的范数越来越接近w_∞，W 的每一项的值也越接近没有正则化时直接求得的系数值。

- **第二题：拟合图片中的点**

【1】 读取数据

本实验中采用 xlrd 模块读取 Excel 数据，读取第二个表格中共 116 行，3 列的数据，经过数据提取后读到 x 和 y 的位置点数据，数据读取代码如下：

```
xl = xlrd.open_workbook("E:\Study\研一\学习\机器学习\作业\附件：第三次作业_第二题.xlsx")
data = xl.sheets()[1]   # 选择第二个表格作为读入的data
rows = data.nrows   # data共116行,3列
x_train = []
y_train = []
# 把表格中的数据变为numpy数组
for i in range(1,rows):
    x = data.cell(i,2).value * -1
    y = data.cell(i,1).value * -1
    x_train.append(x)
    y_train.append(y)
x_train = np.array(x_train)
y_train = np.array(y_train)
x_test= x_train
```

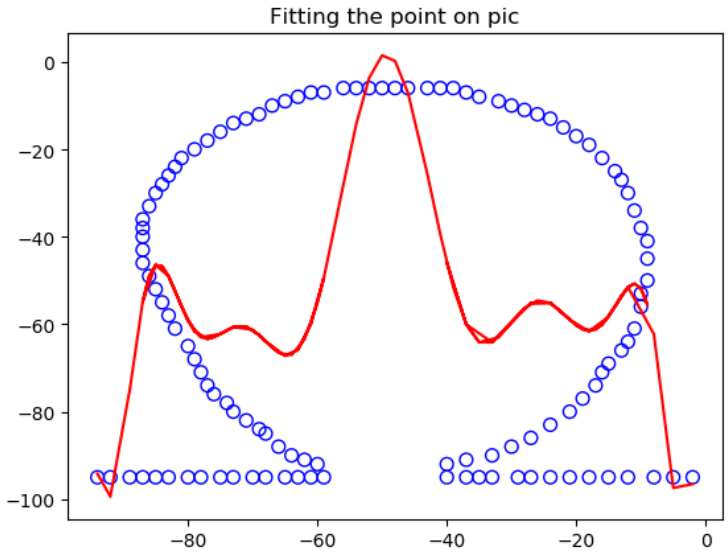
【2】 训练模型

在模型训练中，各参数取值如下，多项式系数 M=20，正则化系数λ=0.01，关键代码如下：

```
# 训练模型
alpha = 0.01
feature = PolynomialFeature(degree=20)
X_train = feature.transform(x_train)
X_test = feature.transform(x_test)
model = RidgeRegression(alpha=alpha)
model.fit(X_train, y_train)
y = model.predict(X_test)
```

【3】 画图

实验最后得到的拟合曲线如下所示：



四、实验总结

通过本次试验，我对于通过直接矩阵运算求解多项式系数以及正则化系数对多项式拟合程度的影响有了更加深入的认识，在λ的值从 1 开始逐渐变小的过程中，MSE 的值逐渐变小，模型的拟合程度也越来越好，同时多项式系数矩阵也更加接近通过直接法求得的系数矩阵的值；同时对于直接拟合图片中曲线的部分，在实验过程中主要问题就是在于调节多项式系数参数以及正则化参数两个值，并经过多次实验最终确定的这两个值。多项式系数并不是越高越好，λ也不是越小越好，所以模型是否 fit 很多情况下不仅仅取决于理论情况，实际过程中对参数的调节才是最主要的。