**第三章 逻辑回归**

**二元分类问题：是与否**

在现实中，我们经常面对一些二元选择（To be，or not to be）。比如说逛街时，是否要购买眼前的这件衣服、是否去考研究生等等。这样的二元问题在生活中大量存在，比如明天天气是晴天还是阴天、是否患有某种疾病等等。

这就是一个分类问题，线性回归模型不能很好地解决这个问题。

**逻辑回归（Logistic Regression）**

线性回归：

随机变量服从参数为的0-1分布（伯努利分布，Bernoulli Distribution），即。

几率（Odds）：，表示一个事件发生和不发生的比率，也称发生比。

对数几率（Log Odds，或Logit）：

定义一个这样的模型（对数几率为线性模型）：

由上式可得：

其中称为连接函数（Link Function），或激活函数（Activation Function）。

函数称为Sigmoid函数（S形函数）：；随着的增加，sigmoid值将逼近于1；随者的减小，sigmoid值将逼近于0。

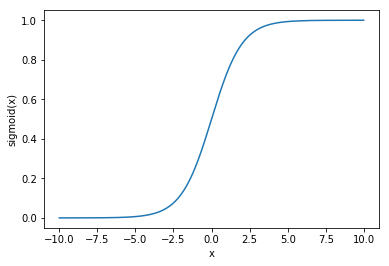


图1 sigmoid函数

的性质：

在范围0~1之间。为了实现分类，我们可以把任何大于0.5的数据归于1类，而把任何小于0.5的数据归于0类。

**极大似然估计（Maximum Likelihood Estimation）：**

极大似然估计的一般步骤：

（1） 写出似然函数；

（2） 求对数似然函数；

（3） 对对数似然函数求导；

（4） 解似然方程。

随机变量服从参数为的0-1分布，现在有。求的极大似然估计。

（1）写出似然函数：

（2）求对数似然函数：

（3）对对数似然函数求导：

（4）解似然方程：令，得。

所以，我们可以得到似然函数的最大值为

我们可以得到，和AIC

其中表示似然函数的最大值，表示模型中参数的个数。

AIC信息准则（Akaike Information Criterion），是衡量统计模型拟合优良性(Goodness of fit)的一种相对标准。由于它是日本统计学家赤池弘次创立和发展的，因此又称赤池信息量准则。它建立在熵的概念基础上，可以权衡所估计模型的复杂度和此模型拟合数据的优良性。

增加自由参数的数目提高了拟合的优良性，AIC鼓励数据拟合的优良性但是尽量避免出现过度拟合(Overfitting)的情况。所以优先考虑的模型应是AIC值最小的那一个。假设在个模型中做出选择，可一次算出个模型的AIC值，并找出最小AIC值相对应的模型作为选择对象。

贝叶斯信息规则（Bayesian Information Criterions，BIC），别名Schwartz Criterion，与AIC一样是对模型的拟合效果进行评价的一个指标，BIC值越小，则模型对数据的拟合越好。BIC的计算公式是：

其中表示似然函数的最大值，表示模型中参数的个数，表示观测样本的数目。

与AIC相比较，BIC对模型的复杂度有一个更大的惩罚项。这里模型的复杂度指的是模型中参数的个数。

AIC和BIC可以用来选择模型，即选择两个模型中的AIC或BIC小的那个模型。

**参数学习**：

逻辑回归模型：

我们可得

上式可以重新写为

我们有个独立的观测数据。

（1）写出似然函数：

（2）对取对数得到：

（3）对对数似然函数求偏导：

其中。

（4）逻辑回归的批量梯度上升（极大值）：

其中，是学习率。

线性回归的批量梯度下降（极小值）：

其中，是学习率。

定义一个损失函数

称为逻辑回归的交叉熵损失函数。

**指数族分布（Exponential Family of Distributions）**

亦称指数型分布族，是统计中最重要的参数分布族，包含了二项分布、正态分布、泊松分布等。

概率密度函数可以表达为如下形式：

如果，则指数族分布为标准形式（Standard Form），称为自然参数（Natural Parameter），如果存在之外的其它参数称为讨厌参数（Nuisance Parameter）。

