2.1 经验误差与过拟合

2.1.1 一些基本概念

1 错误率与精度

- **错误率**: 把**分类错误的样本数**占**样本总数**的比例称为"**错误率**" (error rate), 即如果在 m 个样本中有 a 个样本分类错误, 则**错误率** $E = a/m$
- **精度:** 相应的, $1 - a/m$ 称为**精度** (accuracy), 即 "精度 = 1 - 错误率".

2 误差

- **误差:** 学习器的**实际预测输出**与**样本的真实输出**之间的差异称为"误差" (error)
- **训练误差或经验误差:** 学习器在**训练集上的误差**称为"**训练误差**" (training error) 或"**经验误差**" (empirical error)
- **泛化误差:** 学习器在**新样本上的误差**称为"**泛化误差**" (generalization error)

我们**希望得到泛化误差小的学习器**. 然而, 我们事先并不知道新样本是什么样, 实际能做的是**努力使经验误差最小化**.

2.1.2 过拟合

1 什么是过拟合

希望是能够学得一个, 在新样本上能表现得很好的学习器.

该从训练样本中尽可能学出适用于所有潜在样本的"**普遍规律**"，这样才能在遇到新样本时做出正确的判别. 然而，当**学习器把训练样本学得"太好"**了的时候，很可能已经把**训练样本自身**的一些**特点**当作了所有**潜在样本都会具有的一般性质**，这样就会导致**泛化性能下降**.

这种现象在机器学习中称为"**过拟合**" (overfitting). 与"过拟合"相对的是"**欠拟合**" (underfitting) , 这是指**对训练样本的一般性质尚未学好**.

图 2.1 给出了关于过拟合与欠拟含的一个便于直观理解的类比.

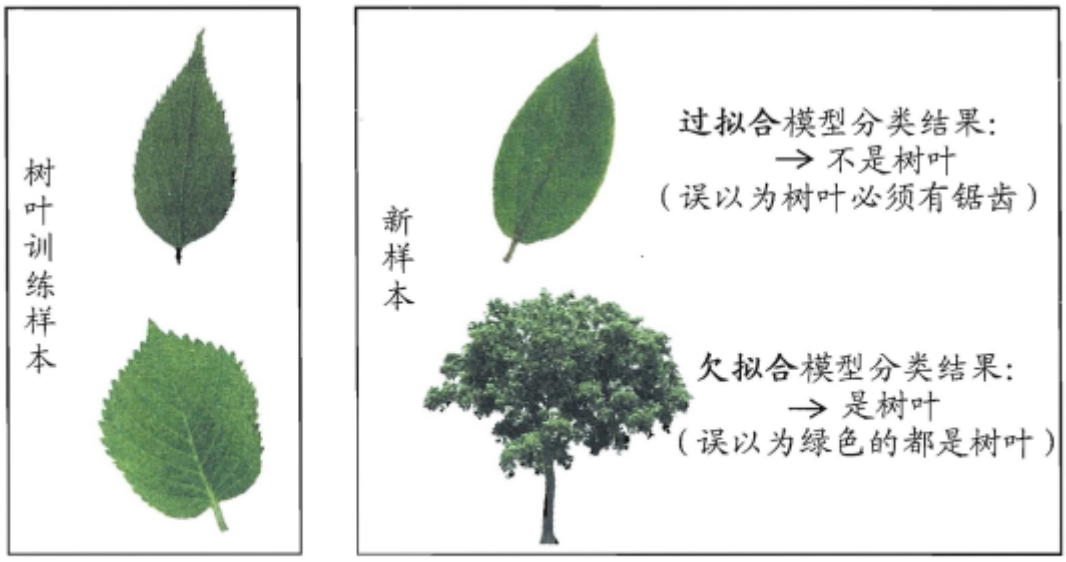


图 2.1 过拟合、欠拟合的直观类比

2 如何克服过拟合

造成过拟合的因素有很多, 其中最常见的情况是**由于学习能力过于强大**, 以至于把训练样本所包含的不太一般的特性都学到了, 而欠拟合则通常是由于**学习能力低下**而造成的.

拟合比较容易克服, 而过拟合则很麻烦. 过拟合是机器学习面临的关键障碍, 各类学习算法都必然带有一些针对过拟合的措施; 然而必须认识到, **过拟合是无法彻底避免的**, 我们所能做的只是"**缓解**", 或者说**减小其风险**.

