机器学习笔记

Pionpill [1]

本文档为作者学习《机器学习》[2]一书时的笔记，
主要对常用机器学习方法做了笔记。

2021 年 10 月 14 日

[1]笔名：北岸，电子邮件：673486387@qq.com，Github：https://github.com/Pionpill
[2]《机器学习》: 周立华，清华大学出版社，2019 年第二版

前言：

笔者为软件工程系在校本科生，空余时间自学了机器学习相关内容，在撰写该文档时，笔者并没有较深的统计学或数学知识，数学知识体系仅限于理工科必修的《高等数学》《概率统计》《离散数学》《线性代数》[1]四本书。数据方面自学了基本的数据处理方式，包括 python 基础，numpy,pandas 两个重要库以及一些常用数据分析工具，这方面的知识参考《利用 Python 进行数据分析》[2]一书。在模型建立方面，笔者有一定的老本，曾在 2021 年 MCMICM[3]中获 M 奖，主要负责编程，数据分析，排版。以上为笔者的基本介绍与知识系统说明，供各位学习时对比参考。

笔记的书写顺序与《机器学习》一致，章节采用了序言中的分法，将 1-3 章分为机器学习基础，4-10 章分为机器学习常用方法，11-16 章分为机器学习进阶方法。

原书对于一些名称或公式并没有详细解读，而是跳跃性说明，因此本文对于一些专有名词，会编写脚注，若经常用到或难以理解，会附上本人收集到的认为有用的学习链接，并放在附录中。若有知识超过本科数学 (理工科必修的四本书)，会被标记本科超纲。此外，本文在编写同时，也在对《统计学习方法》[4]一书做笔记，主要由于部分内容过于深奥，会定期求助于我的一些研究生学长学姐，也可能由于在补充知识体系而长期断更。

本书 (《机器学习》) 不同于《统计学习方法》，本书含有大段原作者的引导型举例，如果说《统计学习方法》一书的笔记涵盖了原书 10 单位的精华内容，那么本书的笔记则仅有 3 单位，在学习顺序上，也更推荐先看完本书再学习其他书籍。

由于本人非常不喜欢中式教科书式的死板，行文风格比较自由，还请谅解。

本笔记只是对原书的马虎概括与整理，如有疑问或需求，还请购买原书；本文所在 Github 仓库采用 GPL v3 协议，但请勿将本文商业使用。

2021 年 10 月 14 日

[1]笔者撰写时已经一年没有接触相关知识了，在撰写过程中过深的数学知识会格外备注

[2]原书英文名：《Data Analysis for Python》

[3]美国大学生数学建模竞赛

[4]《统计学习方法》: 李航，清华大学出版社，2019 年第二版

目录

I 机器学习基础

II 附录

I 机器学习基础

一、绪论

周志华先生的行文风格更贴近于生活，一些术语并没有给出标准解释，而是用例子来说明。因此这章的许多解释都是本人的总结，更为标准的解释可以参考《统计学习方法》一书的笔记。

1.1 基本术语

1.1.1 数据相关

- **数据集 (data set)**
  记录的集合，记录包括属性与值，通常形式为 (属性 1 = 值 1; 属性 2 = 值 2;...)。
- **示例/样本 (instance/sample)**
  数据集中的单条记录称为一个样本。有时整个数据集也称为一个样本[1]。
- **属性/特征 (attribute/feature)**
  在某方面的表现或性质的事项，对应值为属性值/特征值。
- **属性空间/样本空间/输入控件**
  属性张成的空间 (属性作笛卡尔积?)
- **特征向量 (feature vector)**
  一个示例

令 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m\}$ 表示包含 m 个示例的数据集，每个示例由 d 个属性描述，则每个示例 $\boldsymbol{x}_i = (x_{i1}; x_{i2}; \cdots; x_{id})$ 是 d 维样本空间 $\mathcal{X}$ 中的一个向量。$\boldsymbol{x}_i \in \mathcal{X}$；其中 x_{ij} 是第 j 个属性上的取值，d 称为样本**维度 (dimensionality)**。

