# Logistic 回归实验报告

王昊然 PB20010382

2023 年 10 月 29 日

## 1 简介

Logistic 回归是一种广泛应用的统计方法，用于解决二元分类问题。它通过估计一个事件发生的概率来预测类别。Logistic 回归的核心是 sigmoid 函数，该函数能够将任意的线性组合的输入映射到 0 和 1 之间的概率输出。

本实验旨在通过构建和评估 Logistic 回归模型，来理解其在解决实际分类问题中的应用。我们将使用在 github 课程主页上的一个公开的数据集来训练我们的模型，并评估其在 TEST dataset 上的性能。数据的链接为DATA

## 2 理论背景

Logistic 回归是一种用于二分类问题的统计方法。与线性回归不同，Logistic 回归旨在预测一个二元输出变量的概率。它的核心是 Logistic 函数（或称为 Sigmoid 函数），该函数能够将线性组合的输入映射到 $(0, 1)$ 区间的概率值。

### 2.1 数学模型

Logistic 回归模型的基本形式可以表示为:

$$P(Y = 1|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1^T x)}} \tag{1}$$

这里，$x$ 是输入向量，$\beta_0$ 和 $\beta_1$ 是模型的参数。我们也可以将模型写成一个稍微不同的形式，使得参数的表示更紧凑:

$$P(Y=1|x) = (1+\exp(-\beta^T \tilde{x}))^{-1} \tag{2}$$

其中，$\tilde{x} = [1, x]$ 是一个扩展的输入向量，$\beta$ 是合并了 $\beta_0$ 和 $\beta_1$ 的参数向量。

## 2.2 损失函数

为了训练 Logistic 回归模型，我们需要定义一个损失函数来度量模型的性能。通常，我们使用对数似然损失函数:

$$L(\beta) = -\sum_{i=1}^{n}[y_i \log(P(Y=1|x_i)) + (1-y_i)\log(1-P(Y=1|x_i))] \tag{3}$$

## 2.3 优化方法

我们的目标是找到一组参数 $\beta$，以最小化损失函数 $L(\beta)$。通常，我们使用梯度下降法来实现这个目标。梯度下降法的基本思想是沿着损失函数的负梯度方向更新参数，以逐步降低损失函数的值。

梯度下降法的更新规则是:

$$\beta := \beta - \alpha \nabla L(\beta) \tag{4}$$

其中，$\alpha$ 是学习率，$\nabla L(\beta)$ 是损失函数 $L(\beta)$ 关于参数 $\beta$ 的梯度。梯度可以通过以下公式计算:

$$\nabla L(\beta) = -\sum_{i=1}^{n} x_i (y_i - P(Y=1|x_i)) \tag{5}$$

### 2.3.1 步长调整

如果 loss 函数有所增加，lr 进行缩小；如果 lr<1e-7 则停止迭代，或者如果 loss 连续增加则停止迭代

### 2.3.2 使用 L1 或 L2 正则化

**数学原理** 在 Logistic 回归的训练过程中，我们通常最小化对数似然损失函数。为了控制模型的复杂度，我们可以在损失函数中添加一个正则项。

L1 正则化的正则项为：

$$R_{L1}(\beta) = \lambda \sum_{i=1}^{p} |\beta_i| \tag{6}$$

L2 正则化的正则项为：

$$R_{L2}(\beta) = \lambda \sum_{i=1}^{p} \beta_i^2 \tag{7}$$

其中，$\beta$ 是模型的参数，$p$ 是参数的数量，$\lambda$ 是正则化强度的超参数。

**L1 正则化和参数稀疏** L1 正则化有一个重要的属性，就是它倾向于产生稀疏解，即许多参数的值会被压缩到零。这是因为 L1 范数在零点处不可微，且正则化路径是非线性的，导致参数在零点附近被截断。

**正则化的作用** 正则化的主要目的是防止模型过拟合，提高模型的泛化性能。

# 3 实验方法

## 3.1 数据处理

### 3.1.1 处理 NaN 值

处理数据中的 NaN 值是数据预处理的重要步骤。我们采用以下策略来处理不同类型的数据中的缺失值：

- 对于 `float64` 类型的数据：我们使用该列的均值来填充缺失值。
- 对于 `object` 类型的数据：我们首先对数据进行 one-hot 编码，然后使用多项式分布处理缺失值。具体来说，可以采用 Multinomial 分布来估算缺失值。