2.2 评估方法

2.2.0 测试集

用一个"**测试集**" (testing set) 来测试**学习器对新样本的判别能力**，然后以测试集上的"**测试误差**" (testing error) 作为**泛化误差的近似**.

- 通常假设测试样本也是从样本真实分布中**独立同分布**采样而得
- **测试集**应该尽可能与**训练集** *互斥*, 即测试样本尽量不在训练集中出现、未在训练过程中使用过.

2.2.1 留出法

1 留出法的基本概念

"**留出法**" (hold-out) 直接将数据集 D 划分为**两个互斥的集合**, 其中一个集合作为**训练集** S, 另一个作为**测试集** T, 即 $D = S \cup T, S \cap T = \varnothing$.在 S 上训练出模型后，用 T 来评估其**测试误差**, 作为对**泛化误差的估计**.

2 留出法需要注意的问题

- 1 训练/测试集的划分要尽可能**保持数据分布的一致性**, 避免因数据划分过程引入额外的偏差而对最终结果产生影响.

  如分类任务中至少保持样本的类别比例相似, 从采样角度来看, 则保留类别比例的采样方式通常称为"分层采样".

- 即使在给定训练/测试集的样本比例后, 仍**存在多种划分方式对**初始数据集 D 进行**分割**.

  这些不同的划分将导致不同的训练/测试集, 相应的, 模型评估的结果也会有差别.因此, 单次使用留出法得到的估计结果往往不够稳定可靠, 在使用留出法时, 一般要**采用若干次随机划分**、**重复进行实验评估后**取**平均值**作为留出法的评估结果.

- 一般情况下, 将大约 $2/3 \sim 4/5$ 的样本用于训练, 剩余样本用于测试.

2.2.2 交叉验证法

1 交叉验证法的基本概念

- 1 "交叉验证法" (cross validation) 先将数据集 D **划分为** k **个大小相似的互斥子集**. 即 $D = D_1 \cup D_2 \cup \ldots \cup D_k, D_i \cap D_j = \varnothing (i \neq j)$. **每个子集** D_i **都尽可能保持数据分布的一致性**, 即从 D 中通过**分层采样**得到.
- 2 然后, 每次**用** $k-1$ **个子集的并集**作为**训练集**, **余下的那个子集**作为**测试集**
- 3 这样就可获得 k **组训练/测试集**, 从而可**进行** k **次训练和测试**, 最终返回的是这 k **个测试结果的均值**.

交叉验证法评估结果的**稳定性和保真性**在很大程度上**取决于** k **的取值**, 为强调这一点, 通常把交叉验证法称为 " k**折交叉验证**" (k-fold crossvalidation). k 最常用的取值是 10，此时称为 10 折交叉验证.

10 折交叉验证示意图:

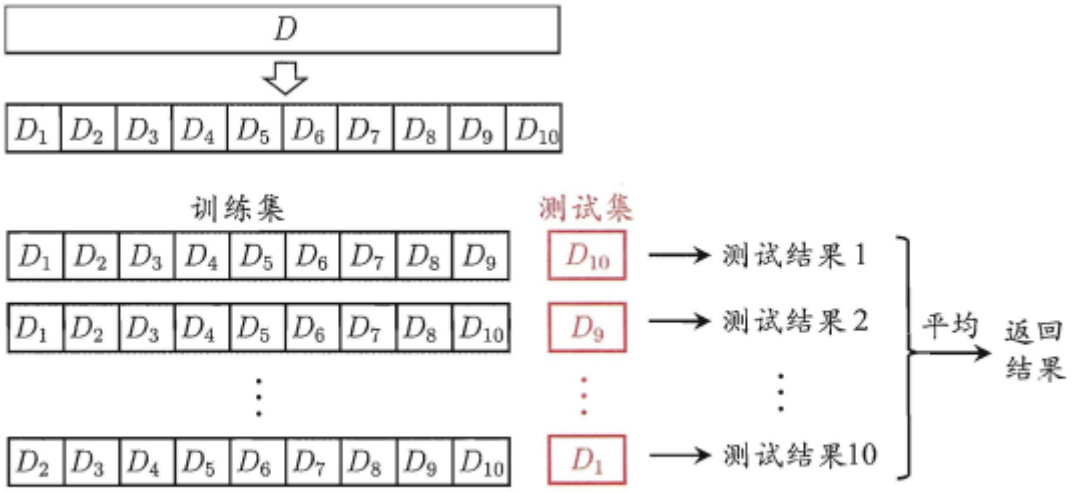


图 2.2　10 折交叉验证示意图

2 交叉验证法的划分方式

与留出法相似, 将数据集 D 划分为 k 个子集同样存在多种划分方式. 为减小因样本划分不同而引入的差别, k 折交叉验证通常要随机使用不同的划分**重复** p **次**, 最终的评估结果是这 p 次 k 折交叉验证结果的**均值**.

