**Logistic回归**

03

分类是一种常见的监督机器学习问题，其在很多领域均有广泛应用。典型应用包括垃圾邮件过滤、金融风控预测、广告点击率预估等。

本章我们将讨论一种线性分类器：Logistic回归。同线性回归模型类似，我们也从模型的形式、目标函数（损失函数和正则函数）、目标函数的优化求解、分类模型的性能评价指标、Scikit-Learn中Logistic回归模型的API，以及应用案例等方面展开讨论。由于Logistic回归实现简单、可解释性强，Logistic回归模型在推荐系统中使用频率极高。

## **标题2** 3.1 **Logistic**回归模型

Logistic回归（Logistic Regression）虽然从名字上来看是回归算法，但其实际上是一个分类算法，亦被称为Logit 回归（Logit Regression）、最大熵分类器（Maximum-Entropy Classification，MaxEnt）。

同线性回归模型类似，Logistic回归首先对输入进行线性组合，得到。由于模型输出为样本属于某个类别的概率，我们使用函数将压缩到[0,1]之间，用表示概率分布的参数。

函数的形式为：

|  |  |
| --- | --- |
|  | （3-1） |

其导数为：

|  |  |
| --- | --- |
|  | （3-2） |

函数呈S型，亦被称为S型函数，其形状如图3-1所示。

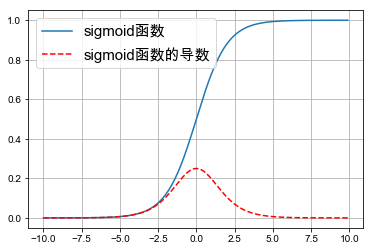


图3-1 Sigmoid函数及其导数

对两类分类问题，Logistic回归模型为：

|  |  |
| --- | --- |
|  | （3-3） |

定义一个事件的几率（odds）为该事件发生的概率与不发生的概率的比值，在Logistic回归模型中，事件的几率为：

|  |  |
| --- | --- |
|  | （3-4） |

两边取log运算，得到该事件发生的对数几率（log odds）为

|  |  |
| --- | --- |
|  | （3-5） |

所以Logistic回归是对事件发生的对数几率采用线性回归进行拟合。线性回归和Logistic回归的区别和联系如图3-2（a）和如图3-2（b）所示。

|  |  |
| --- | --- |
|  |  |
| （a）线性回归 | （b）Logstic回归 |

图3-2 线性回归与Logistic回归示意图

当，，如果取最大后验概率，的类别；

当，，此时取最大后验概率，的类别取；

当，则，的类别和的概率相等，此时位于决策边界上，可将任意分类到某一类，或者拒绝做出判断。

分类器将输入空间X划分为一些互不相交的区域，这些区域的边界称为决策边界（Decision Boundaries）。分类器为每个类别分配一个判别函数，根据判别函数来判断一个样本是否是该类别。假设有个类别，那么分类器会得到个判别函数。令，一般就可以认为样本属于第 类。判别函数和相等的点集即为决策边界。判别函数的形式不同，会使得决策边界或光滑、或粗糙。如果决策边界是的线性函数，称为线性决策边界，形成的分类器是线性分类器。Logistic回归是一个线性分类器，因为决策边界

|  |  |
| --- | --- |
|  | （3-6） |

是的线性组合。

上述两类分类的Logistic 回归与神经元的工作机制相同。当输入的线性组合大于阈值时，神经元发放脉冲。因此Logistic 回归可视为具有单个神经元的单层神经网络（没有隐藏层）。 第8章我们讨论的深度神经网络，可视为是Logistic回归单元的网络集合。

在如例1-4的鸢尾花数据集上，取2维特征（花萼长度、花萼宽度）、将鸢尾花分成山鸢尾（setosa）和非山鸢尾两类，Logistic回归模型的决策边界如图3-3所示，分界线的表达式为。

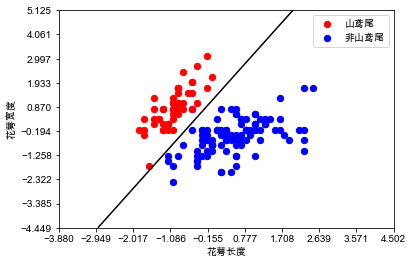


图3-3 Logistic回归在Iris数据集上的决策边界

**说明**

周志华老师的《机器学习》[5]书中采用意译，称Logistic回归为对数几率回归。在此我们直接用英文单词Logistic。Logistic回归用线性回归模型回归对数几率。

## **标题2** 3.2 **Logistic**回归模型的目标函数

同线性回归类似，Logistic回归模型的目标函数也包含两部分：训练集上的损失和正则项。其中损失函数度量模型预测值与真实值之间的差异，正则项惩罚模型的复杂度。Logistic回归的目标函数记为：

|  |  |
| --- | --- |
|  | （3-7） |

其中表示模型预测样本的标签为1的概率。

同线性回归类似，Logistic回归模型中正则项可为：

* L2正则：；
* L1正则：；
* L2正则+L1正则：。

如果当正则参数取合适值时，L1正则的解是稀疏的，可以起到特征选择的作用。需要说明的是，Logistic回归的目标函数必须包含正则项，否则当数据完全线性可分时，权重系数的模长会无限大。

同线性回归类似，Logistic回归模型的损失函数也取负log似然损失。但Logistic回归模型中数据的概率分布不同，因此负log似然损失的具体形式与线性回归不同。

