## 0、概述

模型性能评估（或者说模型性能的度量方式，或者说模型性能评价的准则）是机器学习中很重要的一个问题。当我们想衡量一个模型的效果好坏时，我们需要使用性能度量（比如比较两个模型对于一个任务，哪个更好一些）。另外，在构建模型的目标函数时，我们同样需要性能度量（将模型的具体形式带入到评价准则中，就得到了目标函数）。

## 一、回归模型性能度量准则

最常用的回归模型度量准则是最小均方误差（MSE）。

给定训练集，训练集中有m个样本，每个样本有一个实值的标签；

并给定模型，其中是模型的未知参数，

则最小均方误差为：

|  |  |
| --- | --- |
|  | (1) |

可以看出，MSE的思想是，认为最佳的模型能够使其对训练集中每个样本的输出与样本标签之间的误差的平方的均值最小。这个思想是比较合理的，因为我们最基本的希望的就是模型对一个样本的输出能够尽量接近该样本的标签。

很多模型的目标函数都是基于MSE准则建立的。但是仅仅使用MSE来构建目标函数可能会造成模型的过拟合，这时可以用正则化的方法，从目标函数构建的过程出发来减轻模型的过拟合。

## 二、分类模型性能度量准则

分类模型的性能度量准则有很多，包括准确率/错误率、查准率、查全率、PR曲线与平衡点、 Score/ Score、ROC曲线与AUC、交叉熵等。这些准则中有的只能用于二类分类（但可以扩展成用于多类分类的情况），有的适用于任意多类分类。下面我们分别介绍这些准则：

### 1、准确率与错误率

准确率：分类正确的样本数占总样本数的比例；

错误率：分类错误的样本数占总样本数的比例。

### 2、查准率与查全率

查准率又叫精确率，查全率又叫召回率。这两个指标只适用于二类分类模型。为了解释这两个指标，我们先介绍“混淆矩阵”的概念。

表1 混淆矩阵

|  |  |  |
| --- | --- | --- |
| **真实情况** | **预测结果** | |
| **正例** | **反例** |
| **正例** | TP（真正例） | FN（假反例） |
| **反例** | FP（假正例） | TN（真反例） |

如上表所示为混淆矩阵，它不仅记录了模型预测结果有多少是正确、有多少是错误的，它进一步记录了预测正确（或预测错误）的结果中，有多少是正例、有多少是反例。或者说，它记录了在真实的正例（或反例）中，模型预测正确了多少、预测错了多少。

显然，TP+FP+FN+TN=样本总数。

基于混淆矩阵，我们可以定义查准率和查全率：

查准率：，表示在所有查出来的正例中，有多少是真正例；

查全率：，表示在所有真实正例中，模型查出来多少正例。

查准率和查全率的意义是什么呢？

当分类模型的准确率很高时，模型必定同时拥有高查准率和高查全率。但当模型的准确率不那么高时，模型可能查准率较高，但查全率不高；也可能查全率较高，而查准率不高。对于实际问题，有时候我们更加关心查全率，有时候更关心查准率。比如在疾病检测问题中，我们更关心查全率。因为我们希望所有患病的人（正例）都尽量能被查出来，宁可错查千人，不想让一人漏网。也就是对于所有真实的病患，希望尽可能多地查全，即希望查全率尽量高。

在实际中（而并非理论上），当模型的性能一定时，查准率和查全率是一对矛盾的度量指标。在其他条件不变的情况下（比如，样本数量不变），追求其中一个指标的提高就会造成另一个指标降低。比如，要想追求高查准率，也就是说在所有查出来的正例中，真正例的比例尽可能高，则需要选择最有把握的正例作为预测的正例，这样难免漏掉一些真正例，造成查全率低；而要想追求高查全率，也就是说希望模型尽可能把所有真正例都查出来（识别为正例），则模型需要增加选择的正例的数量，此时必然会将一些不是那么有把握的正例也选了出来，造成查准率低。

基于查准率和查全率，衍生出很多分类性能的度量指标（如P-R曲线平衡点、F1、AUC等），它们都相当于查准率和查全率的某种意义上的综合。

### 3、P-R曲线

P-R曲线是基于查准率和查全率的二类分类模型的性能度量指标。

对于一个分类模型，若该模型属于概率模型，则它能对每一个样本给出其判为每一类别的“可能性”（即概率）。这时，我们可以将训练集按照该模型判为正例的可能性排序。即从左到右为从最可能是正例的样本到最不可能是正例的样本（显然，排序的质量反映了模型的好坏）。然后在排序的样本序列上设置分割点，并假设分割点左侧都是正例，分割点右侧都是反例，再让分割点从左到右移动，对于每个分割点，都有一个查准率和查全率，这就在P-R坐标系中形成一个点，将其标在坐标系中，于是分割点从左到右移动就形成了一条曲线，称为P-R曲线。