3 留一法

假定数据集 D 中包含 m 个样本, 若令 $k=m$, 则得到了**交叉验证法的一个特例**: 留一法 (Leave-One-One, 简称 LOO). 显然, **留一法不受随机样本划分方式的影响**，因为 m 个样本只有唯一的方式划分为 m 个子集——每个子集包含一个样本; 留一法使用的训练集与初始数据集相比只少了一个样本, 这就使得在绝大多数情况下, 留一法中被实际评估的模型与期望评估的用 D 训练出的模型**很相似**. 因此, **留一法**的评估结果往往被认为**比较准确**. 然而, 留一法也有其缺陷: 在数据集比较大时, 训练 m 个模型的计算**开销**可能是难以忍受的.

2.2.3 自助法

1 自助法的需求背景

在留出法和交叉验证法中, 由于保留了一部分样本用于测试, 因此实际评估的模型所使用的训练集比 D 小，这必然会引入一些因**训练样本规模不同而导致的估计偏差**. 留一法受训练样本规模变化的影响较小，但计算**复杂度又太高**了

2 自助法解决方案

自助法以自助采样法为基础: 给定包含 m 个样本的数据集 D, 我们对它进行采样产生数据集 D' : **每次随机从 D 中挑选一个样本, 将其拷贝放入 D' , 然后再将该样本放回初始数据集 D 中,** 使得该样本在下次采样时仍有可能被采到; **这个过程重复执行 m 次后, 我们就得到了包含 m 个样本的数据集 D'** , 这就是自助采样的结果. 显然, D 中有一部分样本会在 D' 中多次出现, 而另一部分样本不出现.

可以做一个简单的估计, 样本在 m 次采样中始终不被采到的概率是 $\left(1-\frac{1}{m}\right)^m$, 取极限得到

$$\lim_{m \mapsto \infty}\left(1-\frac{1}{m}\right)^m \mapsto \frac{1}{e} \approx 0.368 \tag{2.1}$$

即通过自助来样, 初始数据集 D 中约有 36.8% 的样本未出现在采样数据集 D' 中.

于是我们可将 D' **用作训练集**, $D \backslash D'$ **用作测试集**; 这样, 实际评估的模型与期望评估的模型都使用 m 个训练样本, 而我们仍有数据总量约 $1/3$ 的、没在训练集中出现的样本用于测试. 这样的测试结果, 亦称"包外估计".

3 自助法的适用范围

- 自助法在**数据集较小**、**难以有效划分训练/测试集**时**很有用**;
- 此外, 自助法能从初始数据集中产生多个不同的训练集, 这对**集成学习等方法**有很大的**好处**.
- 然而, 自助法产生的数据集改变了初始数据集的分布, 这会引入估计偏差.因此, 在**初始数据量足够**时, **留出法**和**交叉验证法**更**常用**一些.

2.2.4 调参与最终模型

1 参数调节

大多数学习算法都有些参数 (parameter) 需要设定, **参数配置不同**, 学得模型的性能往往有**显著差别**. 因此, 在进行模型评估与选择时, 除了要对适用学习**算法**进行选择, 还需对**算法参数**进行设定, 这就是通常所说的"**参数调节**"或简称"**调参**" (parameter tuning).

2 调参和算法选择的区别与联系

- **相同点**: 调参和算法选择都是**对每种参数配置都训练出模型, 然后把对应最好模型的参数作为结果**

- **区别点**: 学习算法的很多参数是在**实数范围内取值**, 因此, 对每种参数配置都训练出模型来是**不可行**的. 现实中常用的做法, 是对每个参数选定一个**范围和变化步长**. 这种策略是在计算开销和性能估计之间进行这种的结果, 从而使得学习过程变得可行.