对于两类分类问题，。给定，可以用贝努利（Bernoulli）分布表示。在贝努利分布中，参数表示在给定的情况下的概率：

|  |  |
| --- | --- |
|  | （3-8） |

在Logistic回归模型中，参数，因此负log似然损失为：

|  |  |
| --- | --- |
|  | （3-9） |

该损失函数亦被称为交叉熵损失。当和的交叉熵损失函数如图3-3所示。

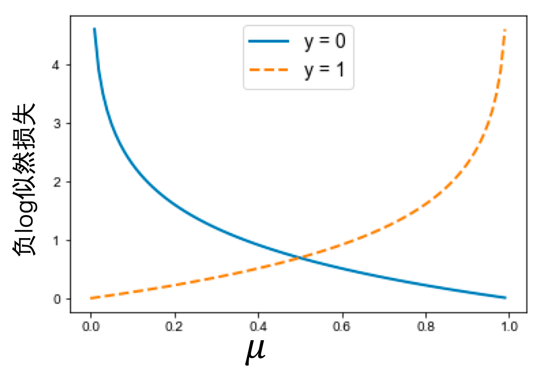


图3-4 交叉熵损失函数示意图。其中横轴为模型预测标签为1的概率，纵轴为对应的交叉熵损失。

**说明**

如果对于同一个随机变有两个概率分布和，可用KL散度（Kullback-Leibler divergence）用于表示两个概率分布之间的差异：

式中第一项表示分布的信息熵，与无关，第二项为和为交叉熵。

在机器学习中，用表示样本的真实分布（对给定样本，是给定的），表示模型所预测的分布。所以和的KL散度越小，交叉熵越小，表示模型预测效果越好。

若用表示样本标签的取值，模型预测标签为1的概率为，模型预测标签为的概率为，则此时交叉熵为：。

若用表示样本标签的取值，模型预测标签为1的概率为，模型预测标签为的概率为，所以综合两种情况，可得到，此时交叉熵为：。

## **标题2** 3.3 **Logistic**回归目标函数优化求解

Logistic回归模型的目标函数没有解析解。如果采用L2正则，目标函数二阶可导，可采用梯度下降法或牛顿法迭代求解。如果采用L1正则，目标函数在原点处不可导，可采用坐标轴下降法求解。由于正则部分的梯度同线性回归，下面我们主要讨论损失函数的梯度和海森矩阵。

### **3.3.1** 梯度下降法

令，Logistic回归中训练集上的损失和为：

|  |  |
| --- | --- |
|  | （3-10） |

函数对的梯度为：

|  |  |
| --- | --- |
|  | （3-11） |

其中

|  |
| --- |
|  |
|  |
|  |

因此L2正则的Logistic回归的梯度为：

|  |  |
| --- | --- |
|  |  |

梯度下降法参数更新公式为：

|  |  |
| --- | --- |
|  | （3-12） |

### **3.3.2** 牛顿法

#### 1．牛顿法

梯度下降法实现相对简单，但其收敛速度较慢。在最小值附近，梯度下降法以一种曲折的慢速方式来逼近最小点，此时考虑牛顿法或拟牛顿法。

假设函数在处二阶可导，则对任意小的，函数的二阶泰勒展开近似为：

|  |  |
| --- | --- |
|  | （3-13） |

其中为函数对的梯度，为函数对的海森矩阵。

我们找最佳的，使得最小。该最小值可通过对上述函数求一阶导数并等于0得到：

从而

|  |  |
| --- | --- |
|  | （3-14） |

对比梯度下降法的式（3-12），可用发现，牛顿法用代替梯度下降法中的学习率，即在原函数的点处用一个二次函数近似原函数，然后用这个二次函数的极小值点作为原函数的下一个迭代点。若原函数本身是一个二次函数，则牛顿法一步就能到达极小点或鞍点。

**算法**3-1**：牛顿法**

1．随机选取起始点；

2．计算目标函数在该点的一阶导数和海森矩阵，若足够小，返回为最佳参数；

3. 计算搜索方向：

4．更新： ；

5. ，转第2步。

由于牛顿法用到了函数的二阶导数，亦被称为二阶优化方法。与梯度下降法相比，牛顿法收敛速度更快。但每一轮迭代中，还需要计算海森矩阵的逆，所以每次迭代的计算量比梯度下降大。实际实现时一般不直接求海森矩阵的逆矩阵，而是求解如下方程组：

得到。求解这个线性方程组一般使用迭代法，如共轭梯度法。

牛顿法并不能保证每一步迭代时函数值下降，也不保证一定收敛。若初始点 充分靠近极值点，极值点的海森矩阵非奇异，并且海森矩阵在极值点附近利普希茨（Lipschitz） 连续，则牛顿法具有二阶收敛性。如果不满足上述条件，牛顿发可能数值计算失败或产生数值不稳定。为此，研究者们提出了一些补救措施，其中的一种是直线搜索（line search）技术，即搜索最优步长。具体做法是让取一些典型的离散值，如0.0001,0.001,0.01等，比较取哪个值时函数值下降最快，作为最优步长，即。