1.1.2 训练相关

- **学习/训练 (learning/training)**
  从数据中学习模型的过程。
- **假设 (hypothesis)**
  学到的模型对应了关于数据的某种潜在的规则。
- **真相/真实 (ground-truth)**
  潜在规律。

[1]因为一般数据集不可能等于样本空间，故可看作对样本空间的采样

- **标记 (label)**
  关于示例结果的信息。如是好瓜/不是好瓜
- **样例 (example)**
  拥有了标记信息的示例。一般用 $(\boldsymbol{x}_i, y_i)$ 表示，其中 $y_i \in \mathcal{Y}$ 为标记。
- **输出空间 (label space)**
  所有标记的集合。

1.1.3 模型相关

- **分类 (classification)**
  输出空间包含的是离散值。若只涉及两个值，称为二分类 (binary classification)。通常称其中一个为正类 (positive class)，一个为反类 (negative class)。
- **回归 (regression)**
  输出空间包含的是连值。

一般预测任务是希望通过对训练集进行学习，建立一个从输入空间到输出空间的映射：$\mathcal{X} \to \mathcal{Y}$。对于二分类任务，通常令 $\mathcal{Y} = \{-1, +1\}$或$0, 1$；对多分类任务，$|\mathcal{Y}| > 2$；对回归任务，$\mathcal{Y} = \mathbb{R}$(实数集)。

- **聚类 (clustering)**
  将训练集中的数据分成若干组，每组称为一个簇 (cluster)。

根据训练数据是否拥有标记信息，学习任务大致可分为两大类：” 监督学习”(supervised learning) 和” 无监督学习”(unsupervised learning)。分类和回归是前者的代表，聚类是后者的代表。

- **泛化 (generalization)**
  学得模型适用于新样本的能力。泛化能力的大小是评判模型好坏的重要标志之一。

通常，假设样本空间中全体样本服从一个未知” 分布”(distribution)$\mathcal{D}$，我们获得的每个样本都是独立地从这个分布采样获得的，即**独立同分布**。

1.2 概念学习与归纳偏好

1.2.1 概念学习

- **归纳 (induction)**
  从许多个别的事物中概括出一般性概念、原则或结论的思维方法。
- **演绎 (deduction)**
  由一般原理推出关于特殊情况下的结论。

归纳学习 (induction learning)，即从样例中学习，有侠义与广义之分，广义的归纳学习大

体相当于从样例中学习，而狭义的归纳学习则要求从训练数据中学到概念，也称概念学习[2]。

1.2.2 归纳偏好

归纳偏好 (inductive bias) 指机器学习算法在学习过程中对某种类型假设的偏好。任何有效的机器学习必然有其归纳偏好。” 奥卡姆剃刀”[3]是一种常用的一般性原则来引导算法确定” 正确的” 偏好，但也有例外[4]。

1.3 发展历程

机器学习是人工智能研究发展到一定阶段的必然产物。

1.3.1 发展历史

- **二十世纪五十年代到七十年代 (推理期)**

认为只要赋予及其逻辑推理能力，机器就能具有智能。
A.Newell, H.Simon 的” 逻辑推理家” 程序证明了《数学原理》中的定理，并获得 1975 年图灵奖。

- **二十世纪七十年代中期 (知识期)**

由人总结出知识交给计算机

- **现在**

由机器自主学习

1.3.2 主流技术

- **符号主义学习**

以决策树和基于逻辑的学习为代表。直接模拟了人类对该你那进行判断的树形流程。基于逻辑学习的著名代表是归纳逻辑程序设计 (Inductive Logic Programming,ILP)。
决策树学习简单易用，ILP 具有很强的知识表示能力，可以容易地表达出复杂数据关系。由于表示能力太强，直接导致学习时假设空间太大，复杂度极高，问题规模大时难以有效学习。二十世纪九十年代中期这方面研究逐渐低调。

- **连接主义学习-神经网络**

二十世纪八十年代之前都不被重视，直到解出 NP 问题。
连接主义最大的局限是” 试错性”；简单地说就是需要手动调节参数，缺乏理论指导。二十世纪九十年代中期也逐渐没落。