（实际上，每个分割点都对应于一个模型）

“排序的质量反映了模型的好坏”这句话怎么理解呢？我们认为，存在一个“完美的模型”，这个完美的模型使用完美的准则，对每个样本被判为正例的可能性给出了恰如其分的评估，并按可能性对样本进行排序。在这个排序上的某个位置设置分割点并将左边的样本都判为正例，右边的样本都判为反例，其判断结果与所有样本的真实分类完全一致（啊，我们多么希望获得这个完美模型）。以这个完美模型作出的排序为参考，我们的模型作出的排序与之越一致，则认为排序质量越好，模型也就越好；反之认为排序质量越差，模型也越差。

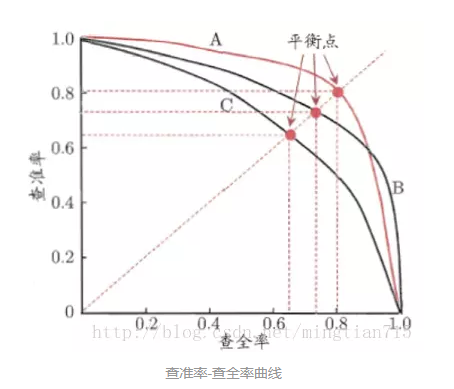


图1 P-R曲线示意图

从图中可以看出，模型的分割点越往左，越倾向于选择有较大把握的正例，则查准率越高；分割点越往右，越倾向于选择较多的正例，自然查全率越高。这就体现了我们上面所说的查准率和查全率是一对矛盾。假如我们更关心查准率，那么我们设置分割点就偏左一点；若更关心查全率，则设置分割点就偏右一点。

我们看到，图中的P-R曲线中在最右端处查准率趋近于0，这并不是因为TP趋近于0（事实上，分割点在靠近右端的时候，TP基本不会发生改变了），而是因为FP趋近于无穷大了，因为右边的样本基本上都是反例，而我们都把它们当作是正例了。当然，只有在正反两类样本极端不平衡（反例样本数趋近于无穷大）时才会出现FP趋近于无穷大，如果样本类别平衡，那么右端处查准率应该趋近于0.5。（也就是说，该示意图针对的是样本集中反例样本数趋近于无穷大的情况，这样做是为了使问题更一般化）

在P-R曲线上，查准率=查全率的点称为“平衡点”，显然，平衡点与原点距离越远，说明查准率和查全率越大，则模型性能越好。因此我们通常将平衡点作为查准率和查全率的一个综合指标。

### 4、 Score/ Score

可以看出，相当于是查准率和查全率的调和平均，相当于是查准率和查全率的加权调和平均（是权重）。

### 5、ROC曲线与AUC

ROC曲线与P-R曲线类似，也是在根据模型对训练集样本的判为正例的可能性进行排序后的序列上设置分割点并从左到右移动，但它针对每个分割点计算的不是查准率和查全率，而是假正例率和真正例率。

假正例率 ，意味着在所有真实的反例中，有多少被模型误判为正例；

真正例率 ，也就是查全率，意味着在所有真实的正例中，模型查出了多少正例。

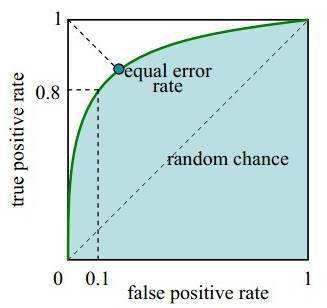


图2 ROC曲线示意图

注意，图中显示的ROC曲线是样本类别平衡，即正反例数目近似相等的情况。

下面我们解释一下ROC曲线的变化规律：

显然，分割点在最右侧时，所有样本都被判为正例，所以真正例率和假正例率都为1；对于一个正常的分类模型，其对样本的排序在左侧大部分为真实正例，在右侧大部分为真实的反例，所以分割点在左半部分移动时，查出的真正例增加很快，即真正例率增加很快，而到了右半部分，该查出的正例基本都已经查完了，所以真正例率增速放缓；同理可知，分割点在左半部分移动时，被误判为正例的样本很少，所以假正例率增加缓慢，而到了右半部分，误判为正例的样本迅速增多，即假正例迅速增多，导致假正例率迅速增加。