#### 2．Logistic回归的牛顿法求解

根据式（3-11），Logistic回归模型的损失函数部分的梯度为

其海森矩阵为：

|  |  |
| --- | --- |
|  | （3-15） |

其中为对角矩阵，第个对角线元素为。由于，所以为正定矩阵，Logistic回归有唯一的全局最优解。

若只考虑损失函数项，牛顿法参数更新公式为：

|  |  |
| --- | --- |
|  | （3-16） |

令，则

|  |  |
| --- | --- |
|  | （3-17） |

对比最小二乘线性回归的目标函数权重求解公式

可以看出，损失函数迭代公式相当于对数据施加权重（OLS公式中的 用代入，用代入，即可得到Logistic回归的权重迭代公式），然后再进行最小二乘求解。因此上述方法亦被称为迭代重加权最小二乘（Iteratively reweighted least squares，IRLS）。

L2正则的梯度为，海森矩阵为单位矩阵，所以L2正则的Logistic回归的牛顿法参数更新公式为：

|  |  |
| --- | --- |
|  | （3-18） |

与L2正则不同，无法计算L1正则项的梯度和海森矩阵，但可以得到稀疏解。结合上述IRLS求解法，L1正则的Logistic回归的每次迭代中，可转换为一个加权的Lasso问题求解：

|  |  |
| --- | --- |
|  | （3-19） |

**说明**

最小二乘线性回归和加权最小二乘线性回归对比：

|  |  |  |
| --- | --- | --- |
| 比较内容 | 最小二乘线性回归 | 加权最小二乘线性回归 |
| 目标函数 |  |  |
| 解析解 |  |  |

### **3.3.3 拟**牛顿法

#### 1．拟牛顿法条件

由于牛顿法需要计算海森矩阵的逆，计算量较大。随着未知数维度增大，海森矩阵（）会越大，需要的存储空间增多，计算量增大，有时候甚至大到不可计算。

拟牛顿法是一类算法的总称，目标是通过某种方式来近似表示海森矩阵（或者其逆矩阵），避免每次迭代都计算海森矩阵的逆，收敛速度介于梯度下降法和牛顿法之间。拟牛顿法虽然每次迭代不像牛顿法那样保证是最优化的方向，但是近似矩阵始终是正定的，因此算法始终是朝着最优化的方向在搜索。

假设经过次迭代后，目标函数在在点处进行二阶泰勒展开：

对公示（3-25）两边同取梯度算子，得到

|  |  |
| --- | --- |
|  | （3-18） |

上式取，得到

|  |  |
| --- | --- |
|  | （3-19） |

令，得到拟牛顿条件为：

|  |  |
| --- | --- |
|  | （3-20） |

拟牛顿条件对迭代过程中的海森矩阵进行约束。因此对海森矩阵的近似和海森矩阵的逆的近似，有

|  |  |
| --- | --- |
|  | （3-21） |

#### 2．BFGS

BFGS是一种拟牛顿法，它是由四个发明人（Broyden, Fletcher, Goldfar, and Shanno）的首字母组合命名，目前BFGS被证明是最有效的拟牛顿优化方法。

BFGS目标是用迭代的方式逼近海森矩阵。设逼近值为，那么希望通过计算达到目的（其中通常取单位矩阵）。

假设，并代入拟牛顿条件，得到

|  |  |
| --- | --- |
|  | （3-22） |

令，以及，可计算得到

|  |  |
| --- | --- |
|  | （3-23） |

综上，得到

|  |  |
| --- | --- |
|  | （3-32） |

牛顿法中需要计算海森矩阵的逆矩阵，根据Sherman-Morrison公式，可得到：

|  |  |
| --- | --- |
|  | （3-33） |

**算法**3-2**： BFGS**

1．初始化变量；

2．计算当前点的梯度，若，退出循环，返回；

3．确定搜索方向： ；

4．更新参数：；

5．令，，更新：

6. 令，转第2步。

这个更新方法跟牛顿法的区别是在更新参数之后更新一下近似海森矩阵，而牛顿法是在更新之前完全的计算一遍海森矩阵。还有一种从计算上改进BFGS的方法称为L-BFGS（Limited Memory BFGS），不直接存储海森矩阵，而是通过存储计算过程中产生的，从而减少参数存储所需空间。

## **标题2** 3.4 多类分类任务

上面我们讨论了两类分类任务的Logistic回归模型。对多类分类任务，一种方式是将其转化为多个两类分类任务。对每一个类别，正样本为本类样本，负样本为其他所有样本，训练一个两类分类器：

|  |  |
| --- | --- |
| 。 | （3-34） |

这种方式我们称之为1对其他（One-vs-Rest，OvR），如图3-5所示。每个两类分类器单独训练，都有自己正则参数和权重参数。

|  |  |
| --- | --- |
|  |  |
|  |
|  |
| （a）3类分类问题 | （a）3个两类分类问题 |

图3-5 采用1对其他（OvR）方式的多类分类器。图中所示为一个3类分类问题，转换为3个两类分类问题

1对其他多类分类方式的优点是普适性广，可以应用于输出值或者概率的分类器，效率高。但缺点是容易造成训练集中两类样本数量的不平衡，尤其在类别较多的情况下，会出现正类样本的数量远远不及负类样本的数量，从而造成分类器的偏向性。

另一种处理多类分类任务的方法是直接用多项分布对后验概率建模，类似用分布表示两类分布的后验概率。

是个人造词，由（多项）+（的后几个字母）组成。分布用于描述多类分类的概率分布，其参数为向量，其中，其中每一个分量表示第个状态的概率。我们用符号表示。经典概率书并没有将其作为一种概率分布单独表示，而是用多项（Multinomial）分布统称。有些参考书称其为广义Bernoulli分布，或离散分布。有些参考书称其为范畴分布（Categorical Distribution）。