3 验证集、训练集和测试集

通常把学得模型在实际使用中遇到的数据称为**测试数据**, 为了加以区分, **模型评估与选择中用于评估测试的数据集**常称为"**验证集**" (validation set). 在研究对比不同算法的泛化性能时, 我们用**测试集**上的判别效果来**估计**模型在实际使用时的**泛化能力**, 而把**训练数据**另外划分为**训练集**和**验证集**, 基于**验证集**上的性能来进行**模型选择**和**调参**.

更为通俗的理解, **训练集用来训练模型, 验证集用来模型选择和调参,测试集用来评估模型泛化性能**

2.3 性能度量

对学习器泛化性能的评估, 在有了可行的实验估计方法后, 还需要有**衡量模型泛化能力的评价标准**, 这就是**性能度量**.

在对比不同模型的能力时, 使用**不同的性能度量往往会导致不同的评判结果**; 这意味着**模型的"好坏"是相对的**, 什么样的模型是好, 不仅取决于**算法**和**数据**, 还决定于**任务需求**.

给定样例集 $D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\ldots,(\boldsymbol{x}_m,y_m)\}$, 其中 y_i 是示例 $\boldsymbol{x}_i$ 的真实标记. 要评估学习器 f 的性能, 就要把**学习器预测结果** $f(\boldsymbol{x})$ 与**真实标记** y 进行**比较**.

回归任务中常用的性能度量是"**均方误差**" (mean squared error)

$$E(f;D)=\frac{1}{m}\sum_{i=1}^{m}\left(f\left(\boldsymbol{x}_i\right)-y_i\right)^2 \tag{2.2}$$

更一般的, 对于数据分布 $\mathcal{D}$ 和概率密度函数 $p(\cdot)$, 均方误差可描述为

$$E(f;\mathcal{D})=\int_{\boldsymbol{x}\sim\mathcal{D}}(f(\boldsymbol{x})-y)^2p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{2.3}$$

2.3.1 错误率与精度

错误率是**分类错误的样本数**占样本**总数**的比例, **精度**则是分类正确的样本数占样本总数的比例.

对于样例集 D , 分类错误率定义为

$$E(f;D)=\frac{1}{m}\sum_{i=1}^{m}\mathbb{I}\left(f\left(\boldsymbol{x}_i\right)\neq y_i\right) \tag{2.4}$$

精度则为

$$\begin{aligned}\mathrm{acc}(f;D)&=\frac{1}{m}\sum_{i=1}^{m}\mathbb{I}\left(f\left(\boldsymbol{x}_i\right)=y_i\right)\\&=1-E(f;D)\end{aligned} \tag{2.5}$$

更为一般的, 对于数据分布 $\mathcal{D}$ 和概率密度函数 $p(\cdot)$, 错误率与精度可以定义为

$$E(f;\mathcal{D}) = \int_{\boldsymbol{x}\sim\mathcal{D}} \mathbb{I}(f(\boldsymbol{x}) \neq y)p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{2.6}$$

$$\begin{aligned}\mathrm{acc}(f;\mathcal{D}) &= \int_{\boldsymbol{x}\sim\mathcal{D}} \mathbb{I}(f(\boldsymbol{x}) = y)p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \\ &= 1 - E(f;\mathcal{D})\end{aligned} \tag{2.7}$$

2.3.2 查准率、查全率与 $F1$

错误率衡量了有**多少比例的瓜被判别错误**，但"**挑出的西瓜中有多少比例是好瓜**"—**查准率** 或者"**所有好瓜中有多少比例被挑了出来**"—**查全率**, 这两个指标同样非常重要.

1 查准率和查全率的定义

对于二分类问题, 可将样例根据其**真实类别**与**学习器预测类别**的组合划分为**真正例** (**true positive**)、**假正例** (**false positive**)、**真反例** (**true negative**) 、**假反例** (**false negative**) 四种情形，令 TP、 FP、 TN、 FN 分别表示其对应的样例数, 则显然有 $TP + FP + TN + FN =$ 样例总数. 分类结果的"**混淆矩阵**" (confusion matrix) 如表 2.1 所示:

表 2.1　分类结果混淆矩阵

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)

查准率 P 和**查全率** R 分别定义为

$$P = \frac{TP}{TP + FP} \tag{2.8}$$

$$R = \frac{TP}{TP + FN} \tag{2.9}$$

查准率和查全率互为矛盾体:

查准率和查全率是一对矛盾的度量. 一般来说, **查准率高时, 查全率往往偏低**; 而**查全率高时, 查准率往往偏低**.

