机器学习课程实验报告

逻辑回归

学　　号	1180301007
姓　　名	赵锦涛
实验时间	2020 年 10 月

一、 实验目的:

理解逻辑回归模型，掌握逻辑回归模型的参数估计算法。

二、 实验要求:

实现两种损失函数的参数估计（1，无惩罚项；2. 加入对参数的惩罚），可以采用梯度下降、共轭梯度或者牛顿法等。

三、 实验环境:

Python 3.8, Windows 10

四、 实验原理:

逻辑回归是一种分类方法。对于二分类问题，给定输入 $X =< X_1, ..., X_n >$，输出 $Y \in \{0, 1\}$，有

$$P(Y = 1|X) = \frac{exp(w \cdot X + b)}{1 + exp(w \cdot X + b)}$$

$$P(Y = 0|X) = \frac{1}{1 + exp(w \cdot X + b)}$$

扩充 X 向量为 $X = \{1, X_1, ..., X_n\}$，w 向量为 $w = \{b, w_1, ..., w_n\}$，则后验类概率可表示为

$$P(Y = 1|X) = \frac{exp(w \cdot X)}{1 + exp(w \cdot X)}$$

$$P(Y = 0|X) = \frac{1}{1 + exp(w \cdot X)}$$

当训练得到参数 w 后，便可将样本代入求得相应的概率，样本分类类别取使概率较大的那类。

逻辑回归模型参数估计时可使用最大似然法估计方法参数，设

$$P(Y = 1|x) = p(x), P(Y = 0|x) = 1 - p(x)$$

似然函数为

$$\prod[p(x_i)]^{y_i}[1 - p(x_1)]^{1-y_i}$$

取对数有

$$
\begin{aligned}
L(w) &= \sum[y_i lnp(x_i) + (1-y_i) ln(1-p(x_i))] \\
&= \sum[y_i(w \cdot x_i) - ln(1+exp(w \cdot x_i))]
\end{aligned}
$$

相应的平均对数似然损失函数为（其中 N 为数据集大小）

$$
J(w) = -\frac{1}{N} L(w)
$$

该函数为一个下凸函数，最小化损失函数可以通过梯度下降法来求解。计算梯度有：

$$
\frac{\partial J(w)}{\partial w_j} = \frac{1}{N} \sum_{i=1}^{N} (p(x_i) - y_i) x_i^j
$$

所以参数的迭代过程为

$$
w_j := w_j - \alpha \frac{1}{N} \sum_{i=1}^{N} (p(x_i) - y_i) x_i^j
$$

设 sigmoid 函数

$$
sigmoid(x) = \frac{1}{1+e^{-x}}
$$

输入矩阵为

$$
X = \begin{bmatrix}
1 & x_1^1 & \cdots & x_1^m \\
1 & x_2^1 & \cdots & x_2^m \\
\vdots & \vdots & \ddots & \vdots \\
1 & x_n^1 & \cdots & x_n^n
\end{bmatrix}
$$

其中每行代表一个样本，这些样本所对应的类别向量 $Y = [y_1, y_2, ..., y_n]^T$，参数向量 $\boldsymbol{w} = [w_0, w_1, ..., w_m]^T$ 所以参数迭代过程的矩阵表示为

$$
\boldsymbol{w}_{i+1} = \boldsymbol{w}_i - \alpha \frac{1}{N} X^T (sigmoid(X\boldsymbol{w}_i) - Y)
$$

根据贝叶斯定理

$$
p(\boldsymbol{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{w}) p(\boldsymbol{w})}{p(\mathcal{D})}
$$

有

$$
posterior \propto likelihood \times prior
$$

根据最大后验准则，给模型加入正则项，此时损失函数变为

$$
J(w) = -\frac{1}{N}(\sum[y_i lnp(x_i) + (1-y_i) ln(1-p(x_i))] + \frac{\lambda}{2} w^T w)
$$

参数的迭代过程变为

$$w_j := (1-\alpha\frac{\lambda}{N})w_j - \alpha\frac{1}{N}\sum_{i=1}^{N}(p(x_i)-y_i)x_i^j$$

矩阵迭代形式为

$$\boldsymbol{w}_{i+1} = (1-\alpha\frac{\lambda}{N})\boldsymbol{w}_i - \alpha\frac{1}{N}X^T(sigmoid(X\boldsymbol{w}_i)-Y)$$

五、 代码实现

本次实验共有 7 个文件，其名称和作用分别为：

- *datagen.py* 生成训练数据
- *calcacc.py* 计算精度
- *loss.py* 计算损失函数
- *plot.py* 绘制图表
- *gradient.py* 梯度下降法
- *uci.py* 对 UCI 数据集进行预处理
- *uci_gradient.py* 对预处理后的 UCI 数据使用梯度下降法进行训练

在本次实验中，自主生成的数据样本共有四个维度，满足贝叶斯假设条件下，各维度符合高斯分布，其均值分别为 $1, 2, 3, 4$，方差均为 0.2。不满足贝叶斯假设条件下，各维度的协方差矩阵为

$$Cov = \begin{bmatrix} 0.2 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.2 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.2 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.2 \end{bmatrix}$$

分类规则为当样本各维度的和大于 10 时分为正类，否则为负类。UCI 数据集使用的是 haberman 数据集，具体信息请见Haberman's Survival Data Set

六、 实验结果与分析

（一）数据集大小

实验过程中，使用了不同大小的数据集进行测试。训练集的样本大小分别为 20 和 100，测试集的样本大小为 100，学习率为 0.1，各运行 20 次，计算其精度，