类似多重贝努利试验成功次数输出用二项（Binnomial）分布表示，多重Multinoulli的输出用多项（Multinomial）分布表示。

将类别用独热编码（编码一个为维向量，，即当时，第维为1，其他元素均为0），记为向量。则Multinoulli分布的概率函数为

|  |  |
| --- | --- |
|  | （3-35） |

或者用标量表示为：

|  |  |
| --- | --- |
| ， | （3-36） |

其中为指示函数（Indicator function），当括号中的条件满足时，函数值为1，否则为0。

类似两类分类模型推导，假设概率可以由再经过函数变换得到，且，得到

|  |  |
| --- | --- |
|  | （3-37） |

上述等式右边为Softmax函数：

|  |  |
| --- | --- |
|  | （3-38） |

Softmax函数为函数的推广，将维向量的每个元素转换为[0,1]的数，且变换后元素之和为1。因此多类分类器又被称为Softmax分类器。对Softmax分类器，我们用表示Multinoulli分布的参数：

|  |  |
| --- | --- |
|  | （3-39） |

用表示标签向量的第个元素，表示模型预测概率向量的第个元素，则Softmax分类模型的负log似然损失函数为：

|  |  |
| --- | --- |
|  | （3-40） |

Softmax分类器的正则项类似两类分类器的正则项，包括个权重向量。

## **标题2** 3.5 分类任务的性能评价指标

分类任务中常用模型性能的评级指标包括：

#### 1．正确率（Accuracy）:

|  |  |
| --- | --- |
|  | （3-41） |

其中为指示示性函数，当括号中的条件满足时，函数值为1，否则为0。

正确率是最常用的分类性能指标，也是Scikit-Learn中默认的分类任务性能评价指标。但准确率只包含了某个样本是否分类正确，并没有包含样本正确/错误程度的信息。接下来的几个损失函数可以弥补这一点。

#### 2．交叉熵损失（logloss）：

|  |  |
| --- | --- |
|  | （3-42） |

其中为类别数目，为第个样本的标签的独热编码向量，为模型预测第个样本为类别的概率。

#### 3．合页损失（Hingeloss）：

|  |  |
| --- | --- |
|  | （3-43） |

合页损失是支持向量机中的损失函数，详见第4章。

#### 4．混淆矩阵（Confusion Matrix）

当不同类别数据不均衡时，上述正确率或错误率（损失）可能还不够。例如对一种罕见疾病设计医疗测试，假设每一百万人中只有一人患病。我们只需要让分类器一直报告没有患者，就能轻易地在检测任务上实现99.9999%的正确率。显然，正确率很难描述这种系统的性能，我们要关注的是那个极少量的标签。

对不均衡数据集的预测性能评估，我们有一个直观好用的工具：混淆矩阵。混淆矩阵的元素 表示真实分类是第类，但是预测值为第类的样本数目。所以混淆矩阵对角线上的元素值越大越好。

例3-1：某分类模型在手写体数字（0-9）识别任务上的的混淆矩阵如下：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 87 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 88 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 85 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 79 | 0 | 3 | 0 | 4 | 5 | 0 |
| 0 | 0 | 0 | 0 | 88 | 0 | 0 | 0 | 0 | 4 |
| 0 | 0 | 0 | 0 | 0 | 88 | 1 | 0 | 0 | 2 |
| 0 | 1 | 0 | 0 | 0 | 0 | 90 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 88 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 88 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 90 |

对两类分类问题，通常将要关注的类作为正类（比如欺诈交易），其他类作为负类（比如正常交易）。定义：

• （Positive）：正样本数目

• （Negative）：负样本数目

• （True Positive）：分类器将正类预测为正类的数量

• （False Negative)：分类器将正类预测为负类的数量

• （FalseA Postive）：分类器将负类预测为正类的数量，虚警样本数目

• （True Negative）：分类器将负类预测为负类的数量

从而得到两类分类问题的混淆矩阵如表3-1所示。

表3-1 两类分类问题的混淆矩阵

|  |  |  |  |
| --- | --- | --- | --- |
| 预测值  真实值 |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

基于表3-1，我们可以定义以下指标：

1. 正确率：，即预测正确的样本（和）在所有样本中占的比例。

在各类样本不均衡时，正确率不能很好表示模型性能，因为会出现大类占主导地位的情况，即大类正确率高，而少数类正确率低。这样情况下，需要对每一类样本单独观察。

1. 错误率： ，即被预测错误的样本在所有样本中所占比例。
2. 精度（查准率）：，即所有被预测为正例的样本中，多少比例是真的正例。
3. 召回率（查全率、真阳率、）：，即所有真的正例中，多少比例被模型预测出来了。
4. 假阳率：，即预测结果将多少负样本预测成了正样本。

不同的问题侧重不同的指标，有的侧重精确率，有的侧重召回率。对于推荐系统，因为给用户展示的窗口有限，必须尽可能的给用户展示其真实感兴趣的结果，因此更侧重精确率，即用户真正感兴趣的比例。对于医学诊断系统，更侧重于召回率，即疾病被发现的比例。因为疾病如果被漏诊，则很可能导致病情恶化。