- **统计学习**

代表性技术是向量机，核方法。在二十世纪九十年代中期闪亮登场。

---

[2]概念学习不是重点，具体例子可见书 P4

[3]若有多个假设与观察一致，则选最简单的那个。

[4]书 P6-9 详细解释了归纳偏好及其解决方案，但多为理解性内容，若需进一步理解请读者查看原书

- **深度学习**

深度学习狭义地说就是” 多层” 神经网络，属于连接主义。虽然缺乏理论支持，但在模型复杂度高，参数调节好的情况下，深度学习地结果往往十分有效。
深度学习地兴起与” 大数据” 和” 计算能力提高” 有密切关系。

二、模型评估与选择

2.1 经验误差与过拟合

2.1.1 经验误差

通常我们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 个样本错误，则错误率：

$$E = \frac{a}{m}$$

相应的，也有**精度** (accuracy):

$$A = 1 - \frac{a}{m}$$

我们把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"，学习器在训练集上的误差称为 "训练误差" 或 "**经验误差**"(empirical error)，在新样本上的误差称为 "**泛化误差**"(generalization error)。

2.1.2 过拟合

当学习器将训练样本训练得太好的时候，往往会将一些特异点当作一般性质，这样会导致模型的泛用性下降，在新样本上表现不好，这种现象在机器学习中称为 "过拟合"(overfitting)。与过拟合相对的也有 "欠拟合"(underfitting)。欠拟合问题很好解决，而过拟合则是机器学习的一大障碍。

2.2 评估方法

通常我们会选定一个**测试集**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" 作为泛化误差的近似。需要注意的是**测试集应尽可能与训练误差互斥，即测试样本尽量不在训练集中出现，未在训练集中使用过。**

一个数据集可以通过适当的处理，从中产生出训练集 S 和测试集 T。

2.2.1 留出法

"留出法"(hold-out) 直接将数据集 D 划分成两个互斥的集合：训练集 S，测试集 T。有 $D = S \cup T, S \cap D = \varnothing$。在 S 上训练出模型后，用 T 来评估期测试误差。

单次使用留出法往往是不够稳定可靠的，一般采用若干次随即划分，重复实验后取平均值作为评估结果。

常常将 2/3 - 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

“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 $D = D_1 \cup D_2 \cup \cdots \cup D_k, D_i \cap D_j = \varnothing (i \neq j)$，每个子集都尽可能保证数据分布一致性，即从 D 中分层采样[5]得到。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；这样可获得 k 组训练/测试集，从而进行 k 次实验，最终返回的是 k 个测试结果的均值。

交叉检验法评估结果的稳定性和保真性很大程序上取决于 k 的取值，为强调这一点，通常把交叉检验法称为 “k 折交叉检验”。k 最常用取值为 10，其他也常用 5，20。

图 1.2.1　10 折交叉检验示意图

与留出法类似，将数据集 D 划分为 k 个子集存在多种划分方式。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 这交叉检验结果的均值。常见的有 “10 次 10 折交叉检验”。

假定数据集 D 中有 m 个样本，若 $k = m$，则得到了交叉检验法的一个特例：留一法。留一法的训练结果往往比较准确，但在数据极大的情形下，考虑到算法开销，并不是十分适用。

2.2.3 自助法

我们希望评估的是用 D 训练出的模型，但在留出法和交叉检验中，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。而留一法又因为计算复杂度太高，往往难以接受。

“自助法”(bootstrapping) 直接以自助采样法[6]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'。

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。样本在 m 次采样中始终不被采到地概率是：

$$\lim_{m\to\infty}(1-\frac{1}{m})^m = \frac{1}{e} = 0.368 \tag{1.2.1}$$

即通过自助采样，初始数据集中约有 36.8% 的样本未出现在采样数据集 D' 中。于是我们可以将 D' 作为训练集。

自助法在数据量较小，难以有效划分训练/测试集时很有用；然而，自助法产生的数据集改变了数据集的分布，这回引入估计偏差。因此，在初始数据量足够时，留出法和交叉检验法更常用一些。

