岭回归

1. 实验目的

了解岭回归算法的原理，并且可以简单应用

2. 算法原理

1)介绍

Ridge regression 通过对系数的大小施加惩罚来解决 普通最小二乘法 的一些问题。岭回归系数最小化的是带惩罚项的残差平方和，数学形式如下：

$$min \sum_{i=1}^{p} \|X\omega_i - y\|^2 + \alpha\|\omega\|^2$$

其中， α >= 0 是一个控制缩减量（amount of shrinkage）的复杂度参数： α 的值越大，缩减量就越大，故而线性模型的系数对共线性（collinearity）就越鲁棒。（L2 正则化）换句话说，让各个特征对结果的影响尽可能的小，但也能拟合出不错的模型。

与普通最小二乘法一样，Ridge 会调用 fit 方法来拟合数组 X， y，并且将线性模型的系数 ω 存储在其成员变量 coef_，截距存储在 intercept_:

2）岭回归曲线图

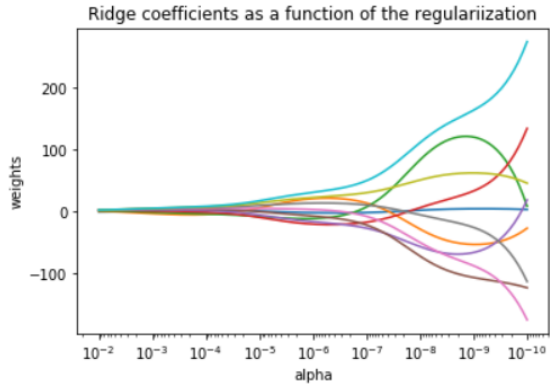
展示共线性（collinearity）对估计器系数的影响

这个例子中用到的模型是岭回归估计器（Ridge）。每种颜色表示系数向量的不同特征，并将其显示为正则化参数的函数。

此示例还显示了将岭回归应用于高度病态（ill-conditioned）矩阵的有效性。对于这样的矩阵，目标变量的微小变化会导致计算出的权重的巨大差异。在这种情况下，设置一定的正则化（alpha）来减少这种变化（噪声）是很有用的。

当 alpha 很大时，正则化效应将会主导（控制）平方损失函数，线性模型的系数也将趋于零。在路径的末尾，当 alpha 趋于零时，系数趋于没有设置正则化项的普通最小二乘法的系数，系数会出现很大的震荡（为高度病态矩阵）。

