哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

2020 年春季学期

计算学部《机器学习》课程

Lab 2 实验报告

姓名	王科龙
学号	1180801203
班号	1803104
电子邮件	1264405807@qq.com
手机号码	19917620613

目录

1 实验目的

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。

2 实验环境

Pcharm2020.2.2x64

Python3.8

Numpy 1.19.2

3 数学原理

在逻辑回归模型中：
逻辑回归模型

$$P(Y=1|X)=\frac{e^{\omega^T X}}{1+e^{\omega^T X}}$$

$$P(Y=0|X)=\frac{1}{1+e^{\omega^T X}}$$

其中$X=\{1,x_1,x_2\dots,x^n\}.T\ \ \omega=\{\omega_0,\omega_1,\dots,\omega_n\}$

只要计算得出ω，通过$P(Y=1|X)$= $P(Y=0|X)$就可以得到一条正例概率等于反例概率的直线，当给出一个新的特征X时，将其带入特征中计算，就可以对其所属的分类做出预判。
定义$sigmod$函数为

$$sigmod(z)=\frac{1}{1+e^{-z}}$$

其中$z=e^{\omega^T X}$

估计ω,可以使用 MLE 和 MAP 两种方法，对应的损失函数一个不带正则项，一个带正则项
最大似然估计：
似然函数，由于对于各个样本的数据做了条件独立假设。

$$l(\omega)=ln\prod_l P(Y^l|X^l,\omega)$$

$$=\sum_l Y^l\omega^T X^l-\ln\left(1+e^{\omega^T X^l}\right)$$

似然函数的梯度，对于ω的每一维来说,将每一维计算出来就可以得到梯度。

$$\frac{\partial l(\omega)}{\partial\omega_i}=\sum_l x_i^l(Y^l-sigmod(e^{\omega^T X^l}))$$

MLE 要求使得似然函数最大，因为似然函数是负定的，存在最大值。就可以使用梯度下降来更新ω^*，使其逼近最大值点。
对ω的每一维的更新公式为

$$\omega_i=\omega_i+\alpha\frac{\partial l(\omega)}{\partial\omega_i}$$

其中α为学习率。

最大后验估计:增加惩罚项

这种方法求解的是

$$\omega \leftarrow \underset{\omega}{argmax}\ ln\left(P(\omega)l(\omega)\right)$$

假设ω服从高斯分布$N(0,\sigma)$
对应的似然函数为

$$l(\omega) = \sum_{l} Y^{l}\omega^{T}X^{l} - \ln\left(1 + e^{\omega^{T}X^{l}}\right) + \frac{\lambda}{2}\omega^{T}\omega$$

梯度：对于ω的每一维来说

$$\frac{\partial l(\omega)}{\partial \omega_i} = \sum_{l} x_i^{l}(Y^{l} - sigmod(e^{\omega^{T}X^{l}})) + \lambda\omega$$

对ω的每一维的更新公式为

$$\omega_i = \omega_i + \alpha\frac{\partial l(\omega)}{\partial \omega_i} + \lambda\omega$$

4 实验做法

4.1 生成数据

由于多维数据不好在平面图中显示，因此本次手工生成数据的X为二维数据
通过分别指定高斯分布均值和方差，可以生成不同类别的数据。
生成不满足朴素贝叶斯假设的数据

朴素贝叶斯假设要求特征的各位数据之间相互条件独立

若协方差矩阵不为 0，则各个维度之间不独立，也不会条件独立。

4.2 计算似然函数

由于原理中的似然函数中有求和号，通过一个循环完成求和

4.3 计算梯度

先计算每一维的导数，然后在将各个维度的导数集合起来形成梯度向量
计算一维的梯度需要一次循环，将每一维都计算需要一次循环。

4.4 梯度下降

先为每一个特征数据增加前置的 1，用来完成和ω^T的计算。

通过计算前一次迭代和这次迭代的似然函数差值来确定是否越过了最大值点，需要降低学习率

```
value0 = value1
value1 = likelihood_function(xx_list, y_list, w, select_size)
if (value1 - value0) < 0:
    learning_rate = learning_rate * 0.5  # 走过了，降低学习率
```

迭代公式

```
g = grad(xx_list, y_list, w, select_size)
w = w + learning_rate * g
```