泊松分布：，可以表示为指数分布

即。

正态分布：

把当作讨厌参数（Nuisance Parameter）。可以表示为指数分布

有或。

二项分布：

可以表示为指数分布

有。

**模型选择（Model Selection）**：

“向前”（Forward Selection）和“向后”（Backward Selection）：

向前是事先用一步一步的方法筛选自变量，也就是先设立门槛。称作“前”。

向后，是先把所有的自变量都进来，然后再筛选自变量。也就是先不设置门槛，等进来了再一个一个淘汰。淘汰自变量的准则可以是AIC或BIC，即选择AIC或BIC值小的模型。

**类别型变量的处理**：

类别型变量转换为哑铃变量（Dummy Variables）。比如：。

表1 类别型变量转换为个哑铃变量

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | 1 | 0 | 0 | 0 |
|  | 0 | 1 | 0 | 0 |
|  | 0 | 0 | 1 | 0 |
|  | 0 | 0 | 0 | 1 |

其中

由于

所以，一个类的类别变量，可以由个哑铃变量来表示。

pd.get\_dummies(): 将类别型变量转换为哑铃变量（Dummy Variables， 0和1）。

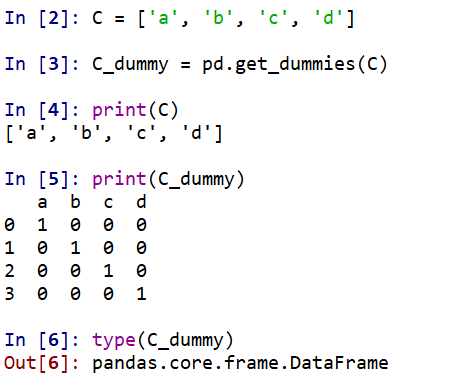


表2 类别型变量转换为个哑铃变量

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | 0 | 1 | 0 | 0 |
|  | 0 | 0 | 1 | 0 |
|  | 0 | 0 | 0 | 1 |
| 基准类别 | 1 | 0 | 0 | 0 |

默认情况下，将所有类别按字母顺序排列，排在第一位的类别为基准类别。

**将数据集分为训练集和测试集**

机器学习的其中两个步骤是：训练算法和测试算法。因此，通常把数据集分为两部分：训练集(Training Set)和测试集(Test Set)。用训练集的数据来训练算法，用测试集的数据来检测算法的效果。通常，训练集和测试集的比例为70%：30%或80%：20%。

from sklearn.model\_selection import train\_test\_split

train\_data, test\_data = train\_test\_split(data, train\_size = 0.7, random\_state = 42)

折-交叉验证（-Fold Cross Validation）：

将数据集均分成个子集，依次把其中的个子集作为训练集，剩下的 1 个子集用作测试集。在折交叉验证的过程中，每个子集均会被验证一次。平均次的结果或者使用其它结合方式，最终得到一个单一估测。

**数据预处理：**

归一化数据（Normalization）：不同特征的取值范围可能是不同的。但是，每种特征可能是同等重要的。在处理这种不同取值范围的特征值时，通常采用的方法是归一化，即将取值范围处理为0到1或-1到1之间。

有一组数据

归一化到区间：

归一化到区间：

标准化（Standardization）：

基于训练集中的数据得到用于归一化或标准化的数值，比如。在归一化或标准化测试集中的数据时，应该使用在训练集中的。

其他转换：log变换，即

一个二分类问题，可以把样本分成正类(Positive)或者负类(Negative)。但是，实际中分类时，会出现四种情况：

(1)若一个样本是正类并且被预测为正类，即为真正类(True Positive，TP)；

(2)若一个样本是正类，但是被预测成为负类，即为假负类(False Negative，FN)；

(3)若一个样本是负类，但是被预测成为正类，即为假正类(False Positive，FP)；

(4)若一个样本是负类，但是被预测成为负类，即为真负类(True Negative，TN)。

表3 一个二类问题的混淆矩阵（Confusion Matrix）

|  |  |  |  |
| --- | --- | --- | --- |
|  | | 预测结果 | |
| 1 | 0 |
| 真实结果 | 1 | 真正例（TP） | 假负例（FN） |
| 0 | 假正例（FP） | 真负例（TN） |