精确率和召回率是一对矛盾的度量。一般来说精确率高时召回率往往偏低，而召回率高时精确率往往偏低。

• 如果希望将所有的正例都找出来（查全率高），最简单的就是将所有的样本都视为正类，此时有，但此时查准率低。

• 如果希望查准率高，则可以只挑选有把握的正例。最简单的就是挑选最有把握的那一个样本，此时有，但查全率就低（只挑出了一个正例）。

1. F1分数：精确率和召回率的调和平均。F1认为两者同等重要。

|  |  |
| --- | --- |
|  | （3-60） |

#### 5．ROC曲线与AUC值

上面我们讨论给定分类器的混淆矩阵，从而得（真阳率）和（假阳率）。给定分类器通常是通过对判别函数取某个阈值得到。如果我们不只关心一个阈值，而是想考察在一系列阈值上运行分类器，并画出和为阈值的隐式函数，这样我们得到接收者操作曲线（Receiver Operation Characteristic Curve，ROC）。

ROC的横坐标为，纵坐标为。随着阈值的减小，更多的样本值归于正类，和也相应增加，所以ROC曲线呈递增趋势。图3-6给出了一个分类模型的ROC曲线，其中45度线（图3-6中虚线形式的对角线）为参照线，如果直接随机将样本分类，即可得到该曲线。当我们使用某个模型进行预测，应该比随机猜测要好，所以ROC要尽量远离参照线。模型的ROC离参照线越远，就越往左上方靠拢，模型预测效果越好。

由于ROC是一条曲线，不方便比较不同的模型，更方便的是使用单值指标，用ROC曲线下面积（Area Under ROC Curve，AUC）来表示模型性能。模型AUC应该大于0.5（在参考线之上），值越大越好。

AUC值可直观理解为一个概率值，当我们随机挑选一个正样本以及一个负样本，当前的分类算法根据计算得到的预测值将这个正样本排在负样本前面的概率就是AUC值。AUC值越大，当前的分类算法越有可能将正样本排在负样本前面，即能够更好的分类。

ROC曲线有个很好的性质：当测试集中的正负样本的分布变化的时候，ROC曲线能够保持不变，而我们下面将要介绍的P-R 曲线的形状则会发生较大变化。在实际的数据集中经常会出现类不平衡现象，即负样本比正样本多很多（或者相反），而且测试数据中的正负样本的分布也可能随着时间变化，此时使用对样本分布不敏感的评价指标就显得十分重要。

#### 6．P-R曲线与AP

P-R曲线（Percison-Recall Curve）为不同阈值分类器的精度——召回率曲线。ROC曲线兼顾了正例和负例，但在当正负样本不均衡时（如目标检测、信息检索、推荐系统），负样本非常多（很大），因此 很小，比较和没有太大意义（ROC曲线中只有左边很小一部分有意义），因此我们只讨论正样本，即精度和召回率。

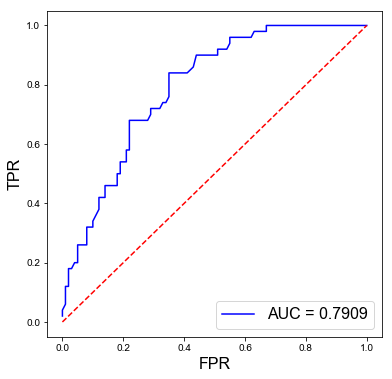
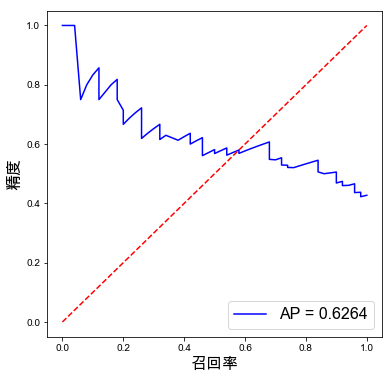
 

  图3-6 ROC曲线示例。 图3-7 P-R曲线示例。

图3-7给出了一个分类模型的P-R曲线。P-R曲线直观显示出分类器在样本总体上的查全率和查准率。因此可以通过两个分类器在同一个测试集上的P-R曲线来比较它们的预测能力：

• 如果分类器B的 P-R 曲线被分类器A的曲线完全包住，则可断言：A的性能好于B。

• 如果分类器A的 P-R 曲线与分类器B的曲线发生了交叉，则难以一般性的断言两者的优劣，只能在具体的查准率和查全率下进行比较。此时一个合理的判定依据是比较 P-R 曲线下面积：

|  |  |
| --- | --- |
|  | （3-44） |

AP也被称为平均精度（Average Percision，AP），其中分别为阈值为对应的精度和召回率。

#### 7．多类分类任务评级指标

精度、召回率和F1分数等均针对两类分类任务。对多类分类任务的评价指标，一种方式是将分类的评价拆成个两类分类的评价，然后综合多个两类分类评价指标得到多类分类任务的评价指标。综合的方式有：微观（Micro）、宏观（Macro）、加权（Weighted）。

• 微观：对每个样本（不分类别）计算全局的评价指标，每个样本的权重相同。多标签任务中首选，也可用于多类分类。

• 宏观：计算每个类别（二分类）评价指标，再求平均，每个类别的权重相同，会放大少数类（样本数目较少的类的影响）。

• 加权：计算每个类别的指标，每类的权重与该类样本数目有关，可处理不同类别样本数目不均衡问题。

• 样本：计算每个样本的评价指标，然后求平均，仅适用于多标签问题。

## **标题2** 3.6 数据不均衡分类问题

数据不均衡分类（Class-Imbalance）问题是指不同类别的训练样本数量不均衡。在实际应用中，数据不平衡问题很常见，如监测信用卡非法交易、客户流失预测、以及医学数据分类等。