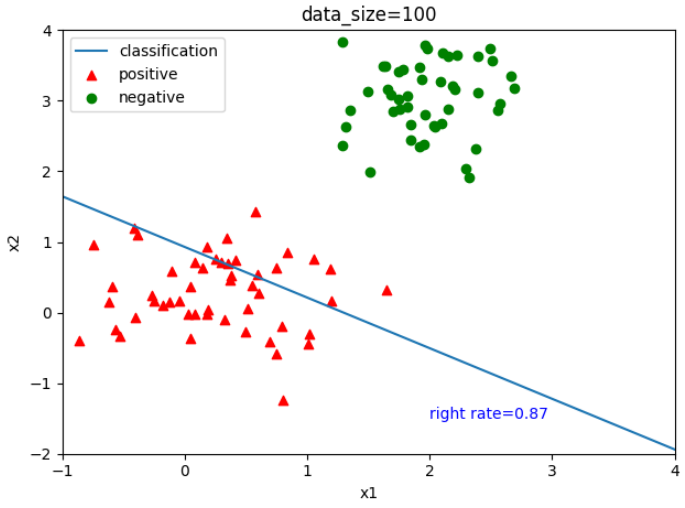
4.5 读取 UCI 的数据

UCI 的数格式为 csv 格式，一行的数据之间通过‘;’隔开，读取到一行数据之后在转换成手工生成的数格式即可。

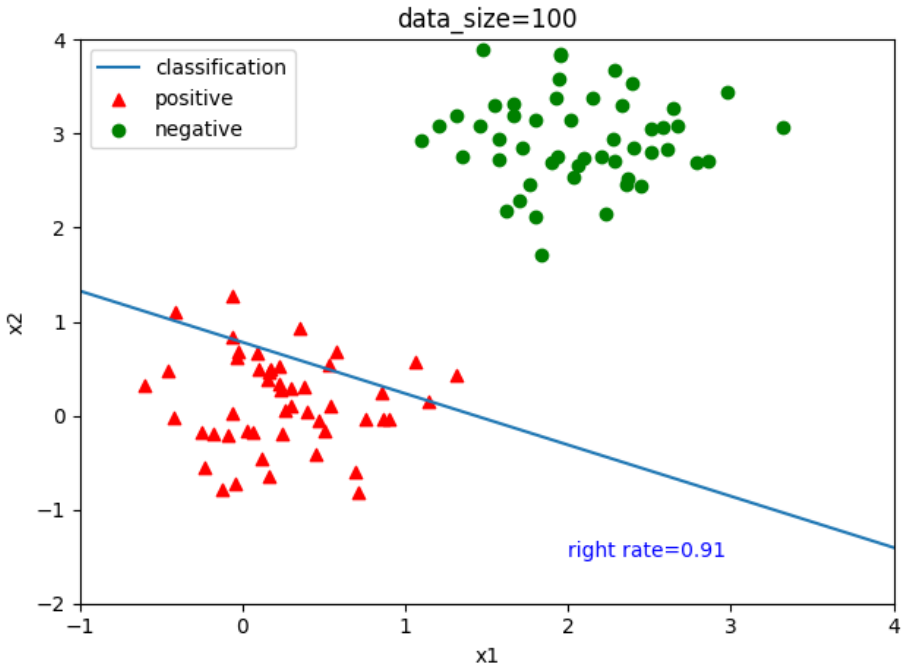
5 实验结果分析

5.1 不携带正则项

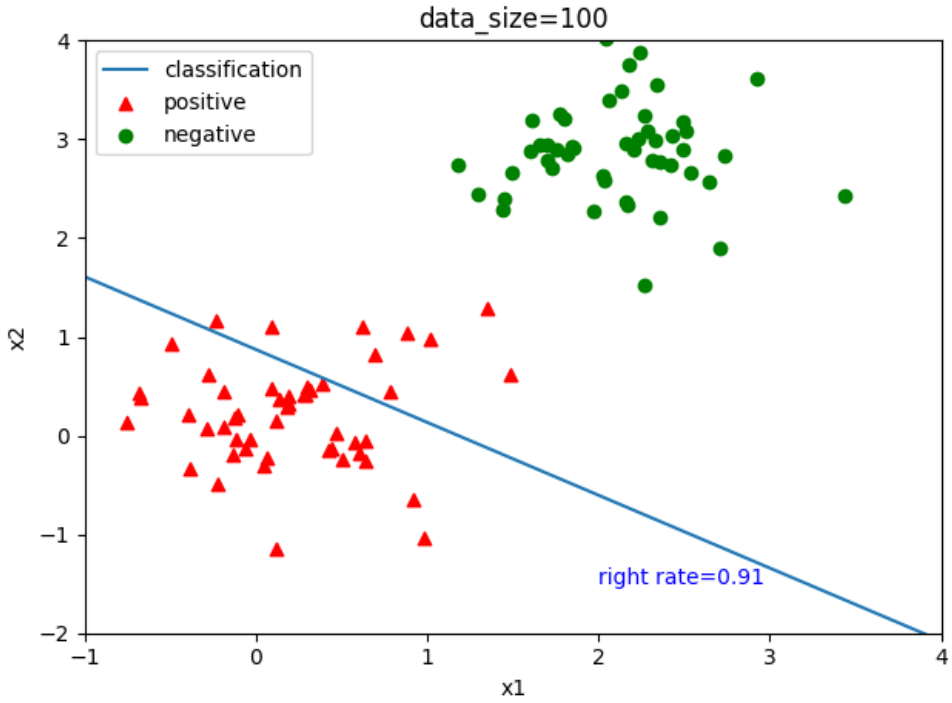
不同的迭代次数，实验的结果不同
迭代 150 次：



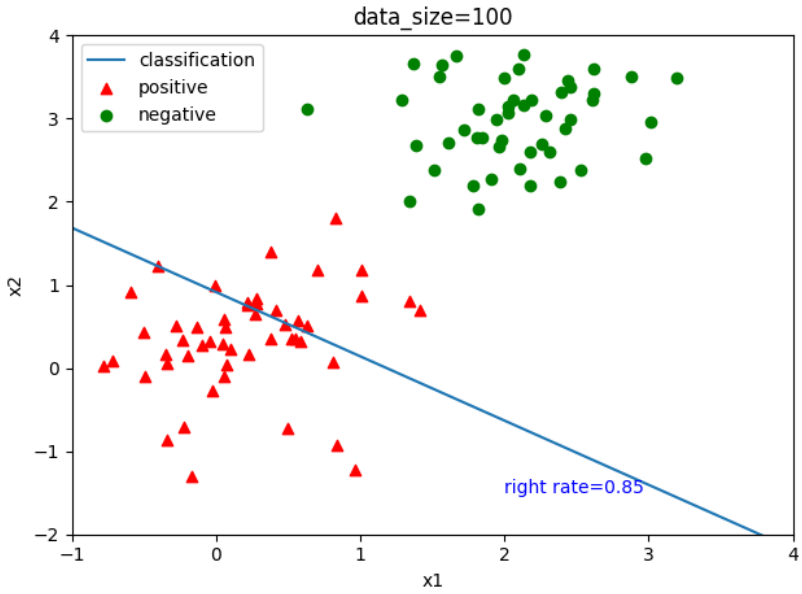