总共有 10 个系数，10 条曲线，一一对应。



3. 实验环境

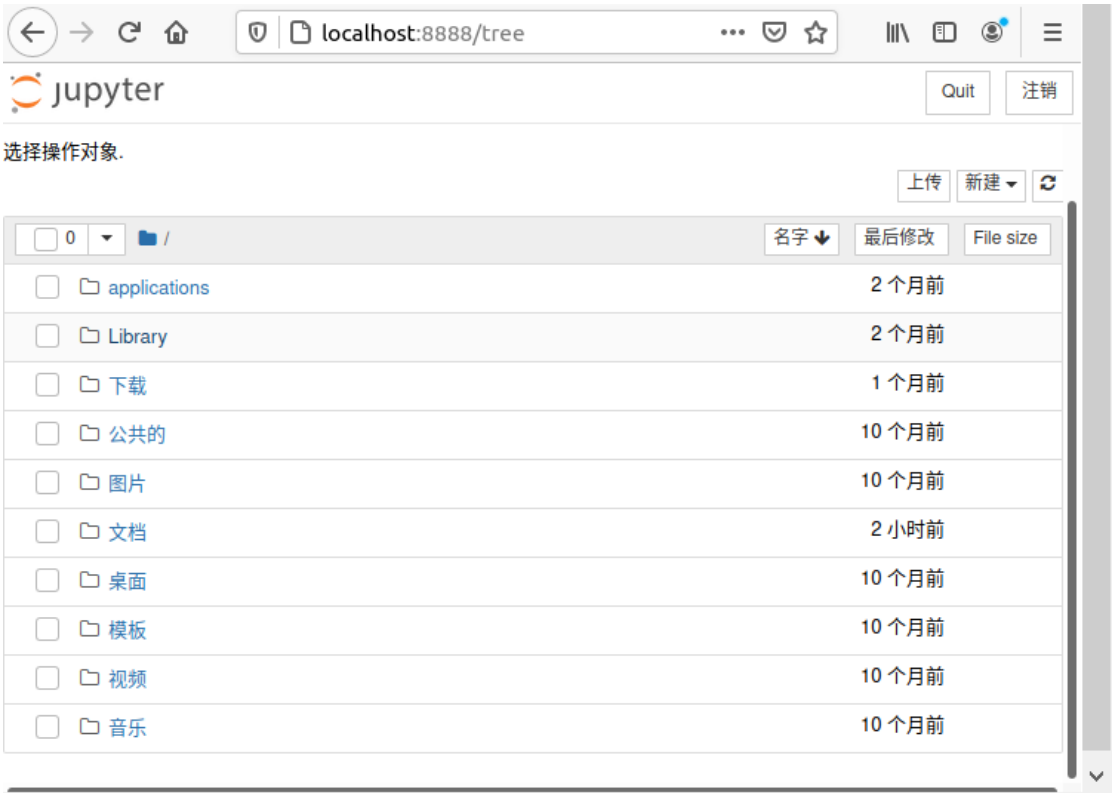
Ubuntu 20.04

Python 3.6

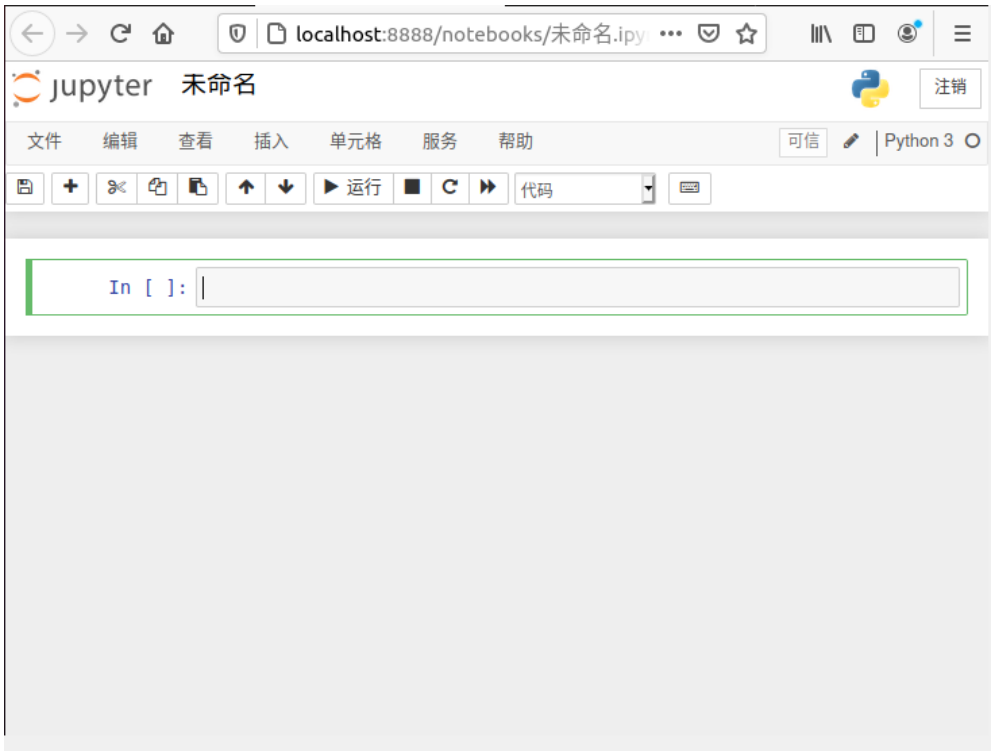
Jupyter notebook

4. 实验步骤

1)打开终端，然后输入 jupyter notebook，出现如下界面



2)选定特定文件夹，新建 ipynb 文件，在未命名出可重命名文件



5. 实操

Step 1:数据预处理

1. 导入库
2. 设置数据
3. 显示数据

```
#导入库
from sklearn.linear_model import Ridge

#设置数据
x = [[2,1,1],[1,2,3]]
y = [3,1]


# 显示数据
# alpha 就是 λ
ridge = Ridge(alpha=0.01)
ridge.fit(x,y)
```

Step 2:研究 λ 对岭回归系数的影响

```
# 研究 λ 对岭回归系数的影响
#
alphas = np.logspace(-10, -2, 200)
alphas
```