结果如图一所示。可以看出，训练集大小为 20 的时候，测试结果较差，出现了欠

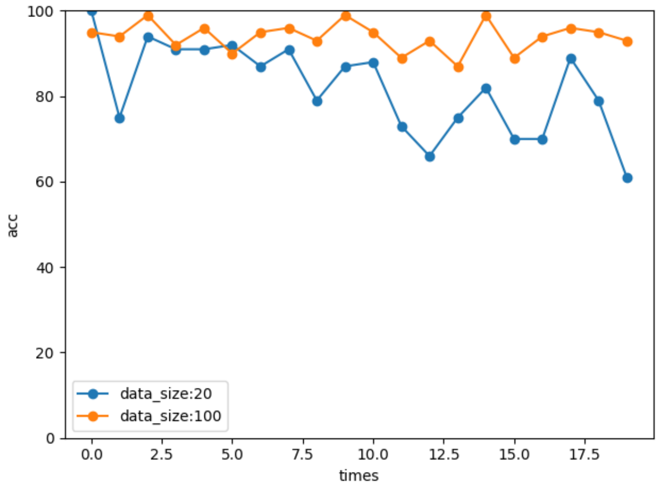

图 1: data_size = 20, data_size = 100

拟合现象。提高样本数量至 100 后，能够较好地提高模型预测的精度，解决了欠拟合的问题。

（二）正则项

图二为加入正则项和不加入正则项的实验结果对比。其中，训练样本个数均为 30，正则化项系数为 0.0008。可以发现，加入正则项后，整体而言精度有所提高，但其效果不如增加训练样本，这可能和正则化项系数的选择有关。

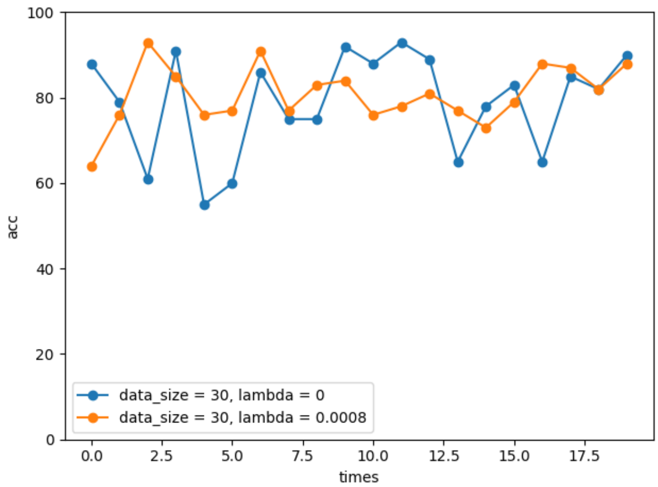

图 2: 加入正则化项

（三）贝叶斯假设

实验中使用了满足贝叶斯假设的数据和不满足贝叶斯假设的数据进行测试，其训练样本个数均为 50，可以发现两者效果相近，都比较好。逻辑回归方法也能对不满足贝叶斯假设的数据进行良好分类。

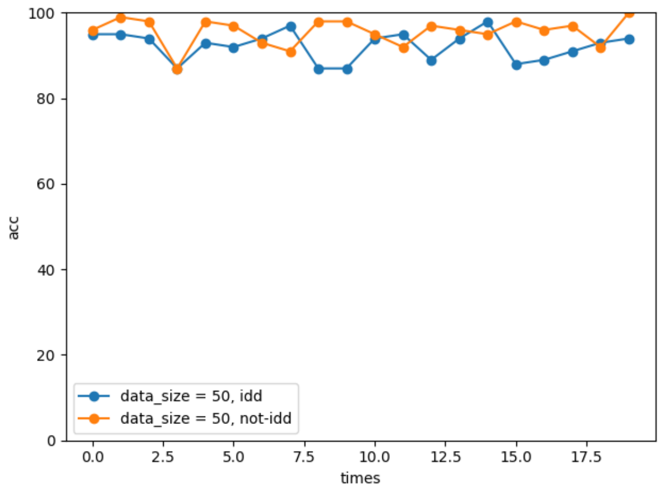

图 3: 贝叶斯假设

（四）UCI 数据集测试

使用 UCI 数据集测试结果如下。将 UCI 数据进行可视化发现，该数据在三维

```
[[-0.42154324]
 [ 0.01698082]
 [-0.02977347]
 [ 0.08835751]]
Accuracy:76.08695652173914%
```

图 4: UCI Haberman

空间并不线性可分，所以模型效果较差。

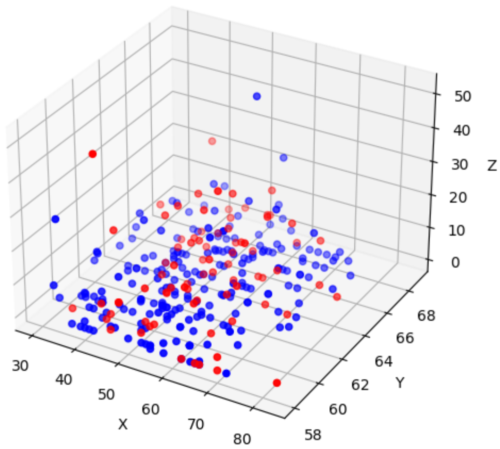

图 5: UCI Data

参考文献

[1] 空 中 的 鱼 1987. 逻 辑 回 归 梯 度 下 降 法 详 解 [EB/OL].https://blog.csdn.net/lookqlp/article/details/51161640,2016-04-19.

[2] 阿泽.【机器学习】逻辑回归（非常详细 [EB/OL].https://zhuanlan.zhihu.com/p/74874291,2019-08-01.