迭代 200 次：



迭代 300 次：



迭代 500 次：



从中可以当迭代次数过多时，会出现震荡现象，反而学习率会下降。

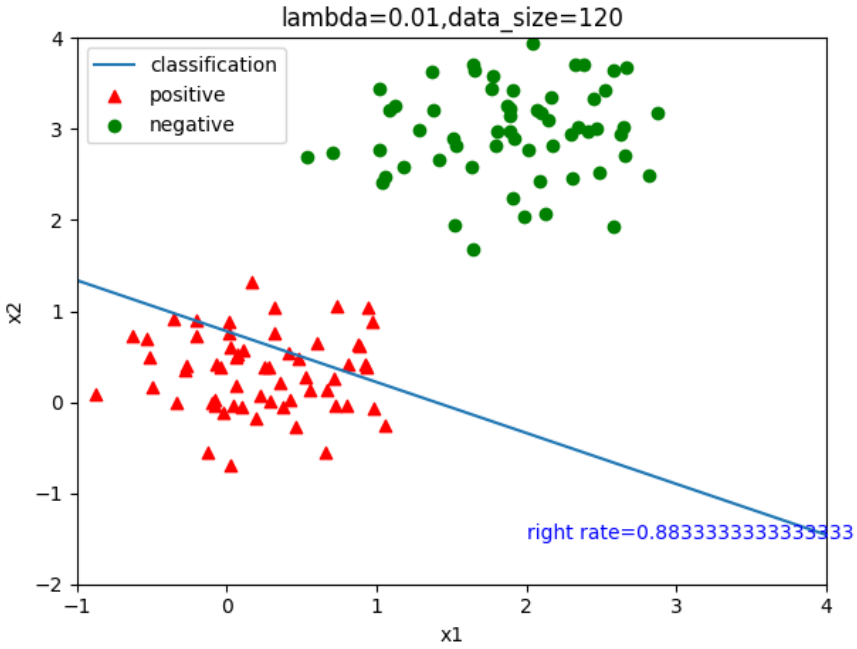
通过比较来看，迭代 200 或者 300 次结果比较好。

5.2 带正则项

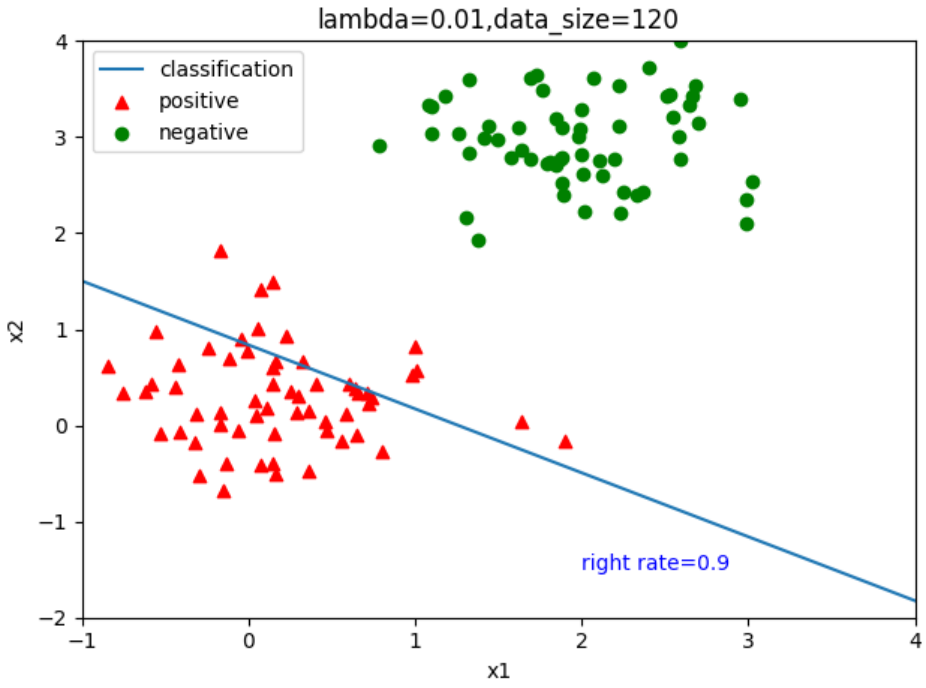
训练集大小为 120,选定$\lambda = 0.01$时，迭代 300 次的结果较好。因此在比较不同的λ对训练效果的影响时，选定迭代次数为 300。

