
华中科技大学

《计算机视觉理论与实践》课程项目报告

电子信息与通信学院

2021 年秋季

姓名：王锴、逯凌云、李雯钰、周雄涛

学号：U201713379、U201713358、U201712427、U201715418

专业：电信 1705、电信中英班

时间：9.14

一、 方法原理

该问题求解的原理即线性回归模型，根据梯度下降法是模型逐渐收敛至符合错误率为止。

线性模型试图学得一个通过属性（特征）的线性组合来进行预测的函数，预测函数的形式如下：

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$

利用梯度下降法求解最优参数，该模型的代价函数为：

$$J(w, b) = \frac{1}{2} (h(x) - y)^2 = \frac{1}{2} (wx - y)^2$$

在第 $t+1$ 次迭代中， w 和 b 两参数的更新公式为：

$$b^{t+1} = b^t - \alpha \frac{\partial J(w, b)}{\partial b} (w^t, b^t), \quad w^{t+1} = w^t - \alpha \frac{\partial J(w, b)}{\partial w} (w^t, b^t)$$

最终求出符合最低要求的模型，可能是一个局部最优解，也可以采取一些手段减少求解过程陷入局部最优解的可能性。

二、 实验步骤

1.1 初始化权值和阈值

将权值 `weight` 和阈值 `b` 都初始化为 0。

1.2 模型训练、更新权值

细分为两个步骤，首先根据线性回归模型计算预测值，其次如果预测结果错误则需根据更新公式修改权值和阈值。

```
for i in range(num):
    x = train_data[i]
    y = train_label[i]
    if y * (np.dot(w, x) + bias) <= 0: # 当预测值错误时，需要更新权重
        w = w + rate * y * x.T # update weight: w = w + r * x_i * y_i
        bias = bias + rate * y # update b: b = b + r * y_i
```

1.3 计算损失函数、判断是否结束训练

根据上一轮对整个样本集的迭代求解出的模型，依次对每个样本做预测，当错误率小于给定值时停止训练。

```
# 计算当前模型的错误率, 如果满足要求则停止训练
loss = 0
for i in range(num):
    y = train_label[i]
    output = np.dot(w, train_data[i]) + bias
    if output * y <= 0:
        loss += 1
if (loss / num) <= error:
    return w, bias
```

1.4 绘图函数

```
for d, sample in enumerate(X):
    # Plot the negative samples
    if d < size:
        plt.scatter(sample[0], sample[1], s=120, marker='_')
    # Plot the positive samples
    else:
        plt.scatter(sample[0], sample[1], s=120, marker='+')
plt.plot([-2, 5], [w[0]/w[1] * 2 - bias/w[1], w[0]/w[1] * -5 - bias/w[1]])
plt.show()
pass
```

三、 方法参数讨论

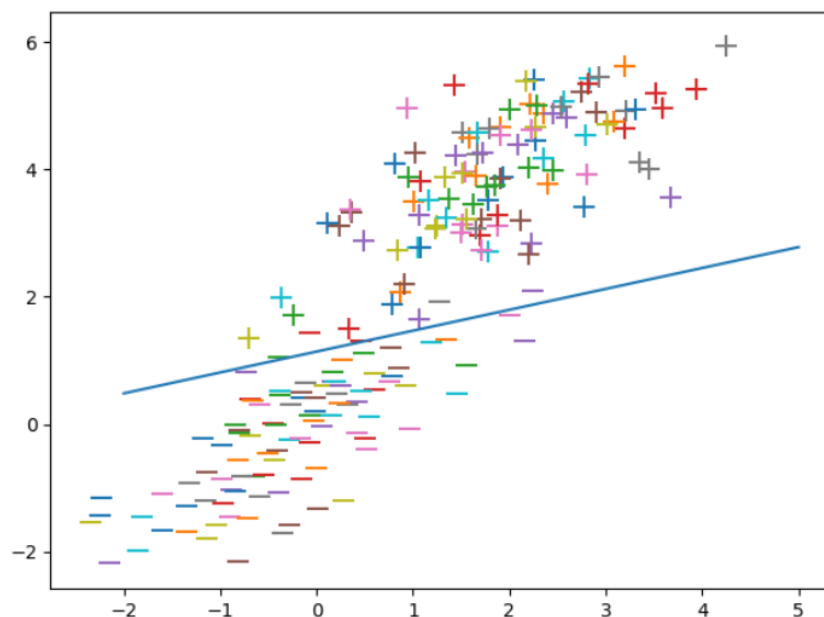
我们的模型训练过程主要有两个参数, 一是学习率, 即更新参数的步长; 二是停止训练的错误率。

