教程 1 : Logistic Regression

提交日期：2020.7.11

提交人：詹紫琦

目录

[1. 题目 3](#_Toc430945232)

[1.1线性可分/不可分的数据集上的logistic regression 3](#_Toc875547529)

[1.2 运行环境 3](#_Toc1551785815)

[2. 算法阐述或实验步骤说明 3](#_Toc1814153117)

[2.1算法阐述 3](#_Toc421670800)

[2.2 实验步骤说明 4](#_Toc1074061444)

[3. 实验结果与截图 5](#_Toc812228170)

[3.1 线性可分数据集逻辑回归实验结果 5](#_Toc181439737)

[3.2 线性不可分数据集逻辑回归实验结果 7](#_Toc241314197)

[4. 总结 10](#_Toc1161692404)

[4.1 总结 10](#_Toc930509947)

[5. 参考文献 10](#_Toc1238396788)

# 题目

## 1.1线性可分/不可分的数据集上的logistic regression

逻辑回归应用于分类问题中，针对该线性可分/不可分数据集而言，类别总共有两种，第一种类别的标签值为0，第二种类别的标签值为1。通过逻辑回归算法对训练集进行模型拟合，最终得到一个分类器。该分类器能实现的功能如下：对应一个输入，能输出一个通过该分类器分类完的所属类别。

## 1.2 运行环境

系统：Ubuntu20.04,python3.7,Anaconda集成工具Jupyter编写。

# 算法阐述或实验步骤说明

## 2.1算法阐述

逻辑回归[1]（Logistic Regression）虽然带有回归二字，但究其根本其实是属于分类任务，作为监督学习中的两大类任务之一，分类任务的目的是通过训练数据拟合出一个模型来形成一个分类器，并通过该分类器实现分类任务。

在这里，我们定义训练集数据为)。其中每一个都对应n个特征值的一个n维列向量。定义假设的向量参数为，逻辑回归中的假设函数的定义如下：

(2-1) X)

其中函数定义为：

(2-2)

进行了假设函数的定义之后，定义损失函数，为进行计算假设预测值与实际值的误差值大小，规定当时预测=1，当时预测=0。经过预测结果的定义之后得到的损失函数定义为：

(2-3)

在这里我们用替代。得到了损失函数后的目标就是要改变参数的值，来最小化损失函数。在这里我们一般使用梯度下降方法。由于凸函数存在全局最小值，随机选取一个点，沿梯度方向下降最快，不断的对求偏导，并同步更新的值，直到损失函数收敛到全局最小值。

梯度下降算法如下：

Repeat until Cost Function convergence

{

;

}

通过设置学习率和迭代次数进行梯度下降，最终可以得到一个使损失函数值最小的对应值，成功拟合出的模型构建成该分类器。

## 2.2 实验步骤说明

首先分析数据，将训练集中数据进行分类可视化，将不同类别的数据进行展示，得到的训练集数据如图2-1，其中图中蓝色散点分布数据为Y值为1的数据，图中红色散点分布数据为Y值为0的数据。横纵坐标分别代表着两个特征。