- 另外，对于经过 one-hot 编码的数据，也可以直接使用列的均值来替代 NaN 值。

### 3.1.2 One-hot 编码

对于分类数据，我们使用 one-hot 编码将 `object` 类型的数据向量化，以便于后续的计算和模型训练。One-hot 编码可以将分类变量转换为二进制向量，从而使得模型能够更好地理解和处理数据。

### 3.1.3 标准化或归一化

在处理完缺失值和编码分类变量后，我们对数据进行标准化或归一化，以确保不同尺度的特征不会影响模型的性能。

**标准化** ：标准化是指将数据按照标准正态分布（均值为 0，标准差为 1）进行缩放。标准化的公式为：

$$z = \frac{x - \mu}{\sigma} \tag{8}$$

其中，$x$ 是原始数据，$\mu$ 是数据的均值，$\sigma$ 是数据的标准差。

**归一化** ：

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{9}$$

其中，$x$ 是原始数据，$\min(x)$ 和 $\max(x)$ 分别是数据的最小值和最大值。

## 3.2 生成训练集和测试集

为了评估模型的性能，我们需要将数据集分为训练集和测试集。在这个步骤中，我们首先将数据进行随机洗牌，然后按照一定的比例（例如，80% 训练，20% 测试）将数据分为两部分。同时，我们确保训练集和测试集中正负样本的比例与原始数据集中的比例相同。

# 4 实验结果

## 4.1 The Loss curve of one training process

下面展示两张不同参数下的 loss curve。可以发现，loss curve 在迭代次数较小时的下降速度很快，到后来趋于平稳并且有小幅度的波动

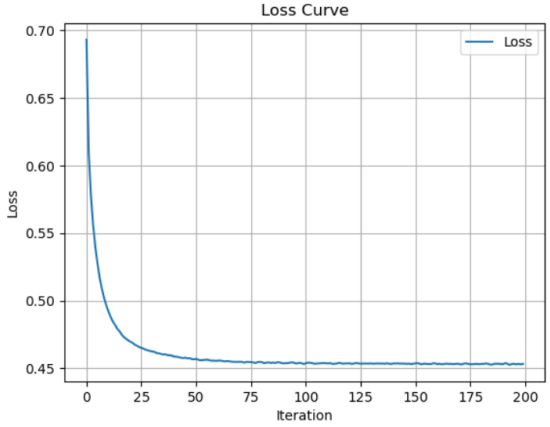



图 1: penalty=l1,$\gamma$=1

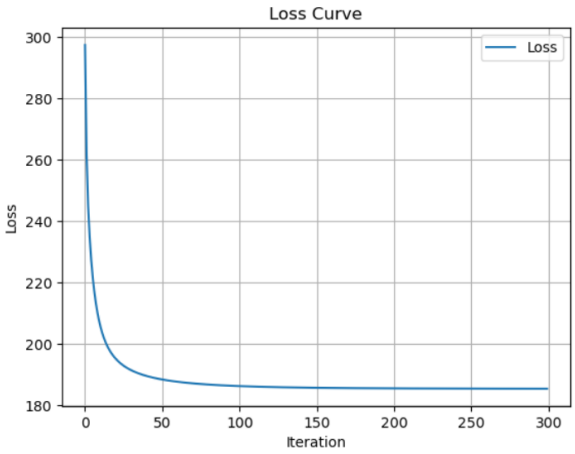



图 2: penalty=l2,gamma=1

## 4.2 The comparation table of different parameters

可以发现，对于此模型和此数据集而言，当步长一定时，适当的正则项对于测试集上的正确率的提升很有帮助。同时，l1,l2 的正则项各有优势，他们对于不同的正则项的超参数有着不同的表现。要注意到，太大的步长 (learning rate) 可能还会带来不好的效果。