学习率, 即参数到达最优值过程的速度快慢, 如何调整学习率是训练出好模型的关键要素之一, 太大容易出现超调现象, 即在极值点两端不断发散, 或是剧烈震荡, 总之随着迭代次数增大 loss 没有减小的趋势; 太小会导致无法快速地找到好的下降的方向, 随着迭代次数增大 loss 基本不变。

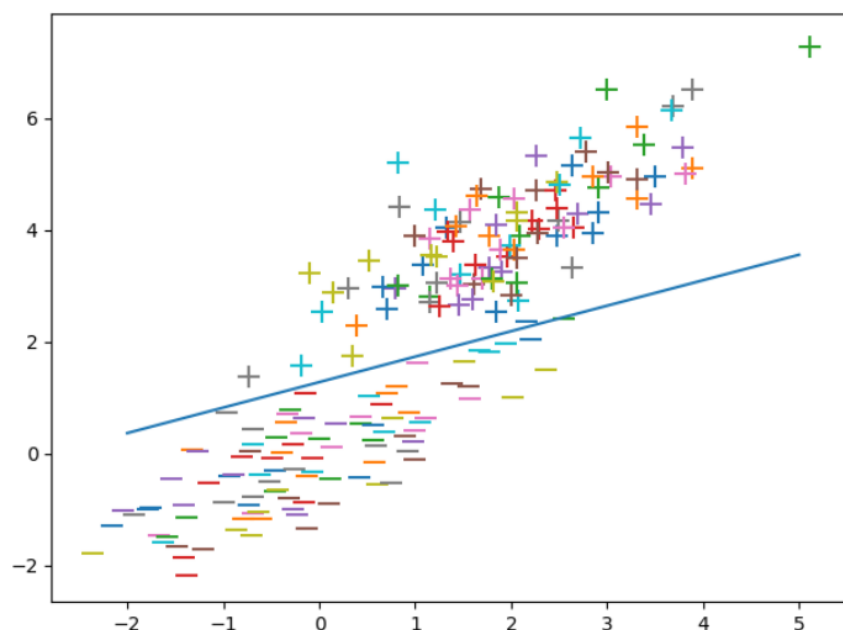
停止训练的错误率过大易导致欠拟合, 模型性能不好, 没能学习到样本集更多的特征; 设置过小则容易导致过拟合, 导致模型的泛化性能不佳。

四、 实验结果

第一次将学习率设置为 0.01, 错误率设为 0.05, 结果如下:



尝试继续降低错误率，会增加一些时间开销，同时当随机数据分布线性不可分时，无法训练出想要的模型，下图为错误率降为 0.05 时的结果：



接下来尝试修改学习率超参数，当我将学习率调为 0.1 时，学习率过大，梯度下降陷入反复震荡，最终无法收敛，无法训练出模型，当我将学习率调为 0.001 时，会增加很多不必要的时间开销，同时还是无法解决线性不可分问题。

五、 小组分工

我们四人先理清了代码的算法思路，随后各自开始编程，最终再整合成一个

完善的版本，然后一人负责撰写报告，一人负责做汇报 PPT，一人最后作报告。

代码地址如下：

<https://github.com/ChengHe9746/CVTasks>