2 $P - R$ 曲线

根据**学习器的预测结果对样例进行排序**, 排在**前面**的是学习器认为**"最可能 "是正例的样本**, 排在最后的则是学习器认为**"最不可能"是正例的样本**.

按**此顺序逐个把样本作为正例进行预测**, 则每次可以计算出**当前的查全率、查准率**. 以查准率为纵轴、查全率为横轴作图, 就得到了**查准率-查全率曲线**, 简称"$P - R$ **曲线**". 如下图

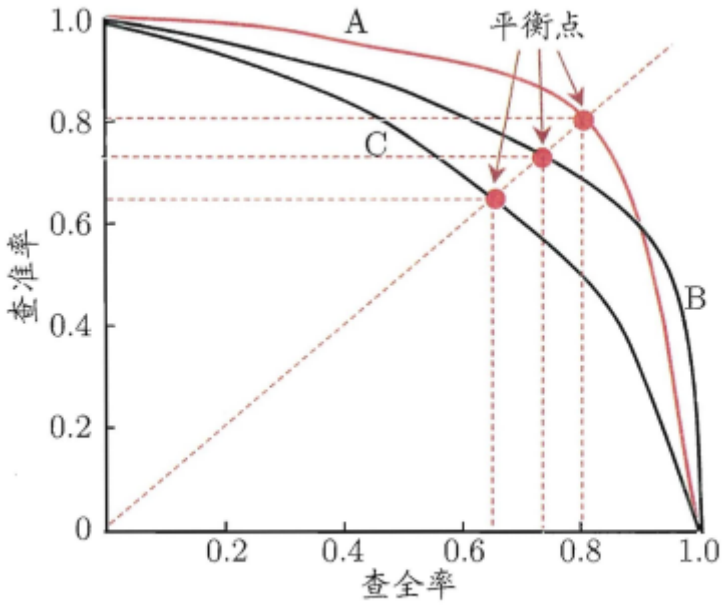


图 2.3 P-R曲线与平衡点示意图

$P-R$ 图直观地显示出学习器在样本总体上的查全率、 查准率 .

- **1 包住**: 在进行比较时, 若一个学习器的 $P-R$ 曲线被另一个学习器的曲线完全"**包住** " , 则可断言**后者的性能优于前者**, 如图 2.3 中学习去 A 的性能优于学习器 C ;
- **2 交叉**: 如果两个学习器的 $P-R$ 曲线发生了**交叉**, 例如图 2.3 中的 学习器 A 与 B , 则难以一般性地断言两者孰优孰劣, 只能在具体的查准率或查全率条件下进行比较. 这时一个比较合理的判据是比较 $P-R$ 曲线节面积的大小. 但这个值不太容易估算. 下面的**平衡点**就时一个综合考虑查准率、查全率的性能度量指标.

3 平衡点

"**平衡点**" (Break-Event Point， 简称 BEP) 就是这样一个度量, 它是**"查准率=查全率"**时的取值, 例如图 2.3 中学习器 C 的 BEP 是 0.65, 而基于 BEP 的比较, 可认为学习器 A **优于** B .

4 $F1$ 度量

但 BEP 还是过于简化了些, 更常用的是 $F1$ 度量, $F1$ 的定义如下:

$$F1=\frac{2\times P\times R}{P+R}=\frac{2\times TP}{\text{样例总数}+TP-TN} \tag{2.10}$$

5 $F1$ 的一般形式 F_β

在一些实际应用中, 对**查准率核查全率的重视程度有所不同**. 因此引入 $F1$ 度量的一般形式—F_β , 能让我们表达出对查准率/查全率的不同偏好, 具体定义为:

$$F_\beta=\frac{\left(1+\beta^2\right)\times P\times R}{\left(\beta^2\times P\right)+R} \tag{2.11}$$

其中 $\beta>0$ 度量了**查全率对查准率的相对重要性**. $\beta=1$ 时退化为标准的 $F1$; $\beta>1$ 时, **查全率**有更大影响; $\beta<1$ 时, **查准率**有更大影响.

