# 机器学习



## 1.模型评估与选择

### 1.1 泛化能力的估计

数据集分为Training 与testing，以测试集上的“测试误差” testing error 作为期望风险/泛化误差的近似.

*泛化误差：在新样本上的误差；经验误差/训练误差：在训练集上的误差*

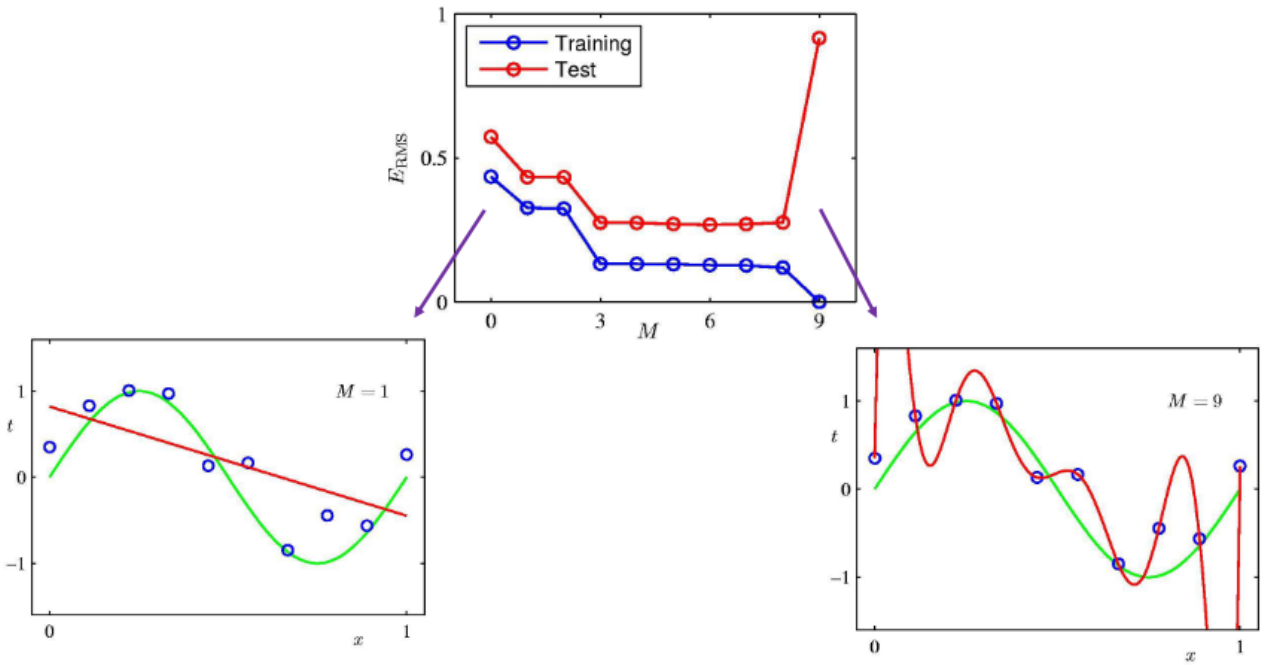
关键步骤：设定测试集，设定误差度量

过拟合与欠拟合：

*欠拟合指的是模型对于训练样本的一般性质尚未学好*

*过拟合指的是模型把训练样本学得"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。*

如下图，前半段是欠拟合，后半段拟合的多项式次数越高，经验误差可能会增大，是过拟合。过拟合不可避免




### 1.2 评估方法

1. 留出法：不能破坏数据集的数据分布，可采取分层抽样抽选出训练集和测试集。

2. 交叉验证法：k折交叉验证：一份作为测试集，剩下k-1份作为训练集。一般会采取十折，会多从把剩下k折分别作为测试集。
3. 自助法：多次放回抽样，可以生成与数据同样规模的训练集。样本在采样种始终采集不到的概率为