[5]先将总体的单位按某种特征分为若干次级总体（层），然后再从每一层内进行单纯随机抽样，组成一个样本。

[6]有放回地采样

2.2.4 调参与最终模型

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。

调参时需要注意，参数往往实在实数范围内取值，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**，例如 [0,0.2] 范围内以 0,05 为步长，则实际要评估的候选参数值就有 5 个，最终结果在折 5 个值中产生。显然这不是最佳结果，但却是在计算开销和性能估计之间的折中值。

在给定包含 m 个样本的数据集 D，在模型评估与选择过程中由于需要流出一部分数据进行评估测试，事实上，我们只使用了一部分数据训练模型。因此，在模型选择完成后，学习算法和参数配置已旋定，此时应该用数据集 D 重新训练模型。这个模型在训练过程中使用了所有 m 个模型，这才是最终的模型。

我们通常把学得模型在实际使用中遇到的数据称为测试数据，为了加以区别，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)。

2.3 性能度量

性能度量是指对模型泛化能力的评价。性能度量反映了任务需求，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。这也意味着模型的 “好坏” 是相对的。

在预测任务中，给定样例集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$。其中 y_i 是真实标记。评估学习器 f 的性能，就要把学习器预测结果 $f(\boldsymbol{x})$ 与真是标记 y 进行比较。

回归任务最常用的性能度量是 “均方误差”:

$$E(f; D) = \frac{1}{m}\sum_{i=1}^{m}(f(\boldsymbol{x}_i) - y_i)^2 \tag{1.2.2}$$

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：

$$E(f; D) = \int_{x\sim D}(f(\boldsymbol{x}) - y)^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{1.2.3}$$

2.3.1 错误率与精度

错误率是分类错误的样本占样本总数的比例，精度则相反。错误率定义：

$$E(f; D) = \frac{1}{m}\sum_{i=1}^{m}\mathbb{I}(f(\boldsymbol{x}_i) \neq y_i) \tag{1.2.4}$$

精度定义：

$$
\begin{aligned}
E(f;D) &= \frac{1}{m}\sum_{i=1}^{m}\mathbb{I}(f(\boldsymbol{x}_i)=y_i) \\
&= 1-E(f;D)
\end{aligned}
\tag{1.2.5}
$$

更一般地，对于数据分布 $\mathcal{D}$ 和概率密度函数 $p(\cdot)$, 错误率和精度有如下定义：

$$
E(f;\mathcal{D}) = \int_{x\sim D}\mathbb{I}(f(\boldsymbol{x})\neq y)p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{1.2.6}
$$

$$
\begin{aligned}
acc(f;\mathcal{D}) &= \int_{x\sim D}\mathbb{I}(f(\boldsymbol{x})=y)p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \\
&= 1-E(f;\mathcal{D})
\end{aligned}
\tag{1.2.7}
$$

2.3.2 查准率，查全率与 F1

对于二分类问题，可以将样例根据其真是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，显然有 TP+FP+TN+FN= 样例总数。

表 1.1　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)

准确率/查准率，即预测结果为正例且正确的比例:

$$
P = \frac{TP}{TP+FP} \tag{1.2.8}
$$

召回率/查全率，即正例被预测出的比例:

$$
R = \frac{TP}{TP+FN} \tag{1.2.9}
$$

查准率与查全率是一对矛盾的度量，一般来说，查准率高时，查全率往往比较低；反之亦然。我们以查准率为纵轴，查全率为横轴作图，就得到了查准率-查全率曲线，简称 “P-R 图”

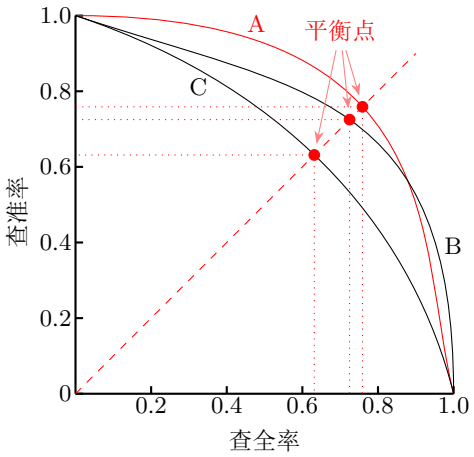

图 1.2.2　P-R 曲线与平衡点示意图

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，如上图的学习器 A 优于学习器 C。如果两个学习器发生了交叉，如学习器 A 和 B，则很难断言两者孰优孰劣。只能在具体的查准率或查全率条件下进行比较。然而，在很多情形下，人们希望比较学习器 A 和 B，则可以通过曲线下面积的大小比较，它在一定程度上表征了学习器在查准率和查全率上取得相对 “双高” 的比例。但是这个值不太容易估算，因此会采用一些替代的方法度量。