对数据不均衡分类问题，首先我们的评价指标不适合用正确率，应该更注重正样本（稀有类样本）的精度和召回率，ROC曲线也是一个可选方案。

解决数据不均衡分类问题的策略可以分为两大类。一类是从训练集入手，通过改变训练集样本分布，降低不均衡程度。另一类是从学习算法入手，根据算法在解决不均衡问题时的缺陷，适当地修改算法使之适应不平衡分类问题。

### **3.6.1.** 重采样

重采样方法是上采样和下采样使不均衡的样本分布变得比较均衡，从而提高分类器对稀有类的识别率。

#### 1．上采样与样本合成

上采样（Up-Sampling）通过增加稀有类训练样本数，降低不均衡程度。最原始的是复制稀有类样本，但易导致过学习，且对提高稀有类识别率影响不大。

启发式的上采样方法有选择地复制稀有类样本，或者生成新的稀有类样本，如合成少数类过采样（Synthetic Minority Oversampling，SMOTE）。SMOTE算法的基本思想是对每个稀有类样本，从稀有类的全部个样本中找到它的个最近邻，再从中随机选择一个样本，然后生成一个0到1之间的随机数，合成一个新的稀有类样本：，新样本为样本 和表示样本的点之间所连线段上的一个点（插值）。

上采样不增加任何新的数据，只是重复或者增加人工生成的稀有类样本，这样增加了训练时间，甚至由于这些重复或是周围生成的新的稀有类样本，使分类器过分注重这些样本，导致过学习。因此上采样不能从本质上解决稀有类样本的缺失和数据表示的不充分性。

#### 2．下采样与集成学习

下采样（Down-Sampling）通过舍弃部分多数类样本，降低不均衡程度。对多数类样本进行下采样，并结合集成学习，通常能得到较好的效果。下采样法代表性的算法简单集成（EasyEnsemble）和级联集成（BalanceCascade）算法。

假设正样本数目为，而负样本数目为，我们可以通过下采样，随机无重复的生成个负样本子集，并将每个子集都与相同数目的正样本合并生成个新的训练样本集合。在个训练样本上分别训练一个分类器，最终将个分类器的结果综合起来（如求平均），这就是EasyEnsemble。

BalanceCascade算法每次从多数类中有效地选择一些样本，与少数类样本合并为新的数据集，训练一个分类器；对于那些分类正确的多数类样本不放回，对这个剩下的多数类样本再次进行下采样，产生第二个训练集，训练第二个分类器；以此类推，直到满足某个停止条件，最终的模型为多个分类器的组合。

#### 3．舍弃所有少数类，转换为异常检测问题

将稀有类样本作为异常点（Outliers），则分类问题转化为异常点检测（Anomaly Detection）或变化趋势检测问题（Change Detection）。

### **3.6.2.** 代价敏感学习

采样算法从数据层面解决数据不均衡的学习问题，在算法层面上解决数据不均衡学习的方法主要是基于代价敏感学习算法（Cost-Sensitive Learning）。

在现实任务中常会遇到这样的情况：不同类型的错误所造成的后果不同。例如在医疗诊断中，错误地把患者诊断为健康人与错误地把健康人诊断为患者，看起来都是犯了“一次错误”，但是后者的影响是增加了进一步检查的麻烦，前者的后果却可能是丧失了拯救生命的最佳时机。为了权衡不同类型错误所造成的不同损失，可为错误赋予“非均等代价”。

代价敏感学习方法的核心要素是代价矩阵，如表3-2所示。其中表示将第类样本预测为第类样本的代价。一般来说，。若将第0类判别为第1类所造成的损失更大，则。损失程度相差越大，与的值差别越大。当 与相等时，为代价不敏感的学习问题。

表3-2 代价矩阵

|  |  |  |
| --- | --- | --- |
| 真实值  预测值 | 0 | 1 |
| 0 |  |  |
| 1 |  |  |

基于代价矩阵分析，代价敏感学习方法主要有以下三种实现方式：

• 从贝叶斯风险理论出发，把代价敏感学习看成是分类结果的一种后处理。首先按照传统方法学习到一个模型，模型输出概率值，最后的分类器为：。

• 从学习模型出发，对具体学习方法的改造，使之能适应不平衡数据下的学习，如设计代价敏感的支持向量机、决策树或神经网络模型。以代价敏感的决策树为例，可从三个方面对其进行改进以适应不平衡数据的学习：决策阈值的选择、分裂标准的选择、剪枝方面。在上述三个方面均可将代价矩阵引入。

• 从预处理的角度出发，将代价用于权重的调整，使得分类器满足代价敏感的特性。如 Scikit-Learn 很多分类器有可选的 class\_weight参数，可设置不同类别样本的权重。如我们可以将不同类别样本的权重设置为与该类样本数目成反比。应该指出的是调整类的重要性通常只能影响假阴性（False Negatives），它会调整分界面以降低误差。

## **标题2** 3.7 案例分析—奥拓（**Otto）**商品分类

本节我们以Kaggle 2015年举办的奥拓商品分类竞赛数据为例，采用Logistic回归模型实现商品分类。

#### 1．数据探索和特征工程

Otto商品分类任务是一个9类商品分类问题。每个样本有93维数值型特征，特征值均为整数，表示某种事件发生的次数，特征名已经进行过脱敏处理，无法知道其物理含义。训练数据集有61878个样本，测试数据集有144368个样本。