比较不同的迭代次数的影响的结果如下

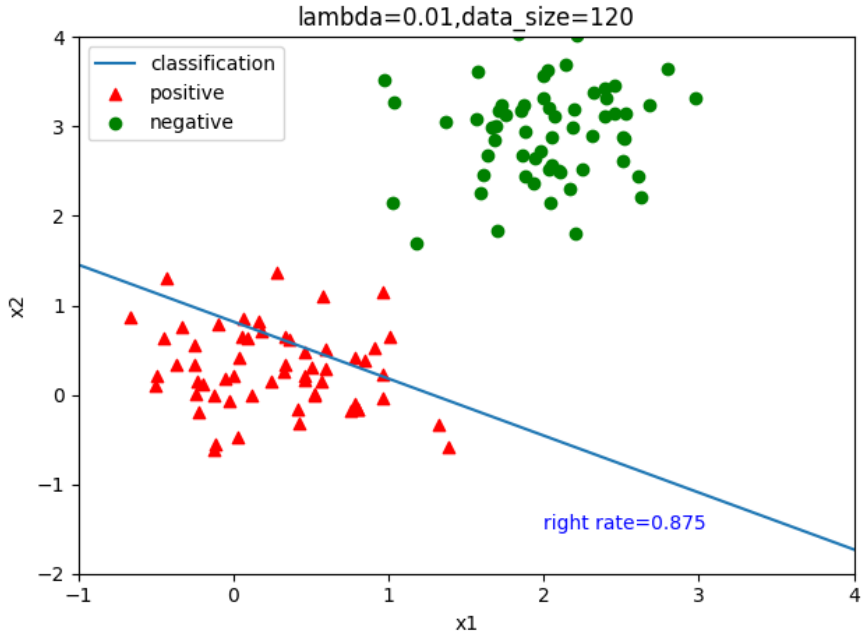
迭代 200 次



迭代 300 次

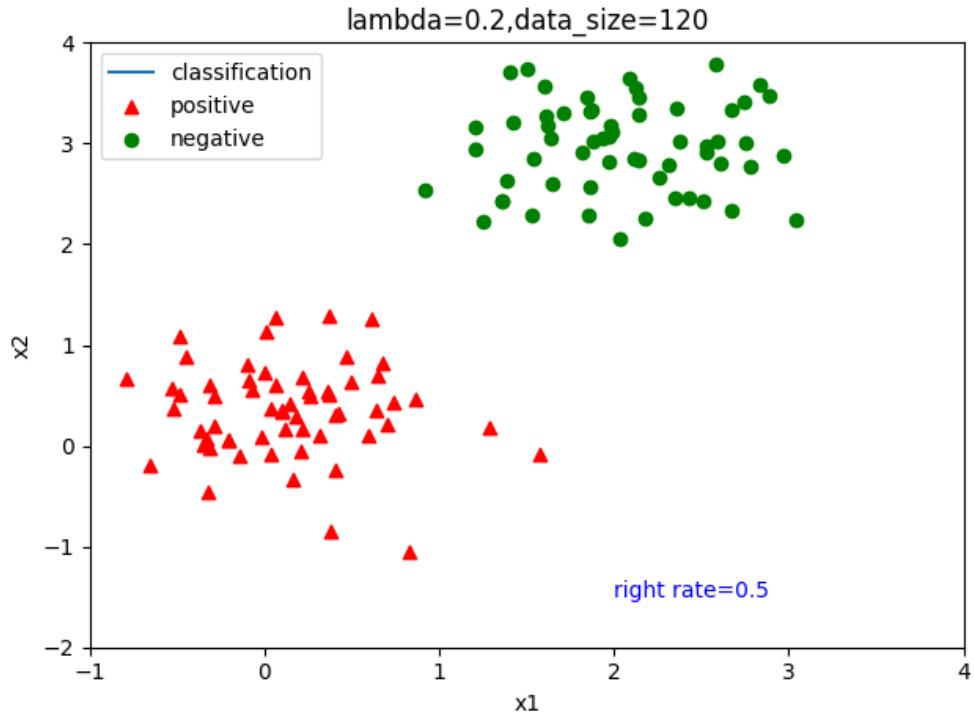


迭代 400 次

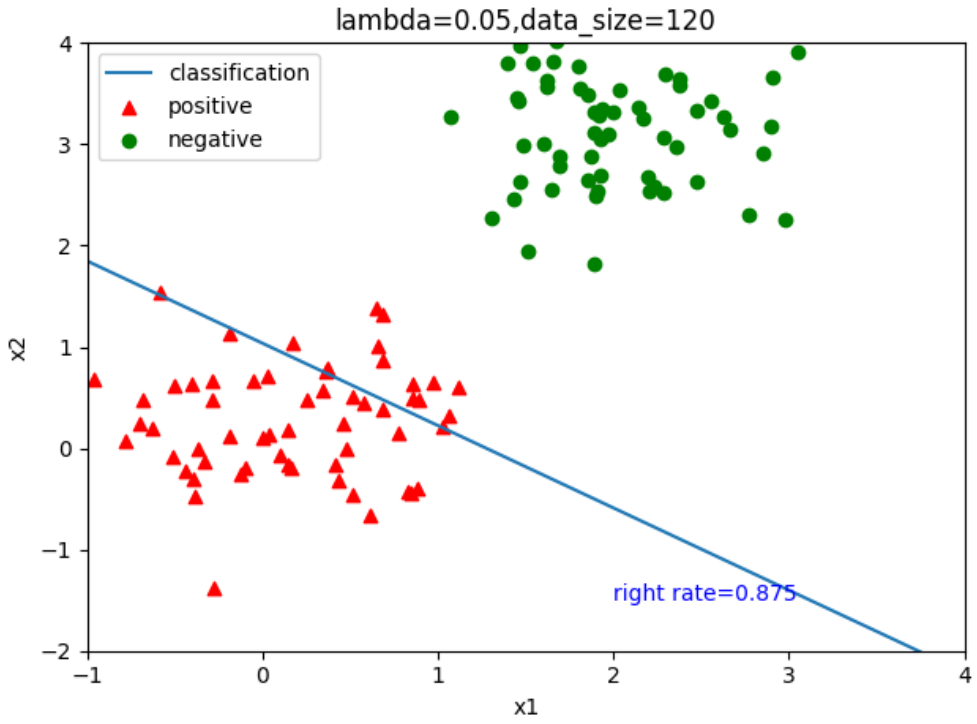


比较不同的λ对于训练结果的影响(选定迭代次数为 300)

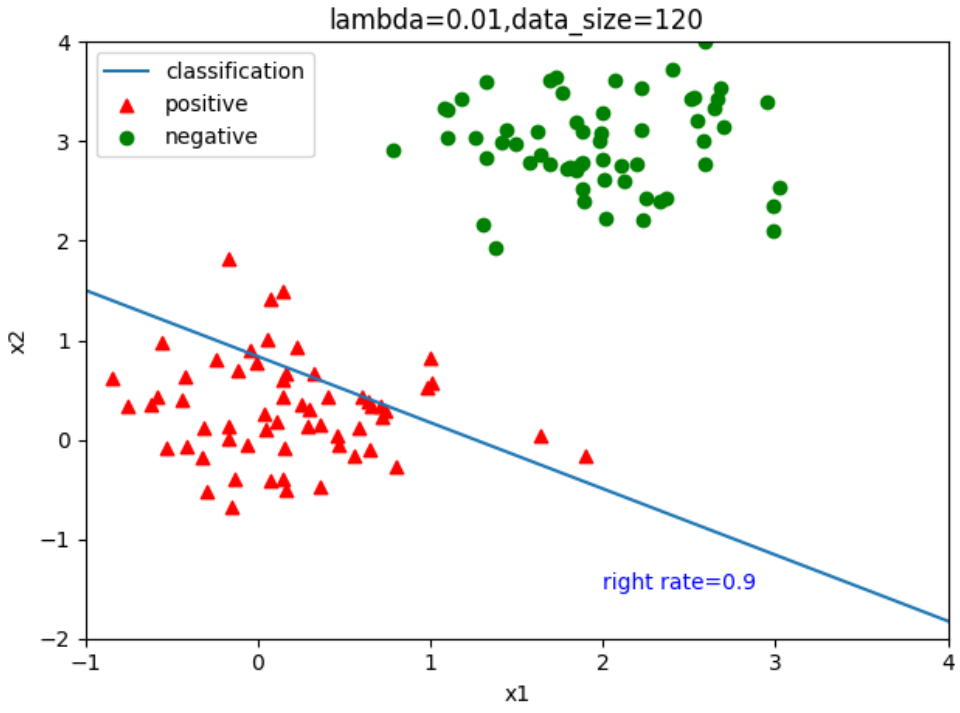
$\lambda = 0.2$:



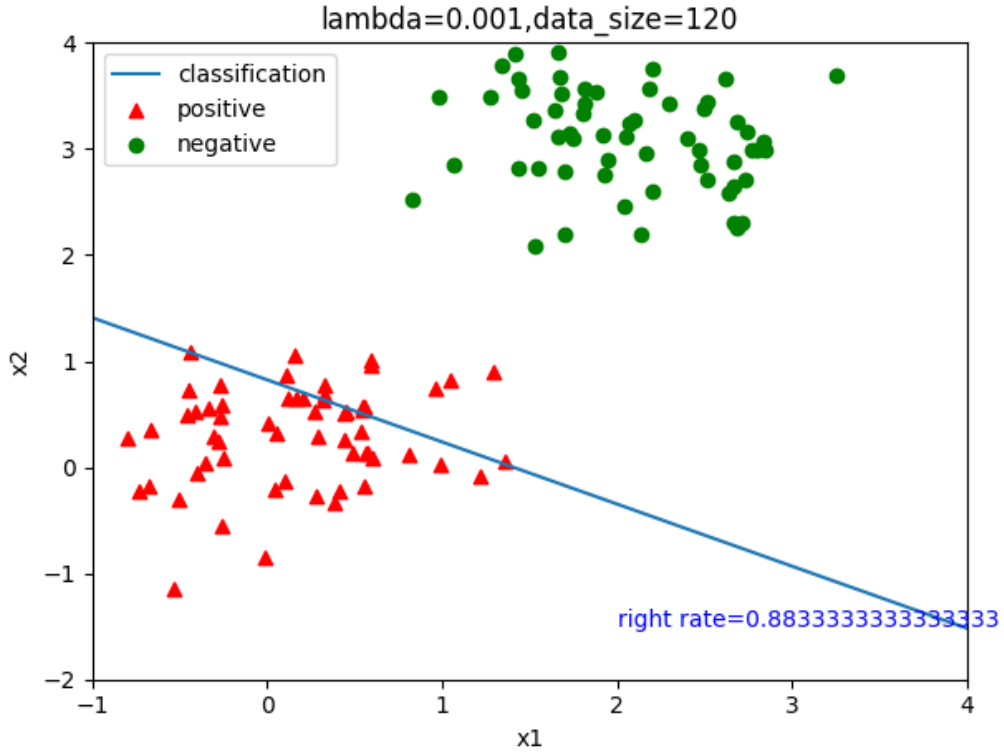
$\lambda = 0.05$:



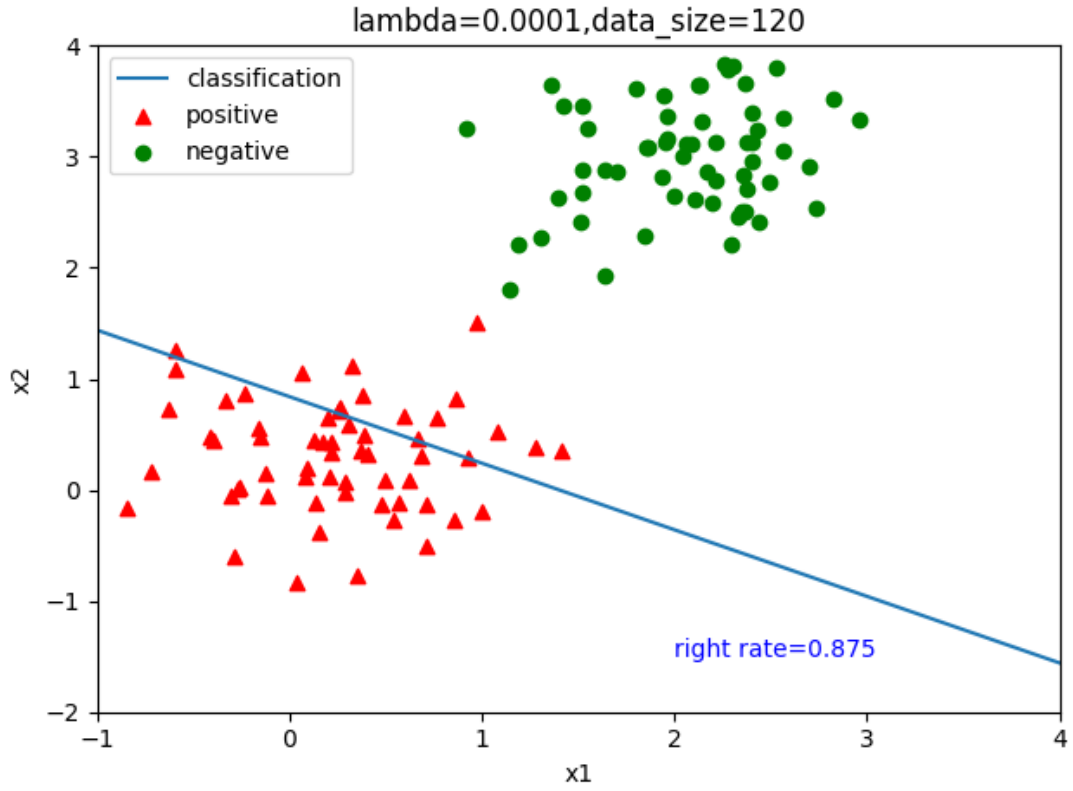
$\lambda = 0.01$:



$\lambda = 0.001$:

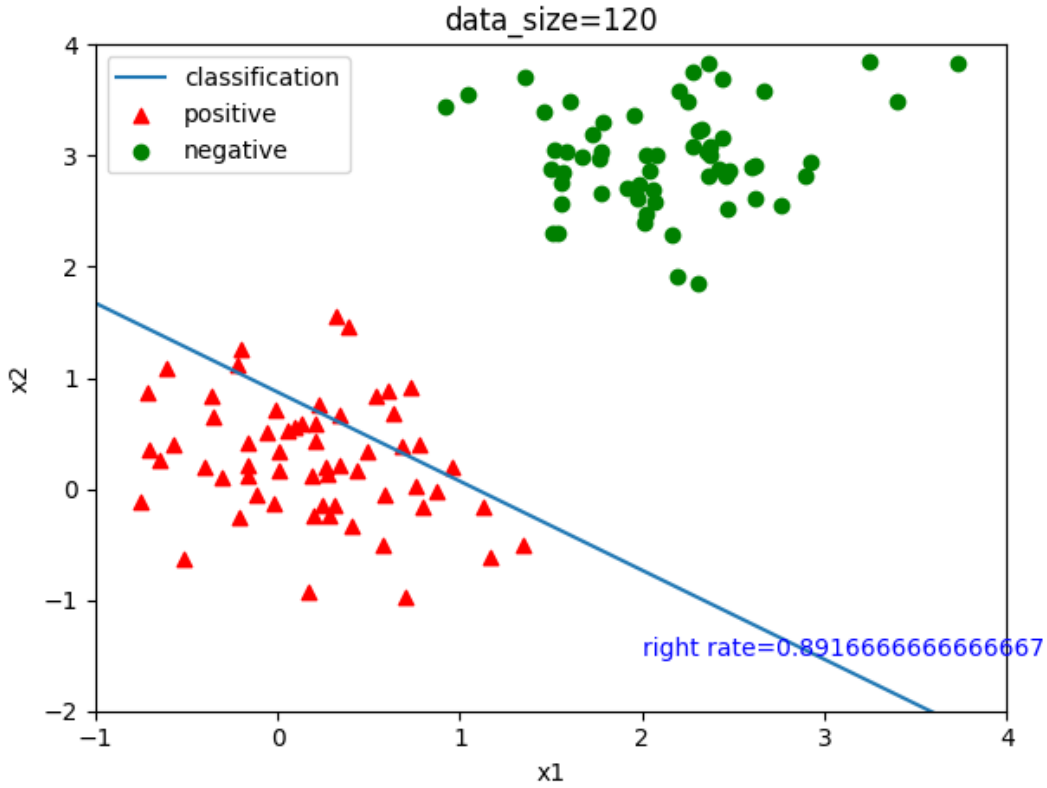


$\lambda = 0.0001$:



比较正则项对于准确率的影响：

不带正则项，120 个数据，迭代 300 次的结果为：



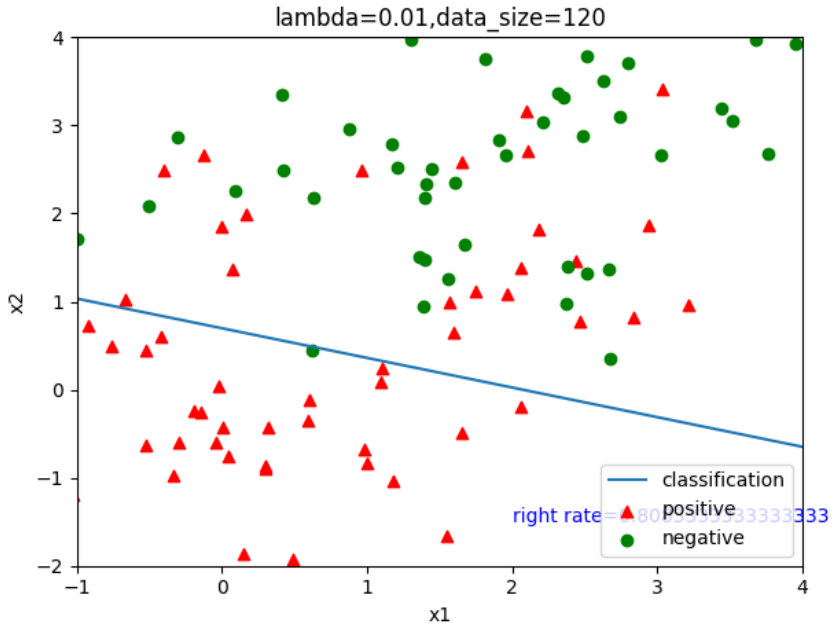
对于携带正则项的 0.9 的准确率来说，λ对于准确率的影响不大。

5.3 测试不满足朴素贝叶斯假设的数据

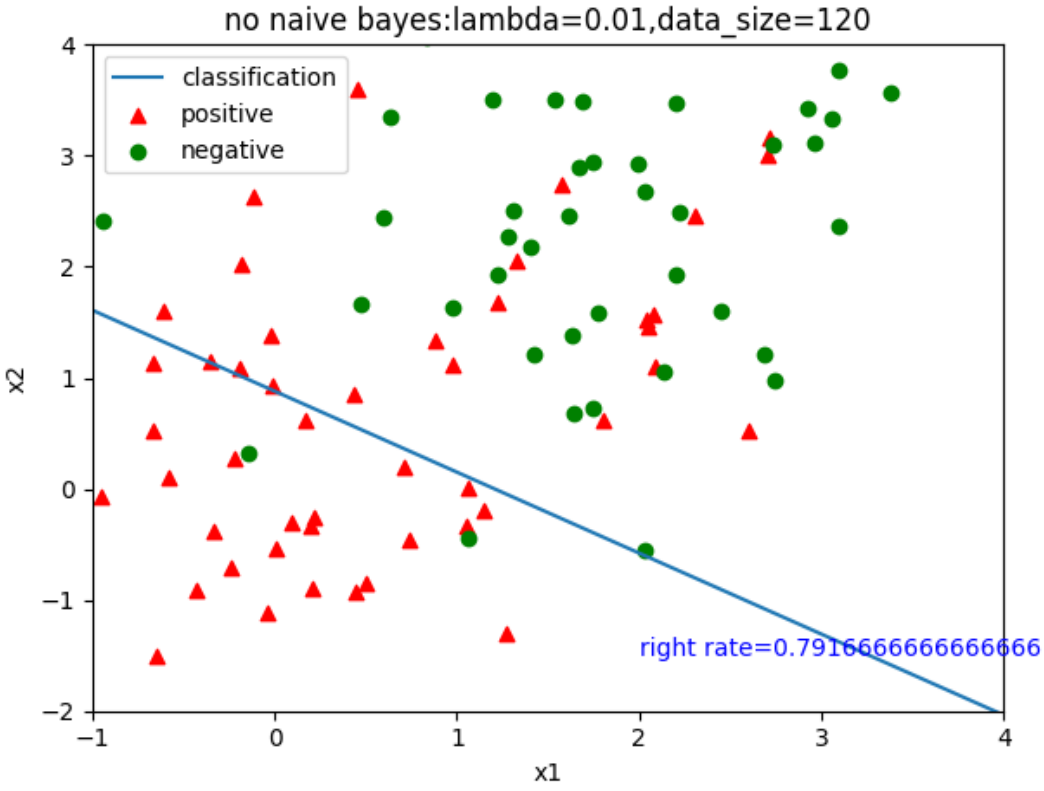
对于不满足朴素贝叶斯假设的数据测试结果

采用了带正则项的计算方式，$\lambda = 0.01$,由于数据不同，因此可能对应的最佳迭代次数也不同

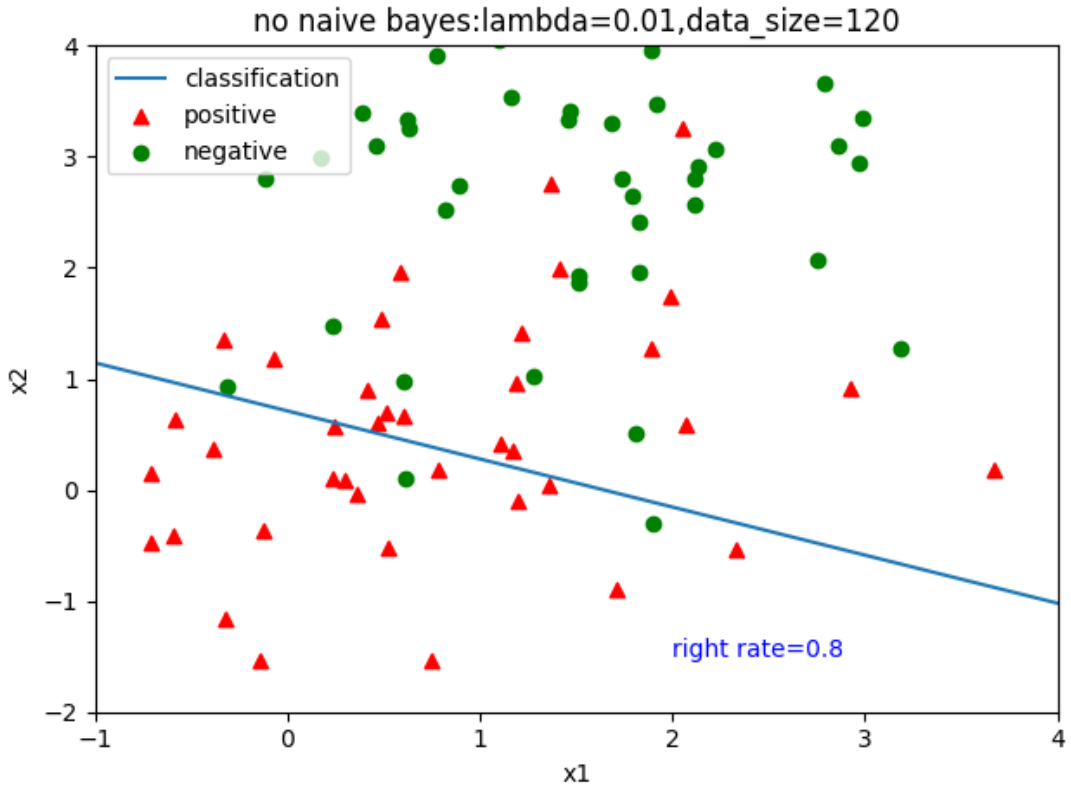
迭代 200 次：准确率为 0.808



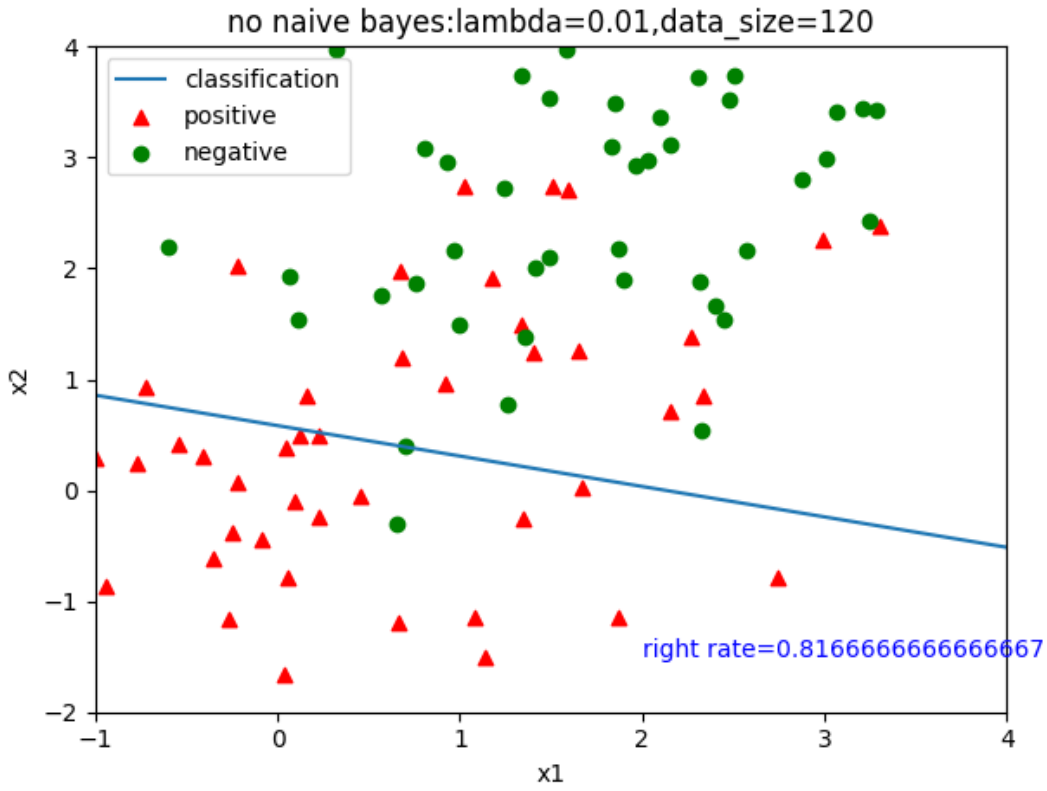
迭代 300 次，准确率为 0.792



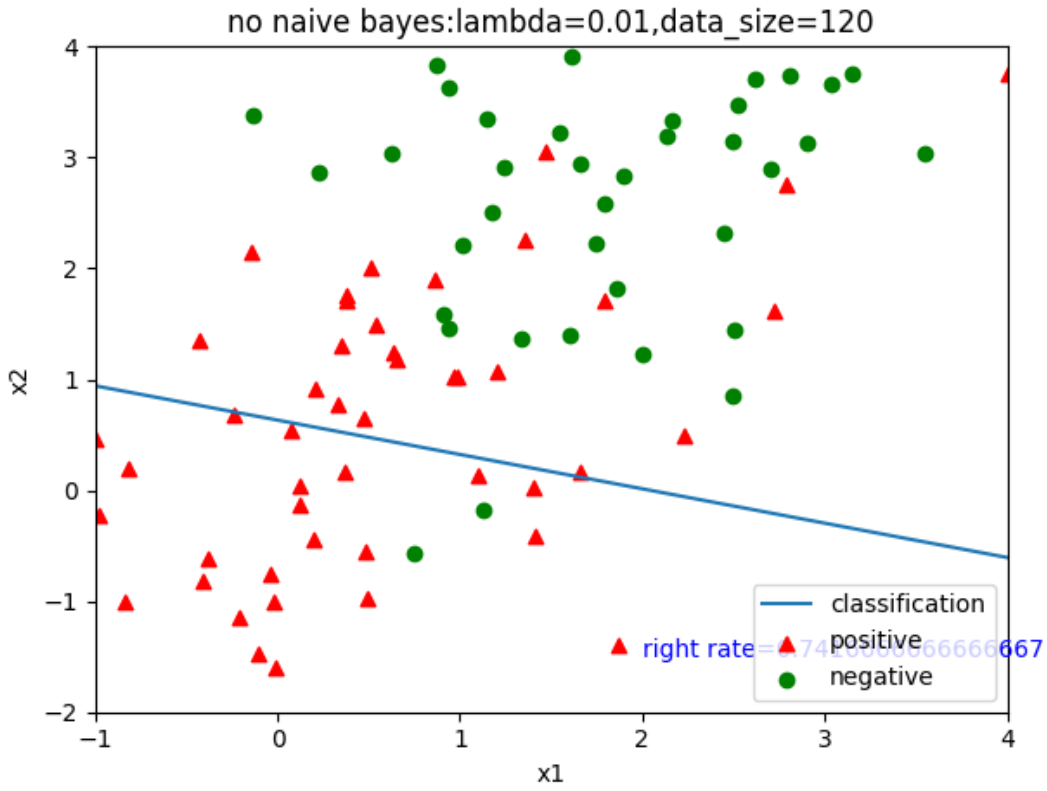
迭代 400 次，准确率为 0.8



迭代 500 次，准确率为 0.817



迭代 600 次,准确率为 0.7412