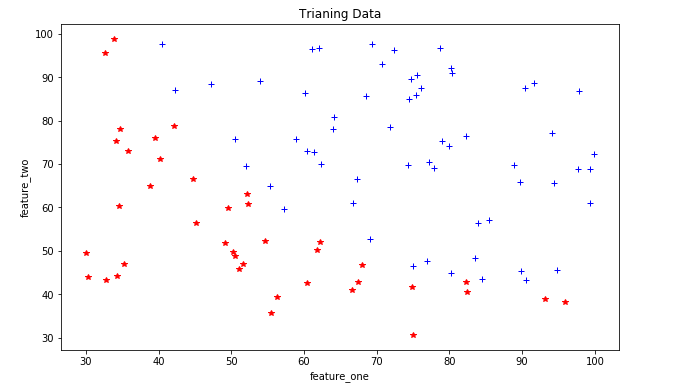


图2-1 线性可分训练集数据可视化

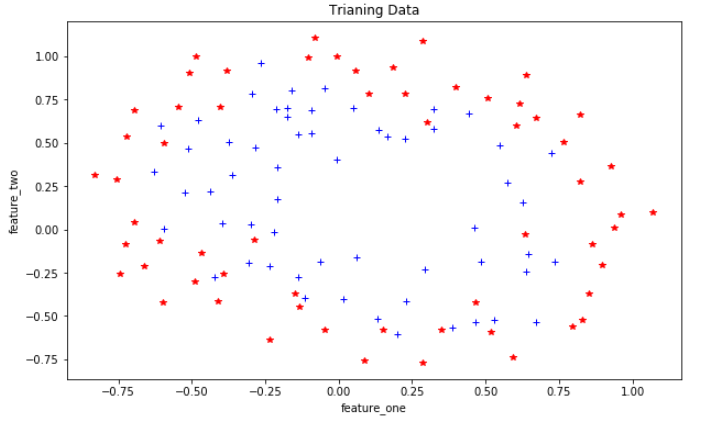


图2-2 线性不可分训练集数据可视化

接下来进行模型初步建立。首先是函数的建立，然后建立假设函数，然后建立逻辑回归的损失函数。然后初始化向量的值，初始化梯度下降的学习率，以及设置迭代次数。

进行了各数据的向量化之后，然后构建梯度下降函数，同时记录迭代过程中不同的取值，以及相对应的损失函数的大小用于实验结果分析。对于线性不可分数据集而言，需要构建特征多项式，设置深度degrees，得到对应的特征多项式，与公式2-1不同，这里的假设函数中会包含变量（特征项）X1和X2的高次幂或者是累乘项。完成以上步骤后，然后进行该迭代次数下的梯度下降，最终选取实验结果较好的梯度下降学习率以及迭代次数。最后得到损失函数与迭代次数相关联系的图像，最终得到一个逻辑回归分类器，并能在训练数据集通过决策边界表示出来。

# 实验结果与截图

## 3.1 线性可分数据集逻辑回归实验结果

经过不同的学习率与迭代次数对比，得到的实验结果在学习率设置为0.00001,以及迭代次数设置为2000000时误差较小，得到的决策边界如图3-1。

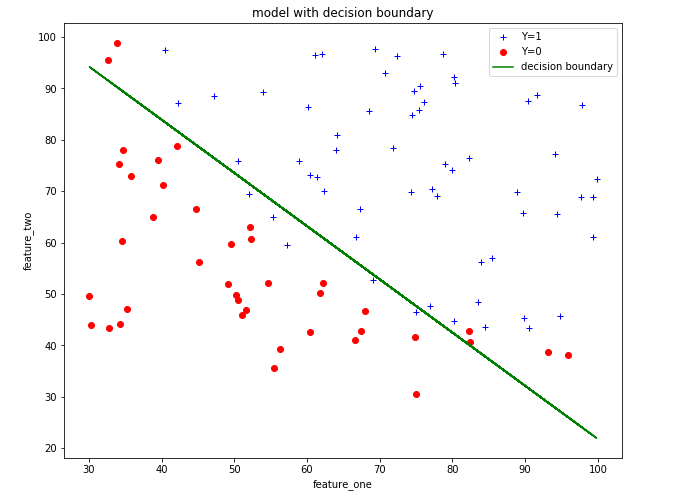


图3-1 线性可分数据集下的决策边界

得到的损失函数大小随迭代次数变化的图像如图3-2。

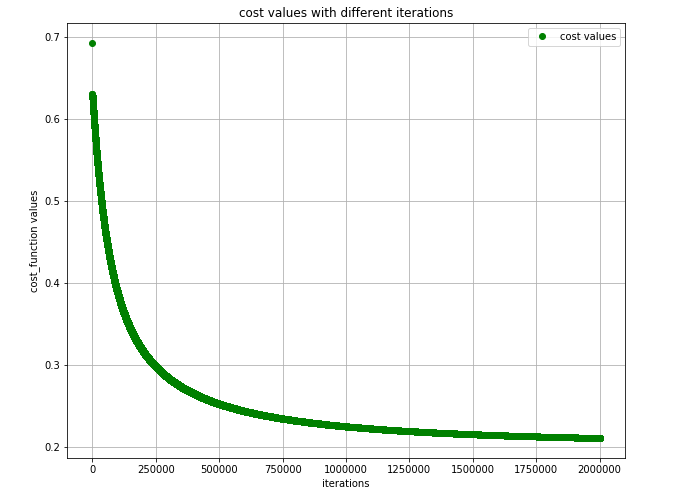


图3-2 不同迭代次数下损失函数的值

最终得到的该分类器的损失函数值为0.21041。

## 3.2 线性不可分数据集逻辑回归实验结果

对于线性不可分数据集中可以调整的参数包括构建特征多项式的深度degree，以及梯度下降过程中的学习率和迭代次数，通过设置不同的参数能得到不同的结果。

如表3-1是特征多项式的深度为2时的情况。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Learning rate  iteration | 200 | 2000 | 20000 | 200000 | 2000000 |
| 0.001 | 0.74254 | 0.51734 | 0.41224 | 0.41076 | 0.41076 |
| 0.0001 | 0.80709 | 0.74240 | 0.51752 | 0.41224 | 0.41076 |
| 0.00001 | 0.81669 | 0.80705 | 0.74239 | 0.51754 | 0.41224 |