$$(1-\frac{1}{m})^m, \text{由于} \lim(1-\frac{1}{m})^m \to 1/e$$

会有36.8%的概率未出现在训练集中。在数据集较小，难以有效划分训练测试集的时候有用，但引入估计偏差，数据集数量较大时通常采用留出法与交叉验证法。

## 1.3 性能度量

衡量模型泛化能力的评价标准，反应了任务需求。对比不同模型的能力时，使用不同型性能度量会导致不同评判结果。

1. 回归任务：

   均方误差：

$$E(f;D)=\frac{1}{m}\sum_{i=1}^{m}(f(x_i)-y_i)^2$$

   平方绝对误差：

$$E(f;D)=\frac{1}{m}\sum_{i=1}^{m}|f(x_i)-y_i|$$

2. 分类任务：定义下面几个概念

- 混淆矩阵 *confusion matrix*

| | | Actual | |
|---|---|---|---|
| | | Class + | Class − |
| Predicted | Class + | TP | FP |
| | Class − | FN | TN |

- 准确度 *accuracy*

$$\frac{TP+TN}{TP+FP+FN+TN}$$

误差率 *Error rate*

$$\frac{FP+FN}{TP+FP+FN+TN}$$

- 精准度 Percision 查准率

$$TP\ /\ (TP+FP)$$

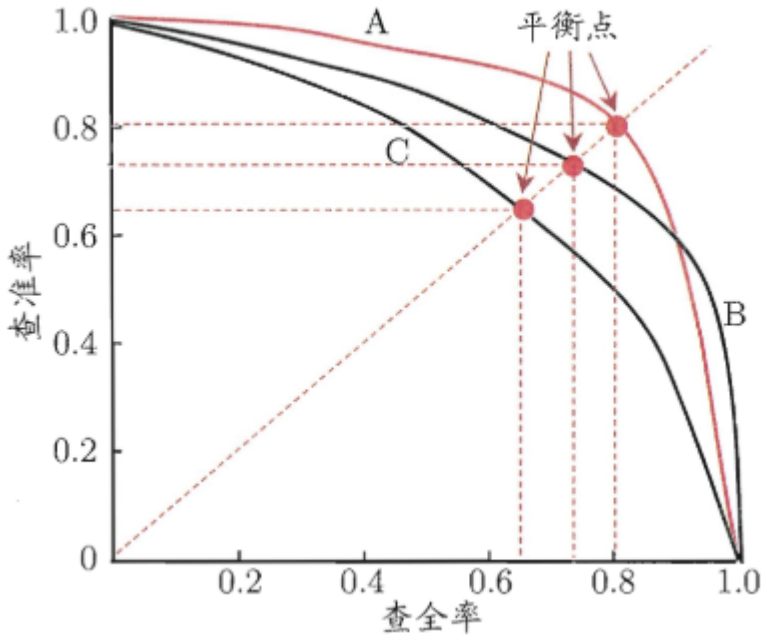
- 召回率 Recall 查全率

$$TP/(TP+FN)$$

查准率高，查全率会低，可以画出查准率与查全率的图像（P-R曲线）

画图方法：算法都会得出每个样本的置信度，通过置信度就可以对所有样本进行排序，再逐个样本的选择阈值，在该样本之前的都属于正例，该样本之后的都属于负例。每一个样本作为划分阈值时，都可以计算对应的precision和recall，那么就可以以此绘制曲线。

对比A B 难以比较好坏，所以引入下面几个度量：




平衡点（BEP）：查准率=查全率的取值

F1 Score：

$$F1 = 2\frac{precision * recall}{precision + recall}$$

F_β Score:β>1，查全率影响更大，β<1，查准率影响更大

$$F_\beta = (1 + \beta_2) * \frac{precision * recall}{\beta^2 * precision + recall}$$