从中可以看出，对于不满足朴素贝叶斯假设的数据，实验的结果差一些，没有到达 90%准确率的迭代次数，但是仍然属于可以接受的范围。

5.4 采用 UCI 的数据进行测试

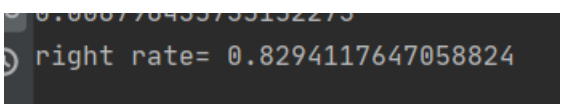
采取的数据网址为

http://archive.ics.uci.edu/ml/datasets/Divorce+Predictors+data+set

是一组根据 54 个问卷调查的结果和实际的离婚状况进来行预测是否会离婚的数据集。

测试结果如下：

迭代 1000 次：



这个结果在可以接受的范围内。

6 结论

(1) 对于逻辑回归来说，采用梯度下降法具有震荡现象，并不是越多的迭代次数，结果就越好。
(2) 对于不满足朴素贝叶斯假设的数据，得出的分类准确率仍然可以接受。
(3) 在实验中 sigmod 函数容易溢出，可以将数据归一化来尽量避免溢出。
(4) 对于逻辑回归而言，使用梯度下降时，惩罚项对于准确率的影响并不大。没有在多项式拟合函数时正则项对于拟合结果的影响大。
(5) 实验中采用矩阵形式的数据结构比 narrays 和 list 等等类型的数据结构更易用。