表3-1 degree=2的情况下损失函数的大小

从表可以看出当学习率设置为0.001,迭代次数刚到200000时就能使损失函数梯度下降收敛到全局最小值0.41076。

如表3-2是特征多项式的深度为4时的情况。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Learning rate  iteration | 200 | 2000 | 20000 | 200000 | 2000000 |
| 0.001 | 0.59739 | 0.45679 | 0.39745 | 0.38495 | 0.38108 |
| 0.0001 | 0.77455 | 0.65740 | 0.45689 | 0.39745 | 0.38495 |
| 0.00001 | 0.81139 | 0.77441 | 0.65739 | 0.46897 | 0.39745 |

表3-2 degree=4的情况下损失函数的大小

从表可以看出当学习率设置为0.001,迭代次数达到2000000时就能使损失函数梯度下降收敛到全局最小值0.38108。下面观察一下特征多项式深度为6时的情况。

如表3-3是特征多项式的深度为6时的情况。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Learning rate  iteration | 200 | 2000 | 20000 | 200000 | 2000000 |
| 0.001 | 0.67560 | 0.45974 | 0.38776 | 0.36166 | 0.33890 |
| 0.0001 | 0.79027 | 0.67537 | 0.45983 | 0.38777 | 0.36167 |
| 0.00001 | 0.81448 | 0.79017 | 0.67534 | 0.45984 | 0.38777 |

表3-3 degree=6的情况下损失函数的大小

从表可以看出当学习率设置为0.001,迭代次数达到2000000时就能使损失函数梯度下降收敛到全局最小值0.3389。

根据上述的实验数据可知，在设置特征多项式深度为6,梯度下降学习率为0.001,迭代次数为2000000时能得到最小损失函数，损失函数值为0.3389。得到的损失函数随迭代次数变化曲线如图3-3。最后得到的逻辑回归分类器形成的决策边界如图3-4

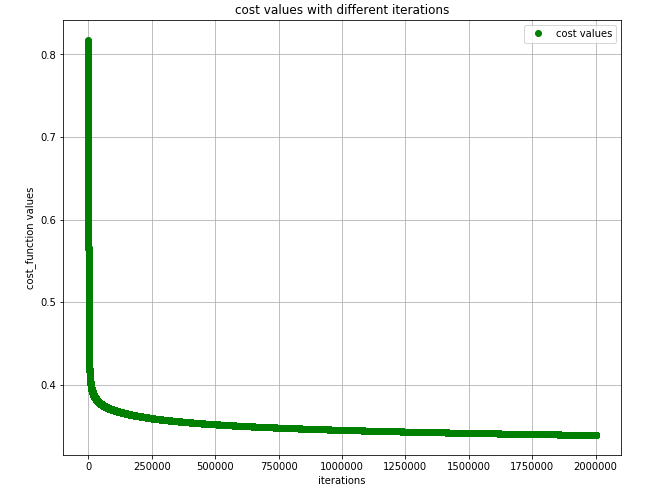


图3-3 随迭代次数变化的损失函数值

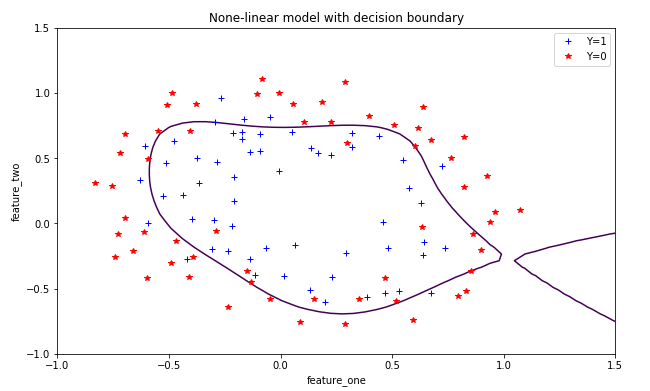


图3-4 非线性可分数据逻辑回归分类器的决策边界

# 总结

## 4.1 总结

逻辑回归算法应用于线性可分数据集分类时，能迭代较少的次数就能使损失函数收敛到全局最小值。而逻辑回归算法应用于线性不可分数据集分类时，相对需要迭代较多的次数才能使损失函数收敛到全局最小值。同时在构建特征多项式时，特征的幂次（即深度）较大时，得出的有效模型能更好的进行分类处理。

# 参考文献

[1] 周志华等.机器学习（西瓜书）