基于ROC曲线，我们提出了AUC指标，AUC就是ROC曲线下的面积。AUC越大，模型性能越好。下面我们直观地解释一下这个结论。

我们考虑两个极限的情况：一是折线(0,0)-(0,1)-(1,0)，也就是沿着y轴向上，再沿着y=1向右形成的曲线——显然它包含的面积最大，那么它对应于什么模型呢？根据ROC坐标的意义可知，这就是完美的模型，即使用(0,1)对应的分割点，该模型的分类结果跟实际完全吻合：左边的样本点都是真实的正例，右边的样本点都是真实的反例；二是对角线——对角线表示的是随机分类器，因为它意味着不管分割点在哪，分类的结果中，模型查出的正例中一半是真正例，一半是真反例（注意，这个结论在训练样本中真正例和真反例的个数相等的情况下（即样本类别平衡）才成立）。

上面我们介绍了PR曲线和ROC曲线，那么二者有什么区别呢？

显然，两条曲线的意义是不同的，你只要深入理解了它们的意义，自然就知道二者的区别。下面介绍一下实际应用中如何选择。

（1）从横纵坐标的意义上看，ROC曲线兼顾正例与负例，而PR曲线则聚焦于正例。因此，如果只关心模型对正例的预测性能，则选PR曲线；

（2）ROC曲线对不同类别样本数的差异的变化不敏感，而PR曲线对此敏感。比如，正例不变，负例增加了10倍，则ROC曲线不会有明显变化，而PR曲线则变化较大。因此，若认为数据不平衡会导致不同性能的模型，那么你需要的是PR而不是ROC，如果你需要度量模型不受数据不平衡影响的那种意义的性能，你应该选择ROC而不是PR；

（3）对于数据不平衡的情况，ROC通常会有一个过分乐观的性能估计，因此应当使用PR。（你可能会说，ROC和PR本身就是性能度量的方式，那么说它评价“乐观”，肯定是跟一个标准的性能度量方式进行比较得出的结论。是的，但是这里的“标准的性能度量方式”，其实是我们普遍承认的一种经验。）

注意，PR曲线和ROC曲线都需要模型给出每个样本被判为正例的“可能性”，因此对概率模型是可以定义PR曲线和ROC曲线的。但是对于非概率模型，是否就不能使用它们来衡量模型的性能了呢？实际上，对于非概率模型，虽然并不是全都可以输出“可能性”，但目前主流的算法都支持输出“可能性”。比如决策树（某个样本判为正例/负例的可能性可以被定义为该样本所在叶子节点中正例/负例的比例）、SVM（通过距离标准化的方式给出“可能性”）。

### 6、交叉熵

首先，我们介绍一下熵、交叉熵、相对熵三个概念。

（1）熵

机器学习中我们谈论的熵都是指“信息熵”。严格的信息熵的概念是针对随机变量而言的，设离散型随机变量X，其取值空间为，则

特别注意，这里用来表示随机变量X的取值空间中的第i个值（它不会重复出现），而不是表示样本x的第i维特征（它会重复出现）。

我们也可以将针对随机变量的信息熵扩展成针对包含许多样本的集合的信息熵的概念。设集合D包含很多样本，这些样本的类别是1,2,…,K，每类样本的个数记为，则：

我们可以看出，对于随机变量而言，它的熵表示了它的不确定程度，熵越大，其不确定程度就越大。对于取值空间相同的随机变量，最大的熵对应于均匀分布；最小的熵对应于随机变量确定地取其中一个值（取该值的概率为1，取其他值的概率为0）。

实际上，对于随机变量而言，熵的本质的意义是消除随机变量的不确定性所需的最小代价。显然，代价越大，表示随机变量的不确定程度越大。

这里需要解释一下“最小”代价是什么意思：对于一个随机变量，我们可以用不同策略去消除它的不确定性，每一种策略对应于一种分布，相当于用不同的分布去消除该随机变量的不确定性，不同的策略（分布）产生的代价是不同的，当使用该随机变量本身的分布去消除不确定性时，代价最小。