我们探查了特征的统计量（最小值、最大值、中位数、四分之一分位数、四分之三分位数），如表3-3所示。从表中可用看出，特征的最小值到四分之三分位数几乎均为0，各特征的最大值不等。这说明特征值大部分是0（稀疏），是长尾分布（有少量特征值很大的样本）。特征的直方图也说明了这一点（图3-8（a）给出了特征的直方图）。图3-8（b）给出了商品类别的直方图，虽然每个类别的商品样本数目不同，但差异不显著，不必特别考虑。

表3-2 Otto商品分类数据集中特征统计量

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 统计量 | **feat\_1** | **feat\_2** | **feat\_3** | **feat\_4** | **feat\_5** | **feat\_6** |
| 平均值 | 0.38668 | 0.263066 | 0.901467 | 0.779081 | 0.071043 | 0.025696 |
| 标准差 | 1.52533 | 1.252073 | 2.934818 | 2.788005 | 0.438902 | 0.215333 |
| 最小值 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 0 | 0 | 0 | 0 | 0 | 0 |
| 中位数 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 0 | 0 | 0 | 0 | 0 | 0 |
| 最大值 | 61 | 51 | 64 | 70 | 19 | 10 |

|  |  |
| --- | --- |
|  |  |
| （a）特征feat\_1的直方图 | （b）商品类别直方图 |

图3-8 Otto商品分类数据集中特征的直方图。

#### 2．特征工程

本数据集中特征均为数值型特征，但特征值大部分是0（稀疏），且是长尾分布。因此特征工程可以考虑变换，减弱长尾中大特征值的影响，同时还能保持稀疏性。另外特征值表示某种事情发生的次数，类似文本分析中词频（Term Frequency, TF）特征的处理，特征工程也可考虑将出现频数进一步编码为更有判别力的（Term Frequency–Inverse Document Frequency，TF-IDF），突出对特定类别有贡献的低频词。。最后我们再对这些特征进行去量纲预处理，采用Scikit-Learn中实现，使得预处理后的仍然保持稀疏性。

需要指出的是，上述特征编码过程中用到的参数（如TF-IDF中每个单词出现的文档数目、中特征的最小值和最大值）是从训练数据中训练得到，在对测试数据编码时须用相同的参数。

#### 3．Logistic回归模型的应用

在Scikit-Learn中，Logistic回归模型为，目标函数为，其中相当于式（3-7）中的。模型超参数包括正则项和正则系数，需要结合实现超参数调优。采用原始数据做为特征，模型超参数对模型训练的影响如图3-9所示（原始特征）。图中给出了L1正则和L2正则下、不同正则参数对应的模型在训练集上测试集上的负log似然损失（）。可以看出在训练集上越大（正则越少）的模型性能越好； 但在测试集上当时性能最好（L1正则）。

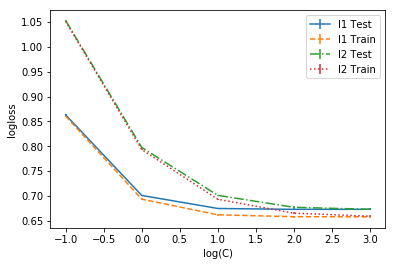


图3-8 Otto商品分类数据集上不同超参数对应模型的交叉验证测试误差估计和训练误差。

我们分别尝试了采用原始特征、变换后的特征、以及TF-IDF变换后的特征作为Logistic回归模型的输入，得到模型在Kaggle的Private Leaderboard分数为如表3-3所示。从表中可以看粗，Logistic回归在该任务上的性能并不好，可能是该任务比较复杂，Logistic回归只是一个线性模型，不足以处理这么复杂的任务。后续我们会讨论更复杂的模型在该数据集上的性能。

表3-3 不同特征编码方案的Logistic回归模型在Otto商品分类任务上的性能

|  |  |
| --- | --- |
| 特征编码方案 | 分类性能（Logloss） |
| 原始特征 | 0.66683 |
| 特征编码 | 0.67317 |
| TF-IDF特征 | **0.63319** |

## **标题2** 3.8 小结

本章从以下7个方面介绍了Logistic回归模型。

（1）模型的形式：Logistic回归模型假设事件发生的log几率用为输入特征的线性关系：，等价于、或者，其中为函数。

（2）模型的目标函数：Logistic回归模型的目标函数包含两部分：训练集上的损失函数之和、正则项。Logistic回归的损失函数采用负log似然损失（交叉熵损失），正则项可取L2正则、L1正则、或L2正则+L1正则。

（3）目标函数的优化求解：Logistic回归模型的目标函数无解析解，只能采用梯度下降法、牛顿法、坐标轴下降法等优化算法求解，可根据数据的规模和特点选择合适的优化算法。

（4）多类分类的实现：将两类分类中的贝努利分布扩展为Multinoulli分布，等于于，其中为S函数。

（5）模型性能指标：分类任务的性能指标包括正确率、log损失、混淆矩阵、以及针对两类分类任务的ROC、P-R曲线等，可根据任务要求选择合适的性能评价指标。

（6）超参数调优：Logistic回归模型的超参数调优通过验证集上的性能评价。验证集可采用交叉验证的方式得到，需要注意的是在交叉验证时要采用分层交叉验证，保证每份数据中各类别的样本数目比例相同。

