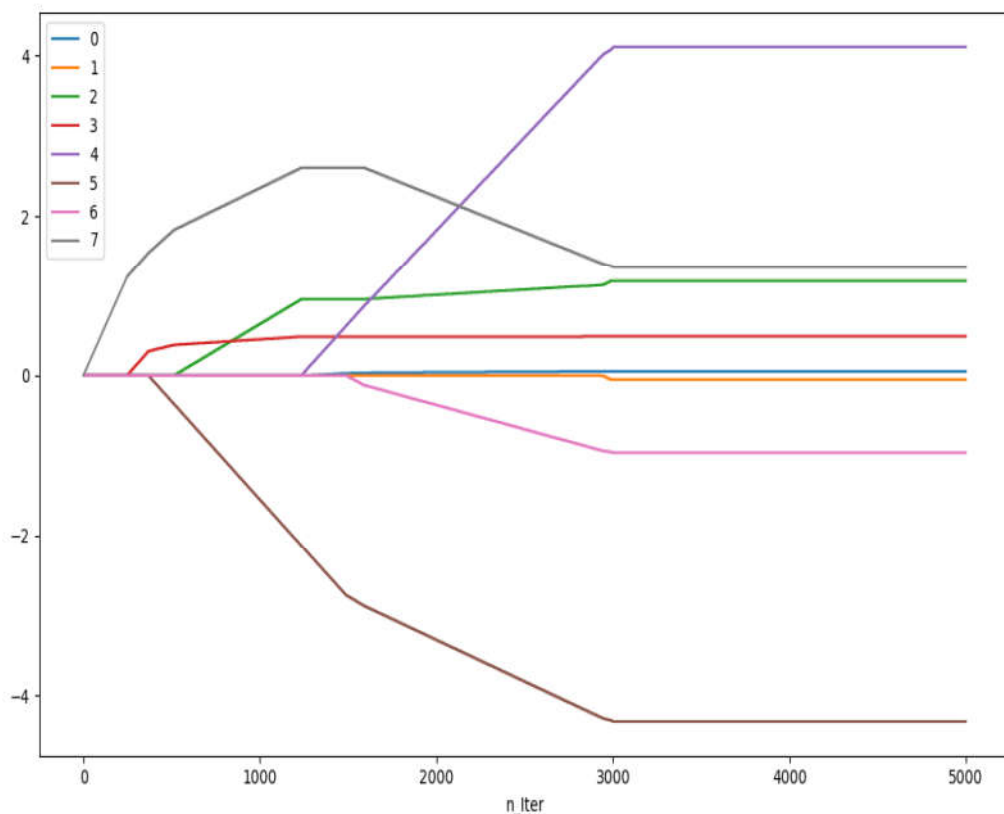


# Python编程与人工智能实践



## 算法篇：前向逐步回归 StageWise Regression

于泓

鲁东大学

信息与电气工程学院

2022.11.18

# 前向逐步回归

- 前向逐步回归属于一种贪心算法，通过对学习权重的逐步试探，得到最优的结果。
- 该方案不对需要对损失函数进行复杂的求导，可以应用于各种不同的损失函数上。

(1) 确定损失函数

$$L_{\text{linear}} = \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

$$L_{\text{ridge}} = \sum_{i=1}^N (y_i - \hat{y}_i)^2 + \lambda \|\mathbf{w}\|_2$$

$$L_{\text{lasso}} = \sum_{i=1}^N (y_i - \hat{y}_i)^2 + \lambda \|\mathbf{w}\|_1$$

(2) 求解 $\mathbf{w}$  令  $\frac{\partial L}{\partial \mathbf{w}} = 0$

有时很难得到  
闭式解。

# 算法流程

数据初始化

w初始化

在每轮的迭代中:

设置误差 lowest\_error 为无穷大

对w的每个特征维度:

增大或者缩小 eps

计算误差 error

若  $error < error\_lowest$

更新 w

```
def stageWise(X,Y,eps=0.01,n_iter=100):
    # 获取数据信息
    N,D= np.shape(X)
    # 记录每次迭代得到的w
    return_ws = np.zeros((n_iter,D))
    ws = np.zeros((D,1))
    ws_Test = ws.copy()
    ws_Max = ws.copy()
    # 最小误差
    lowest_Error = np.inf
    for i in range(n_iter):
        # print(ws.T)
        lowest_Error = np.inf
        # 每一个维度进行搜索
        for j in range(D):
            # 2个搜索方向
            for sign in [-1,1]:
                ws_Test = ws.copy()
                ws_Test[j] += eps*sign
                Y_hat = np.dot(X,ws_Test)
                # 计算误差
                error = lasso_error(Y_hat,Y,ws_Test,lam=0.1)
                # error = rss_error(Y_hat,Y)
                # 如果误差减小则进行w保存
                if error< lowest_Error:
                    lowest_Error = error
                    ws_Max = ws_Test
            ws = ws_Max.copy()
        print(ws.T,lowest_Error)
        return_ws[i] = ws.T
    return return_ws,ws
```

可以代入不同的  
损失函数

```
def rss_error(Y_hat,Y):  
    N,_ = Y.shape  
    error = np.sum((Y_hat-Y)*(Y_hat-Y))/N  
    return error
```

最小二乘回归

```
def lasso_error(Y_hat,Y,w,lam):  
    N,_ = Y.shape  
    error = np.sum((Y_hat-Y)*(Y_hat-Y))/N + lam*np.sum(np.abs(w))  
    return error
```

Lasso回归

```
def ridge_error(Y_hat,Y,w,lam):  
    N,_ = Y.shape  
    error = np.sum((Y_hat-Y)*(Y_hat-Y))/N + lam*np.sum(np.abs(w)**2)  
    return error
```

岭回归

绘图函数  
绘制权重的  
变化过程

```
def plot_ws(ws_all):  
    fig = plt.figure() # 创建绘图对象  
    ax = fig.add_subplot(1,1,1)  
    n_iter,D = np.shape(ws_all)  
  
    for i in range(D):  
        ax.plot([i for i in range(n_iter)],ws_all[:,i],label=str(i))  
    ax.set_xlabel('n_iter')  
    plt.legend(loc='best')  
    plt.show()
```

```
if __name__ == "__main__":  
    # 真实数据测试:  
    X,Y= load_DataSet('鲍鱼.txt',col_X=(0,1,2,3,4,5,6,7),col_Y=(8))  
  
    # 对数据进行正则化  
    mean_X= np.mean(X,axis=0,keepdims=True)  
    std_X = np.std(X,axis=0,keepdims=True)  
    X = (X-mean_X)/std_X  
  
    mean_Y = np.mean(Y,axis=0,keepdims=True)  
    Y= Y-mean_Y  
  
    # 进行 stageWise 回归  
    ws_all,ws_best = stageWise(X,Y,eps=0.001,n_Iter=8000)  
  
    plot_ws(ws_all)
```

线性回归

LASSO回归

岭回归