“平衡点”(Break-Even Point, 检查 BEP) 是 “查准率 = 查全率” 时的取值，可以通过这点对应的查全率 (或查准率) 进行比较。

但 BEP 还是过于简化了，更常用的是 $F1$ 度量[7]。

$$F1 = \frac{2 \times P \times R}{P + R} = \frac{2 \times TP}{\text{样例总数} + TP - TN} \tag{1.2.10}$$

在一些应用中，对查准率和查全率的重视程度有所不同、此时我们需要 $F1$ 度量的一般形式 F_β 度量[8]：

$$F_\beta = \frac{(1 + \beta^2) \times P \times R}{(\beta^2 \times P) + R} \tag{1.2.11}$$

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 $F1$; $\beta > 1$ 时查全率有更大影响；$\beta < 1$ 时查准率有更大影响。

一种直接的做法是先在各混淆矩阵上分别计算出查准率与查全率，记为 $(P_1, R_1), (P_2, R_2), \cdots, (P_n, R_n)$, 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)

[7] $F1$ 是基于调和平均数定义的：$\frac{1}{F1} = \frac{1}{2} \cdot (\frac{1}{P} + \frac{1}{R})$

[8] F_β 是基于加权调和平均数定义的：$\frac{1}{F_\beta} = \frac{1}{1+\beta^2} \cdot (\frac{1}{P} + \frac{\beta^2}{R})$

$$marco-P=\frac{1}{n}\sum_{i=1}^{n}P_i \tag{1.2.12}$$

$$marco-R=\frac{1}{n}\sum_{i=1}^{n}R_i \tag{1.2.13}$$

$$marco-F1=\frac{2\times macro-P\times macro-R}{macro-R+macro-R} \tag{1.2.14}$$

还可以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 $\overline{TP}, \overline{FP}, \overline{TN}, \overline{FN}$ 再基于这些平均值计算出 “微查准率”(micro-P)，“微查全率”(micro-R), 以及相应的 “微 F1”(micro-F1)

$$micro-P=\frac{\overline{TP}}{\overline{TP}+\overline{FP}} \tag{1.2.15}$$

$$micro-R=\frac{\overline{TP}}{\overline{TP}+\overline{FN}} \tag{1.2.16}$$

$$micro-F1=\frac{2\times macro-P\times macro-R}{macro-R+macro-R} \tag{1.2.17}$$

2.3.3 ROC 与 AUC

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，若大于阈值则为正类，否则为反类。这个实质或概率预测结果的好坏，直接决定了学习器的泛化能力。实际上，根据这个实值或概率预测的结果，我们可将测试样本进行排序，“最可能” 是正例排在最前面，“最不可能” 是正例排在最后面。这样，分类过程就相当于在这个排序中以某个 “截断点” 将样本分成了两部分，前一部分判作正例，后一部分判作反例。

在不同的应用任务中，我们可根据任务需求不同采用不同的截断点，例如若我们更中式 “查准率”，则可选择排序中靠前的位置进行截断。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。ROC 曲线则是从这个角度出发来研究学习器泛化性能的有力工具。

ROC 全称 “受试者工作特征” 曲线，与 P-R 曲线类似，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”。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 Positive Rate, 简称 FPR)，两者定义为：