6 全局性能度量

如何在 n 个二分类混淆矩阵上**综合考察查准率和查全率**

宏查准率和宏查全率

先在各混淆矩阵上**分别计算出查准率和查全率**, 记为 $(P_1, R_1), (P_2, R_2), \ldots, (P_n, R_n)$, 再**计算平均值**, 这样就得到"**宏查准率**" ($macro-P$) 、 "**宏查全率**" ($macro-R$) ， 以及相应的"**宏** $F1$ " ($macro-F1$):

$$\text{macro-P} = \frac{1}{n}\sum_{i=1}^{n} P_i \tag{2.12}$$

$$\text{macro-}R = \frac{1}{n}\sum_{i=1}^{n} R_i \tag{2.13}$$

$$\text{macro-}F1 = \frac{2 \times \text{macro-}P \times \text{macro-R}}{\text{macro}-P + \text{macro}-R} \tag{2.14}$$

微查准率和微查全率

先将各混淆矩阵的对应元素进行**平均**，得到 TP, FP, TN, FN 的平均值, 分别记为 $\overline{TP}, \overline{FP}, \overline{TN}, \overline{FN}$, 再基于这些平均值计算出"**微查准率** "($micro-P$) 、 "**微查全率**" ($micro-R$) 和"**微** $F1$" ($micro-F1$)

$$\text{micro}-P = \frac{\overline{TP}}{\overline{TP} + \overline{FP}} \tag{2.15}$$

$$\text{micro}-R = \frac{\overline{TP}}{\overline{TP} + \overline{FN}} \tag{2.16}$$

$$\text{micro}-F1 = \frac{2 \times \text{micro-}P \times \text{micro}-R}{\text{micro}-P + \text{micro}-R} \tag{2.17}$$

小结:

先定义查准率 P **和查全率** R **, 接着引入** $P-R$ **曲线, 如果完全包住, 这包住的为更优学习器, 其他情况这计算面积; 但** $P-R$ **曲线中计算面积不易求得, 接着引入比较"平衡点"(**BEP**), 即查准率=查全率的点; 但平衡点也过于简单, 于是引入** $F1$ **; 但** $F1$ **中, 查准率与查全率的地位相同的, 为了在实际使用中对查准率或查全率有所侧重, 引入** $F1$ **的一般形式** F_β **; 最后, 对于多个混淆矩阵, 如何计算全局的** $F1$ **, 分为宏和微.**

即 查准率 P 和查全率 $R \rightarrow P-R$曲线 →"平衡点"(BEP) → $F1$ →F_β

2.3.3 ROC 与 AUC

1 ROC曲线

很多学习器是**为测试样本产生一个实值**或**概率预测**, 然后将这个**预测值**与一个**分类阈值** (threshold) 进行**比较**, 若**大于阈值**则分为**正类**, **否则为反类**. 这个实值或概率预测结果的好坏, 直接决定了学习器的**泛化能力**.

根据这个实值或概率预测结果, 我们可将测试样本进行**排序**, **"最可能"是正例**的排在最**前面**, **"最不可能"是正例**的排在最**后面**. 这样, 分类过程就相当于在这个排序中以某个**"截断点**" (cut point) 将**样本分为两部分**, **前一部分判作正例, 后一部分则判作反例**.

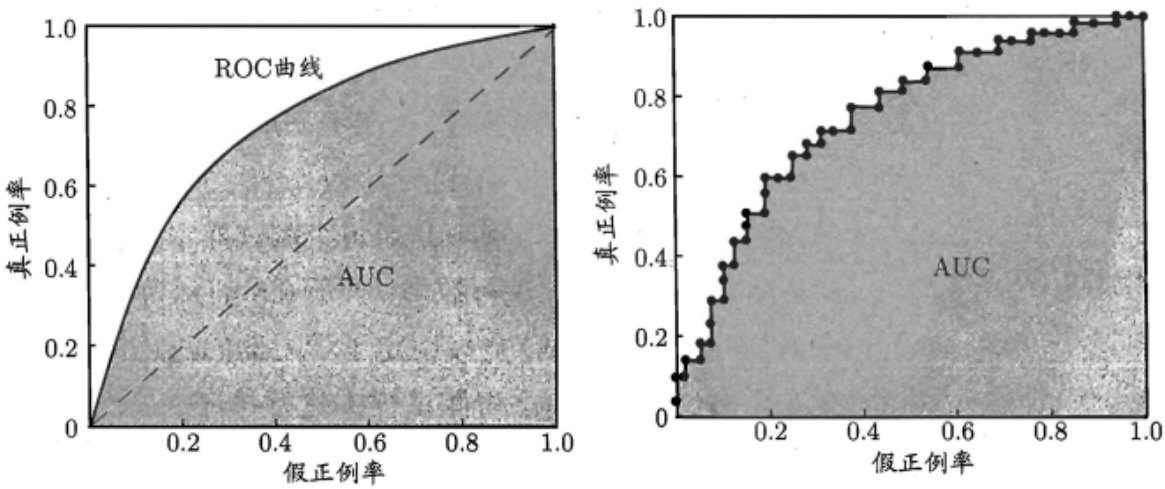
可根据任务需求来采用不同的截断点, 若我们更重视**"查准率"**, 则可选择排序中**靠前的位置进行截断**; 若更重视**"查全率"**, 则可选择**靠后的位置进行截断**. 因此, **排序本身的质量好坏**, 体现了综合考虑学习器在**不同任务下的"期望泛化性能"的好坏**, 或者说**"一般情况下"泛化性能的好坏**.