更具体地，“消除不确定性”指的是对随机变量进行“编码”，而其代价则是编码长度的期望。

而对于集合而言，它的熵表示了它的“纯度”，熵越大，则集合越不纯，也就是它里面的样本的类别数越多且各类样本数目越接近；熵越小，则集合越纯。

另外值得一提的是，衡量一个随机变量的不确定程度的指标并不是只有信息熵，比如随机变量的方差也能在一定程度上衡量随机变量的不确定程度，方差越大则不确定程度越大。

（2）交叉熵

设在一个取值空间上有两个随机变量，它们的分布分别是p和q，则它们的交叉熵是：

交叉熵衡量了p和q的相似性，p与q越一致，则它们的交叉熵越小。

也可以将交叉熵理解为用分布p去消除具有分布q的随机变量的不确定性的代价。p与q越接近，则代价越小。

交叉熵是不对称的，。

（3）相对熵

相对熵又叫作KL散度，是衡量两个分布p和q的“距离”的指标，其定义如下：

相对熵越大，则p和q“距离”越大，即p和q越不相似；相对熵越小，则p和q距离越小，即p和q越相似。

相对熵与交叉熵有密切的关系：

注意，相对熵对p和q也是不对称的，即。

（4）交叉熵用作分类问题的概率模型的性能度量：

给定分类问题的训练集，训练集中有m个样本，每个样本有一个类别标签。给定一个具体的概率模型，它能计算出样本集上的条件分布（“预测分布”）；另外，我们从训练集上能统计出基于训练集的条件分布（“真实分布”或者说“经验分布”），然后计算二者的交叉熵，这相当于用预测分布去消除具有真实分布的训练集的不确定性，因此二者的交叉熵可以作为模型性能度量指标，它越小，说明预测分布越接近真实分布，模型的预测性能越好；反之模型的预测性能越差。

对于二类分类问题的概率模型，其交叉熵目标函数如下：

其中取值为0或1，是模型将样本判为正例的概率：（对于具体的模型而言，函数有具体的形式）。

为什么是这样呢？下面我们具体说明一下：

我们可以对样本集按照的取值分组，每组中的取值相同。我们取其中一组为例，假设该组的值都为，该组共有样本个，其中正样本个，负样本个。并且因为是二类分类问题，只取0和1，则该组上的交叉熵就是

上式也就等于

对于其他组，也是如此。然后把它们加权相加（权重等于，因为个数越多的组肯定权重越大，或者这样理解：因为是对每个样本计损失，所以累加时，每组前面要乘），这样，就约掉了每一组前面的项，就得到了最终的形式。

另外，我们可以证明，根据交叉熵导出的目标函数与根据极大（对数）似然估计导出的目标函数是一样一样的。这也并不奇怪，两者本质上都是希望将模型的预测结果跟观测的数据尽量地匹配。

## 三、聚类模型性能度量准则

聚类性能度量，或称聚类有效性指标，包括外部指标和内部指标。外部指标是指将聚类结果与某个参考模型比较，来衡量聚类的有效性；内部指标是指不利用参考模型，而衡量聚类的有效性。外部指标主要考虑聚类结果与参考模型的一致程度；内部指标主要基于样本间距离，考察“簇内样本距离是否尽量近”和“簇间样本距离是否尽量远”。

### 1、外部指标

给定数据集，设聚类算法将数据分为K个簇，参考模型则分为S个簇（K与S不一定相等）。

首先定义四个量：

其中是待评价的聚类算法将第i个样本归入的簇的标记，是参考模型将第i个样本归入的簇的标记。

可见，a表示被聚类算法分到同一个簇，且参考模型中确在同簇的样本对数；b表示被聚类算法分到同簇，但参考模型中不在同簇的样本对数；c表示被聚类算法分到不同簇，但参考模型中在同簇的样本对数；d表示被聚类算法分到不同簇，且参考模型中不在同簇的样本对数。显然，。

基于a、b、c、d，可以定义以下常用的外部指标：

这几个指标的结果值都在[0,1]区间，且值越大，聚类效果越好。

### 2、内部指标

内部指标通过量化簇内相似度和簇间相似度来衡量聚类的效果。

首先定义四个指标：

其中第一个指标表示簇C内样本间的平均距离；第二个指标表示簇C内样本间最远距离；第三个指标表示簇和最近的样本间的距离；第四个指标表示簇和中心点间的距离。

这四个指标从不同角度简单地衡量了聚类结果中的每个簇的簇内相似度或各簇之间的簇间相似度。将它们综合，即得到常用的内部指标：

DBI的值越小，聚类效果越好；DI则相反，值越大，聚类效果越好。