（7）不同类别样本不均衡的解决方案：可从样本和分类算法两方面考虑。

## **标题2** 3.9 习题

1. 以下哪个选项是正确的？

（A）线性回归误差值必须为正态分布，但Logistic回归并非如此；

（B）Logistic回归误差值必须为正态分布，但线性回归并非如此；

（C）线性回归和Logistic回归误差值都必须为正态分布分布的

（D）线性回归和Logistic回归误差值都不是正态分布的

2. 使用梯度下降训练Logistic回归分类器后，您发现它对训练集欠拟合，在训练集或验证集上没有达到所需的性能。以下哪些项可能是有希望采取的步骤？

（A）采用其他优化算法，因为梯度下降得到的可能是局部最小值

（B）减少训练样本

（C）增加多项式特征值

（D）改用较多隐含结点的神经网络模型

3. 在Logistic回归中，关于一对其他（One vs. Rest）方法，以下哪个选项是正确的?

（A）我们需要在类分类问题中拟合个模型

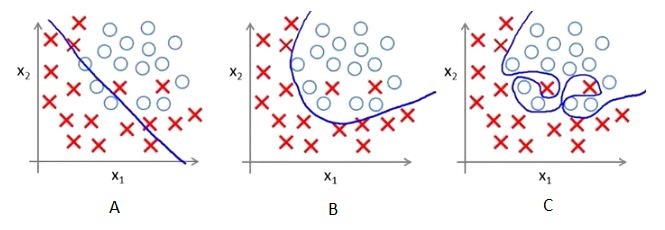
（B）我们需要拟合个模型来分类类

（C）我们只需要拟合1个模型来分类类

（D）这些都不是

4. 下面是三个散点图和手工绘制的Logistic回归决策边界。

下列图中哪个图的决策边界过拟合了训练数据？



（A）图A

（B）图B

（C）图C

（C）以上都不是

5. 根据第4题中的图，下列说法哪些是正确的？

（A）与图B和图C相比，图A中的模型的训练误差最大。

（B）最佳模型是图C中的模型，因为它训练误差最小（零）。

（C）图B中的模型比图A和图C中的模型更健壮。

（D）图C中的模型比图A和图B中的模型更为过拟合。

（E）由于我们没有看到测试数据，所以所有模型的性能相同。

6. 假设第4题图中的决策边界是由不同的正则参数生成的。上面哪个决策边界的正则最大？

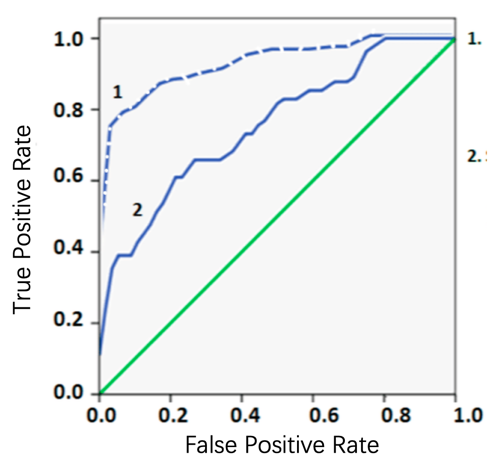
（A）图A

（B）图B

（C）图C

（D）三者的正则参数相同

7. 下图显示了2个Logistic回归模型的AUC-ROC曲线。 曲线1和曲线2哪个AUC-ROC将给出最佳结果？



8. 请推导岭回归的牛顿法。如果采用牛顿法求解，岭回归需要迭代多少次？

9. 给定个训练样本，为维向量，，我们训练一个线性分类器，损失函数为指数损失：。

（1）请给出带L2正则的分类器的目标函数。

（2）请给出第（1）题中目标函数的梯度。

（3）请给出第（1）题中目标函数的海森矩阵。

如何对正则超参数调优？

10. 采用Logistic回归模型对心脏病数据集进行建模。数据集该数据包含了270个样本，其中120个病人心脏有问题。我们根据病人的基本情况和一系列的医学检查来预测病人是否患有心脏病。数据集中各字段说明如下：

|  |  |
| --- | --- |
| 字段名 | 说明 |
| age | 年龄 |
| sex | 性别 1=male, 0=female |
| cp | 胸痛类型(4种) 1:典型心绞痛，2:非典型心绞痛，3:非心绞痛，4:无症状 |
| trestbps | 静息血压 |
| chol | 血清胆固醇 |
| fbs | 空腹血糖 >120mg/dl ,1=true; 0=false |
| restecg | 静息心电图(值0,1,2) |
| thalach | 达到的最大心率 |
| exang | 运动诱发的心绞痛(1=yes;0=no) |
| oldpeak | 相对于休息的运动引起的ST值(ST值与心电图上的位置有关) |
| slope | 运动高峰ST段的坡度 1: upsloping向上倾斜, 2: flat持平, 3: downsloping向下倾斜 |
| ca | 主血管数目 (0-3) |
| thal | 地中海贫血的血液疾病 (3 =正常;6 =固定缺陷;7 =可逆转缺陷) |
| target | 标签，是否生病0=no,1=yes |

（1）分析特征分布，对特征进行适当变换。注意数据集中既有数值型特征，也有离散型特征。

（2）用训练数据训练Logsitic回归模型，请用10折交叉验证，对模型的正则超参数和正则函数（L1正则、L2正则）进行调优，评价指标为logloss。