ROC 全称是**"受试者工作特征"** (Receiver Operating Characteristic) **曲线** . 与 $P-R$ 曲线相似, 我们根据学习器的**预测结果**对**样例进行排序**, 按此顺序**逐个**把样本作为**正例**进行**预测**, 每次计算出两个重要量的值, 分别以它们为横、纵坐标作图, 就得到了"ROC 曲线. 与 $P-R$ 曲线使用查准率、查全率为纵、横轴不同, ROC 曲线的**纵轴**是**"真正例率"** (True Positive Rate, 简称 TPR), **横轴**是**"假正例率"** (False PositiveRate，简称 FPR) , 基于表 2.1 中的符号, 两者分别定义为:

$$\mathrm{TPR}=\frac{TP}{TP+FN} \tag{2.18}$$

$$\mathrm{FPR}=\frac{FP}{TN+FP} \tag{2.19}$$

显示 ROC 曲线的图称为"ROC 图". 如图 $2.4(a)$



(a) ROC 曲线与 AUC　　(b) 基于有限样例绘制的 ROC 曲线与 AUC

图 2.4　ROC 曲线与 AUC 示意图

现实任务中通常是利用有限个测试样例来绘制 ROC 图，此时仅能获得**有限个** (真正例率, 假正例率) **坐标对**, 无法产生图 $2.4(a)$ 中的光滑 ROC 曲线, 只能绘制出如图 $2.4(b)$ 所示的近似 ROC 曲线.

2 AUC 曲线

进行学习器的比较时, 与 $P-R$ 图相似, 若一个学习器的 ROC 曲线被另 一个学习器的曲线完全**"包住"**, 则可断言**后者**的性能**优**于前者; 若两个学习器的 ROC 曲线发生**交叉**, 则难以一般性地断言两者孰优孰劣. 此时如果一定要进行比较, 则较为合理的判据是比较 ROC **曲线下的面积**, 即 AUC (Area UnderROC Curve), 如图 2.4 所示

AUC 的计算方法:

AUC 可通过对 ROC 曲线下各部分的面积求和而得. 假定 ROC 曲线是由坐标为 $\{(x_1,y_1),(x_2,y_2),\ldots,(x_m,y_m)\}$的点按序连接而形成 $(x_1=0,x_m=1)$, 参见图 $2.4(b)$, 则 AUC 可估算为:

$$\mathrm{AUC}=\frac{1}{2}\sum_{i=1}^{m-1}(x_{i+1}-x_i)\cdot(y_i+y_{i+1}) \tag{2.20}$$

排序损失:

AUC 考虑的是样本预测的排序质量, 因此它与**排序误差**有紧密联系. 给定 m^+ 个正例和 m^- 个反例, 令 D^+ 和D^- 分别表示正、反例集合, 则排序"**损失**" (loss)定义为:

$$\ell_{\mathrm{rank}}=\frac{1}{m^+m^-}\sum_{x^+\in D^+}\sum_{x^-\in D^-}\left(\mathbb{I}\left(f\left(\boldsymbol{x}^+\right)<f\left(\boldsymbol{x}^-\right)\right)+\frac{1}{2}\mathbb{I}\left(f\left(\boldsymbol{x}^+\right)=f\left(\boldsymbol{x}^-\right)\right)\right) \tag{2.21}$$

即考虑每一对正、反例, 若**正例的预测值小于反例**, 则**记一个"罚分"**, 若**相等**, 则**记** 0.5 **个"罚分"**.

ℓ_{rank} 对应的是 ROC 曲线之上的面积:

$$\mathrm{AUC}=1-\ell_{\mathrm{rank}} \tag{2.22}$$

> **注1:** 公式 (2.22) 不是很明白, 暂放.