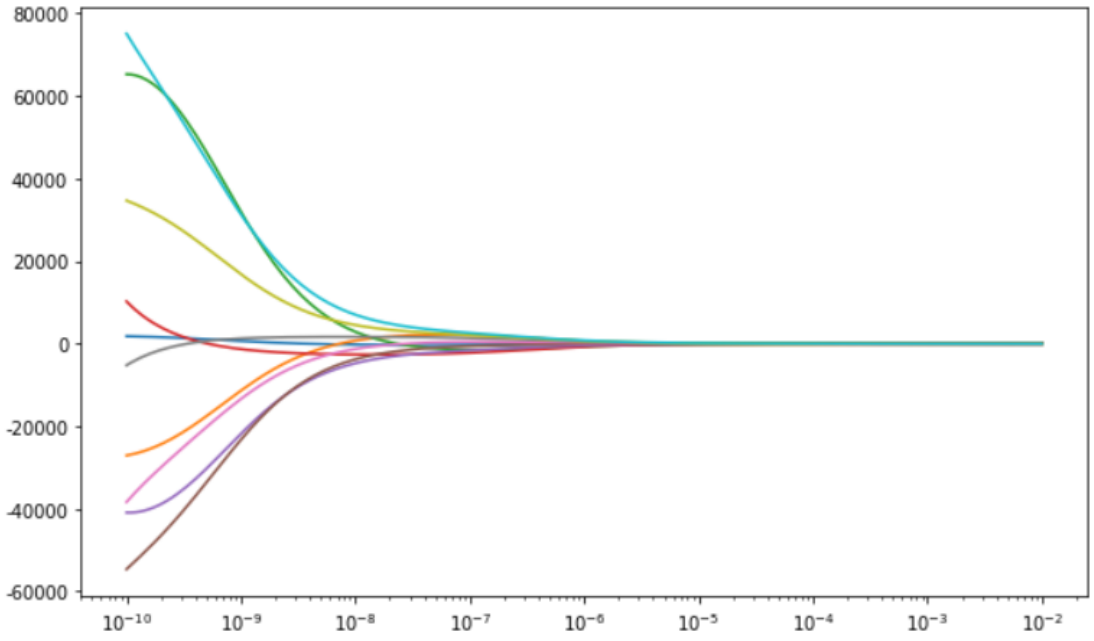
Step 3:训练模型

```
ridge = Ridge()

coefs = []
for alpha in alphas:
    ridge.set_params(alpha=alpha)
    # 使用不同的 λ 系数的岭回归模型，训练相同的一组数据集
    ridge.fit(X,y)
    # 每训练一次，都会得到一组系数
    coefs.append(ridge.coef_)
```

Step 4：绘图展示

```
# 绘图展示 λ 和 coef 之间的关系
plt.figure(figsize=(10,6))
data = plt.plot(alphas,coefs)
plt.xscale('log')
# 对测试集数据进行处理
X_test_pca = pca.transform(X_test_std)
```


80000
60000
40000
20000
0
-20000
-40000
-60000
10^{-10}
10^{-9}
10^{-8}
10^{-7}
10^{-6}
10^{-5}
10^{-4}
10^{-3}
10^{-2}