考虑全局上面的衡量指标：（两种方法）

宏方式：先计算，后平均

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

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

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

微方式：先平均，后计算

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

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

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

ROC与AUC：绘制方法与P-R图思路一样，横纵坐标不一样，横坐标：FPR，纵坐标TPR。

$$TPR = \frac{TP}{TP + FN}$$
$$FPR = \frac{FP}{TN + FP}$$

AUC:ROC曲线对应的面积，可以用来对比算法的优劣。

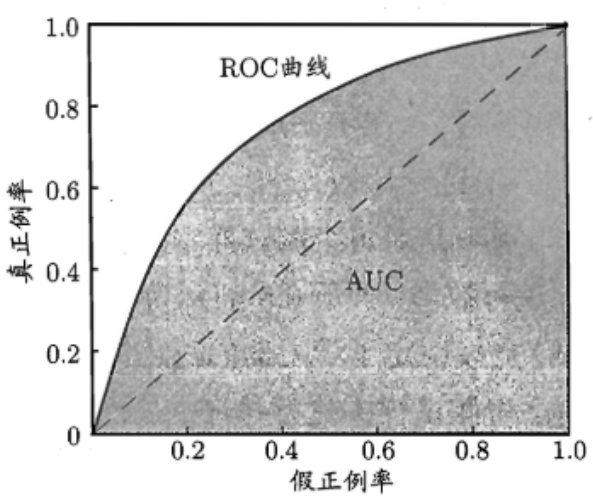
越靠近（0，1）点越好。

形式化地看, 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

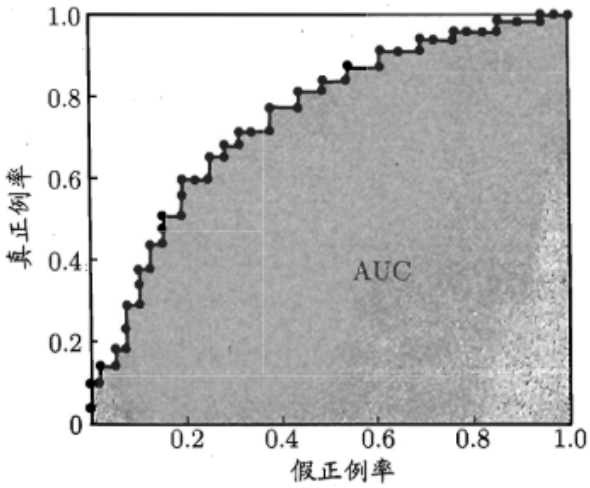
$$\ell_{rank} = \frac{1}{m^+ m^-} \sum_{\boldsymbol{x}^+ \in D^+} \sum_{\boldsymbol{x}^- \in D^-} \left( \mathbb{I}\left(f(\boldsymbol{x}^+) < f(\boldsymbol{x}^-)\right) + \frac{1}{2}\mathbb{I}\left(f(\boldsymbol{x}^+) = f(\boldsymbol{x}^-)\right) \right), \tag{2.21}$$

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”. 容易看出, $\ell_{rank}$ 对应的是 ROC 曲线之上的面积: 若一个正例在 ROC 曲线上对应标记点的坐标为 $(x, y)$, 则 $x$ 恰是排序在其之前的反例所占的比例, 即假正例率. 因此有

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




(a) ROC 曲线与 AUC




(b) 基于有限样例绘制的 ROC 曲线与 AUC

损失敏感场景：有些错误的代价不一样，引入“代价矩阵”

可根据任务的领域知识设定一个“损失矩阵” *cost matrix*

| 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 |

在非均等损失下的性能度量：

$$E(f;D;cost)$$

$$=\frac{1}{m}\left(\sum_{\boldsymbol{x}_i\in D^+}\mathbb{I}\left(f\left(\boldsymbol{x}_i\right)\neq y_i\right)\times cost_{01}\ +\sum_{\boldsymbol{x}_i\in D^-}\mathbb{I}\left(f\left(\boldsymbol{x}_i\right)\neq y_i\right)\times cost_{10}\right)$$