$$TPR=\frac{TP}{TP+FN} \tag{1.2.18}$$

$$FPR=\frac{FP}{TN+FP} \tag{1.2.19}$$

下图给出了一个示例，显然，对角线对应于“随即猜测”模型，而点 (0,1) 则对应于将所有正例排在所有反例之前的“理想模型”。

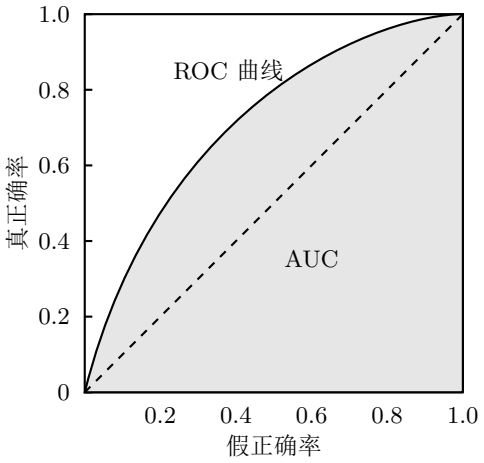

图 1.2.3 ROC 与 AUC

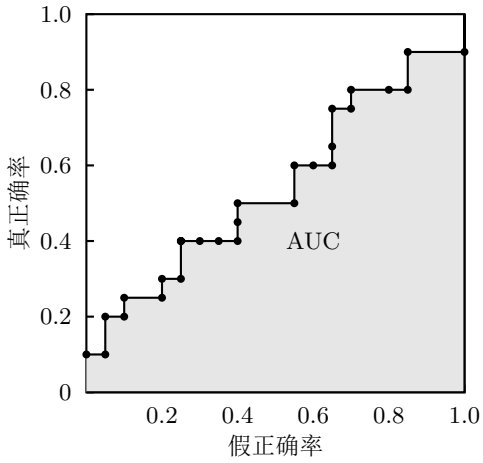

图 1.2.4 基于有限样例的 ROC 曲线与 AUC

现实中往往不能产生光滑曲线，只能产生近似 ROC 曲线。绘制过程如下：给定 m^+ 个正例和 m^- 个反例。根据学习器预测结果对样例进行排序，然后把分类阈值设为最大，即把所有样例预测为反例，此时真正利率和假正例率均为 0，在坐标 (0,0) 处标记一个点。然后，将分类阈值依次设为每个样例的预测值，即依次将每个样例划分为正例。设前一个标记点坐标为 (x,y), 当前若为真正例，则对应标记点的坐标为 $(x, y+\frac{1}{m^+})$；当前若为假正例，则对应标记点的坐标为 $(x+\frac{1}{m^-}, y)$，然后用线段连接相邻点即得。

进行学习器的比较时，与 P-R 曲线类似，若一个学习器的 ROC 曲线完全被另一个学习器的曲线“包住”则可断言后者的性能由于前者；若两个学习器的 ROC 曲线发生交叉，则难以断言。此时若一定需要断言，则可以通过比较 ROC 曲线下方的面积，即 AUC(Area Under ROC Curve)。

从定义可知，AUC 可通过对 ROC 曲线下各部分的面积求和而得。假设 ROC 曲线是由坐标 (x_i, y_i) 点组成的，则 AUC 可估算为：

$$AUC = \frac{1}{2}\sum_{i=1}^{m-1}(x_{i+1}-x_i)\cdot(y_i+y_{i+1}) \tag{1.2.20}$$

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^- 个返例，令 D^+ 和 D^- 分别表示正，反例集合，则排序“损失”定义为：

$$l_{rank} = \frac{1}{m^+m^-}\sum_{\mathbf{x}^+\in D^+}\sum_{\mathbf{x}^-\in D^-}\left(\mathbb{I}(f(\mathbf{x}^+)<f(\mathbf{x}^-))+\frac{1}{2}\mathbb{I}(f(\mathbf{x}^+)=f(\mathbf{x}^-))\right) \tag{1.2.21}$$

即考虑每一对正，反例，若正例的预测值小与返例，则记一个“罚分”，若相等，则记 0.5 个“罚分”。可以看出 l_{rank} 对应的就是 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返例所占的比例，即假正例率，则有：

$$AUC = 1 - l_{rank} \tag{1.2.22}$$

2.3.4 代价敏感错误率与代价曲线

在现实任务中，不同错误所造成的后果往往不同。例如错误地将健康人诊断为病人与错误地将病人诊断为健康人所带来的后果不可等价。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”(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”，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 $cost_{ii} = 0$。