真阳性率（True Positive Rate， TPR）：TPR = TP / ( TP+FN )（敏感性: Sensitivity）

真阴性率（True Negative Rate，TNR）：TNR= TN / (FP + TN) （特异性：Specificity）

假阳性率（False Positive Rate，FPR）：FPR = FP / ( FP + TN )

假阴性率 (False Negative Rate，FNR）：FNR = FN /（TP + FN）

准确率（Accuracy）：Acc = ( TP + TN ) /(TP+TN+FP+FN)

精确度 (Precision)：TP / ( TP+FP )

召回率（Recall）：TP / (TP + FN )

ROC曲线：接收者操作特征(Receiver Operating Characteristic)。ROC曲线给出的是当阈值变化时假阳性率和真阳性率的变化情况。

横轴：假阳性率(False Positive Rate, FPR)特异度；1-Specificity，1-真阴性率；

纵轴：真阳性率(True Positive Rate, TPR)灵敏度，Sensitivity。

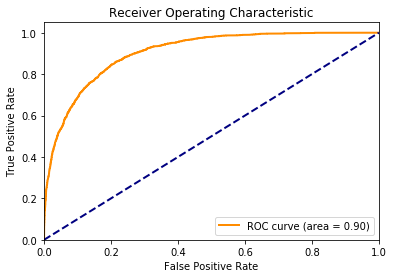


图2 ROC曲线

在理想情况下，最佳的分类器应该尽可能位于左上角，也就是分类器在假阳性率很低的同时获得了很高的真阳性率。

对不同的ROC曲线进行比较的一个指标是曲线下的面积（Area under the Curve，AUC）。AUC给出的是分类器的平均性能。一个完美分类器的AUC为1，而随机猜测的AUC则为0.5。

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import roc\_curve, auc ###计算roc和auc

confusion\_matrix(y\_true, y\_predict) ### y\_predict 预测的分类结果，0或1

tn, fp, fn, tp = confusion\_matrix(y\_true, y\_predict).ravel()

fpr, tpr, thresholds = roc\_curve(y\_true, y\_predicted\_values) #y\_predicted\_values预测的概率[0， 1]

roc\_auc = auc(fpr, tpr)

**模型结果的解释**：

我们得到逻辑回归模型，但应该如何理解这得到的模型呢？比如，模型中的参数表示什么？当一个变量变动时，它将如何影响最终的结果呢？

我们假设有一个简单的模型：

**模型参数的意义:**

令变量保持不变，从变为

也就是。这表示，在其他变量不变的情况下，当变量增加1时，对应的几率（odds）变为之前的倍。

**边际效应（Marginal Effects）：变量变动对概率的影响**

大多数情况下，我们更关心变量的变动对最终结果的影响。对逻辑回归来说，也就是变量的变动对事件发生概率的影响：当某个变量增加1时，事件发生的概率将如何变化？这被称为变量的边际效应。从数学上讲，某个变量的边际效应等于被预测量对这个变量的导数。

对两边分别对求导得

上式表明当变量增加1时，事件发生的概率的变化为。由于表示事件发生的概率，对于不同的数据点，参数是不变的，但是概率可能是不一样的。这样，对于某一个变量，不同数据点上的边际效应是不一样的。

在实际中，针对某一变量，我们首先计算样本集中所有数据点的边际效应，然后计算这些边际效应的平均值。这个平均值就被认为是这个变量的边际效应。