表 1: Logistic 回归参数和准确率

| Index | Penalty | Gamma | Learning Rate (lr) | Max Iterations (max_iter) | Accuracy |
|---|---|---|---|---|---|
| 1 | l1 | 1 | 0.01 | 300 | 80.60% |
| 2 | l1 | 1 | 0.001 | 300 | 81.40% |
| 3 | l1 | 10 | 0.001 | 300 | 81.50% |
| 4 | l2 | 1 | 0.01 | 300 | 79.80% |
| 5 | l2 | 1 | 0.001 | 300 | 80.50% |
| 6 | l2 | 10 | 0.001 | 300 | 82.30% |
| 7 | None | None | 0.001 | 300 | 79.80% |

## 4.3 The best accuracy of test data

下面给出较为优秀的超参数设置:

$$penalty = l2$$

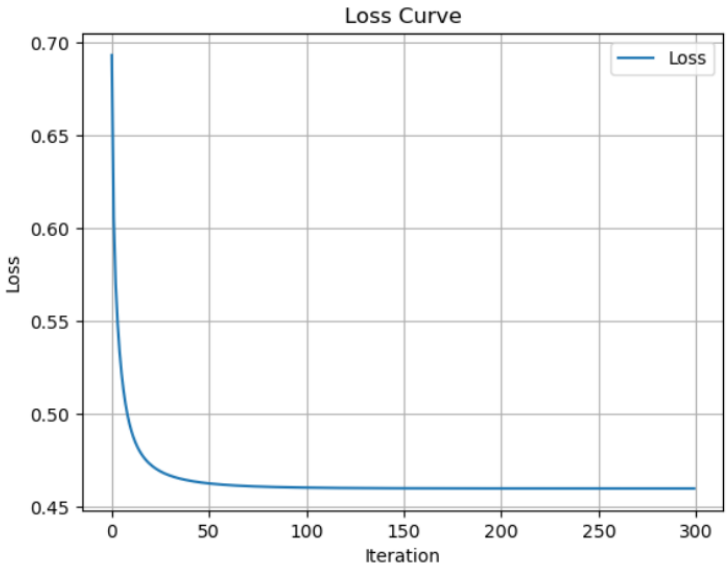
$$\gamma = 10$$

$$learningrate = 0.001$$

$$maxiter = 300$$

下面给出相应的正确率的截图和 loss 曲线值得注意的是，这组超参数并不一定是真正意义上最优的超参数，或许还有更好的超参数值的被发现，但那或许需要验证集来选择有意义的超参数，否则，只通过测试集选择超参数感觉不是很稳妥

```
iteration:292,    loss:2.25e+02
iteration:293,    loss:2.25e+02
iteration:294,    loss:2.25e+02
iteration:295,    loss:2.25e+02
iteration:296,    loss:2.25e+02
iteration:297,    loss:2.25e+02
iteration:298,    loss:2.25e+02
iteration:299,    loss:2.25e+02
test_acc:accuracy:82.2581%
```



# 5 结论

本实验通过实现和评估 Logistic 回归模型，深入探讨了其在二分类问题中的应用和性能。

- **模型性能**：Logistic 回归模型在处理二分类问题上表现出还行。通过适当的参数选择和正则化，我们能够得到满意的分类准确率。尤其是在应用 L1 和 L2 正则化时，模型的泛化性能得到了一定的提升，显示了正则化在防止过拟合和提升模型泛化能力方面的重要性。
- **数据预处理**：通过对数据的预处理，包括处理缺失值、one-hot 编码和数据标准化/归一化，我们确保了模型能够在清洁和规范的数据上进行有效的训练，突显了数据预处理在机器学习项目中的基础和重要作用。
- **损失曲线分析**：损失曲线的分析帮助我们理解了模型在训练过程中的表现和收敛情况。

综上所述，Logistic 回归作为一种简单而有效的分类方法，在适当的数据预处理和参数选择下，能够为解决实际的分类问题提供有力的支持。