2.3.4 代价敏感错误率与代价曲线

现实任务中, 会有这样的情况: **不同类型的错误所造成的后果不同**.

为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**" (unequal cost).

以二分类任务为例, 我们可根据任务的领域知识设定一个"**代价矩阵**" (cost matrix) , 如表 2.2 所示，其中 cost_{ij} **表示将第** i **类样本预测为第** j **类样本的代价**. 一般来说, $\mathrm{cost}_{ii}=0$; 若将第 0 类判别为第 1 类所造成的损失更大, 则$\mathrm{cost}_{01}>\mathrm{cost}_{10}$; 损失程度相差越大, $cost_{01}$ 与 cost_{10} 值的差别越大.

表 2.2　二分类代价矩阵

真实类别	预测类别	
	第 0 类	第 1 类
第 0 类	0	$cost_{01}$
第 1 类	$cost_{10}$	0

前面介绍的性能度量，都隐式地假设了**均等代价**. 在非均等代价下, 我们所希望的不再是**简单**地最小化**错误次数**, 而是希望**最小化"总体代价"**(total cost) .

若将表 2.2 中的第 0 类作为正类、第 1 类作为反类, 令 D^+ 与 D^- 分别代表样例集 D 的正例子集和反例子集, 则"**代价敏感**" (cost-sensitive)**错误率**为

$$\begin{aligned}E(f;D;\mathrm{cost})=&\frac{1}{m}\left(\sum_{\boldsymbol{x}_i\in D^+}\mathbb{I}\left(f\left(\boldsymbol{x}_i\right)\neq y_i\right)\times\mathrm{cost}_{01}\right.\\&\left.+\sum_{\boldsymbol{x}_i\in D^-}\mathbb{I}\left(f\left(\boldsymbol{x}_i\right)\neq y_i\right)\times\mathrm{cost}_{10}\right)\end{aligned} \tag{2.23}$$

代价曲线:

在非均等代价下, ROC 曲线不能直接反映出学习器的期望总体代价, 而**"代价曲线"** (cost curve) 则可达到该目的. 代价曲线图的**横轴是取值为** $[0,1]$**的正例概率代价**

$$P(+)\cos t=\frac{p\times\cos t_{01}}{p\times\cos t_{01}+(1-p)\times\cos t_{10}} \tag{2.24}$$

其中 p **是样例为正例的概率**

纵轴是取值为 $[0,1]$ **的归一化代价**

$$\mathrm{cost}_{\mathrm{norm}}=\frac{\mathrm{FNR}\ \times p\times\cos t_{01}+\mathrm{FPR}\times(1-p)\times\cos t_{10}}{p\times\cos t_{01}+(1-p)\times\mathrm{cost}_{10}} \tag{2.25}$$

其中 FPR 是式 (2.19) 定义的**假正例率**, $FNR=1-TPR$ 是**假反例率**

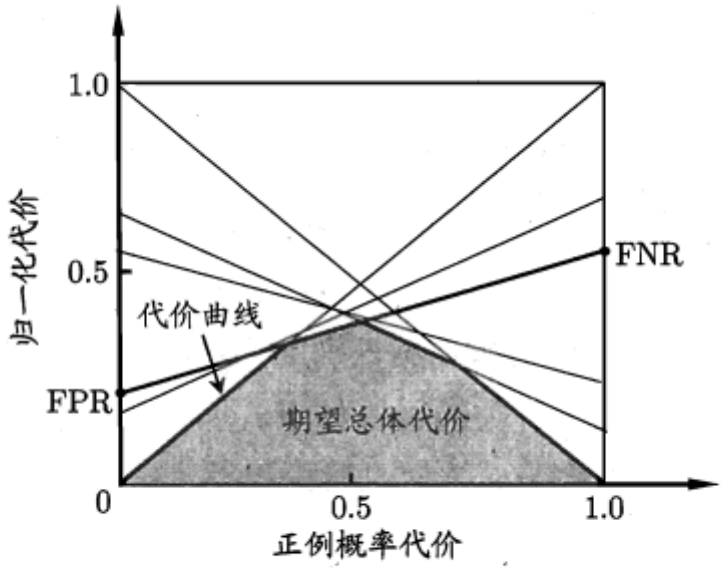


图 2.5 代价曲线与期望总体代价