表 1.2　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

在非均等代价中，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”。若将表 1.2 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” 错误率为

$$E(f;D;cost) = \frac{1}{m}\left(\sum_{m\in D^+}\mathbb{I}(f(\mathbf{x}_i)\neq y_i)\times cost_{01} + \sum_{m\in D^-}\mathbb{I}(f(\mathbf{x}_i)\neq y_i)\times cost_{10}\right) \tag{1.2.23}$$

类似的，可给出基于分布定义的代价敏感错误率，以及其他一些性能度量如精度的代价敏感版本。若令 $cost_{ij}$ 中 i, j 取值不限于 0,1，则可定义出多分类任务的代价敏感性能度量。

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” 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

$$P(+)cost = \frac{p\times cost_{01}}{p\times cost_{01} + (1-p)\times cost_{10}} \tag{1.2.24}$$

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。

$$cost_{norm} = \frac{FNR\times p\times cost_{01} + FPR\times(1-p)\times cost_{10}}{p\times cost_{01} + (1-p)\times cost_{10}} \tag{1.2.25}$$

其中 FPR 是假正例率，FNR=1-TPR 是假反例率。代价曲线的绘制很简单：ROC 曲线上每一点对应代价平面上的一条线段，设 ROC 曲线上点的坐标是 (FPR,TPR)，则可相应计算出 FNR，然后在代价平面上绘制一条从 (0,FPR) 到 (1,FNR) 的线段，线段下的面积即表示了该条件下的期望总体代价；如此将 ROC 曲线上的每个点转换为代价平面上的一条线段，然后取所有线段的下界，围成的面积即为在所有条件下学习器的期望总体代价。

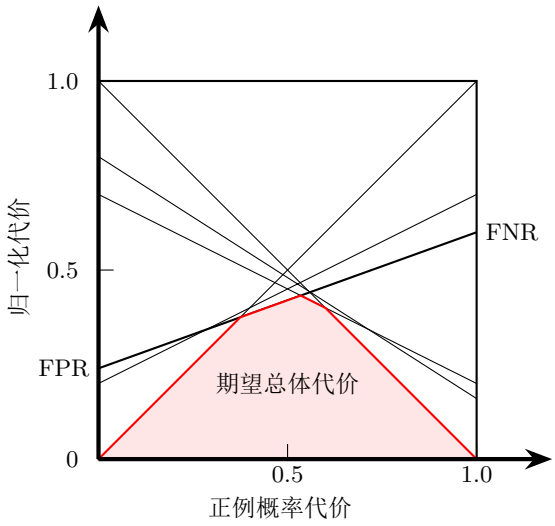

图 1.2.5　代价曲线与期望总体代价

2.4 比较检验

机器学习的“比较”十分复杂，常用的方法有统计假设检验 (hypothesis test)。基于假设检验结果我们可推断出，若在测试集上观察到学习器 A 比 B 好，则 A 的泛化性能是否在统计意义上优于 B。这主要有以下两种最基本的假设检验。

2.4.1 假设检验

假设检验中的“假设”是对学习器泛化错误率分布的某种判断或猜想，例如“$\epsilon = \epsilon_0$”。现实任务中我们并不知道学习器的泛化错误率，只能获知其测试错误率 $\hat{\epsilon}$。泛化错误率与测试错误率未必相同，但直观上两者相近，可根据测试错误率估算出泛化错误率的分布。

泛化错误率为 ϵ 的学习器在一个样本上犯错的概率是 ϵ；测试错误率 $\hat{\epsilon}$ 意味着在 m 个测试样本中恰有 $\hat{\epsilon}\times m$ 个被误分类。假定测试样本是从样本总体分布中独立采样得到的，那么泛化错误率为 ϵ 的学习器将其中 m' 个样本误分类，其余样本全都分类正确的概率是 $\binom{m}{m'}\epsilon^{m'}(1-\epsilon)^{m-m'}$；由此可估算出其恰将 $\hat{\epsilon}\times m$ 个样本误分类的概率如下所示，这也表达式在包含 m 个样本的测试集上，泛化错误率为 ϵ 的学习器被测得测试错误为 $\hat{\epsilon}$ 的概率：

$$P(\hat{\epsilon};\epsilon) = \binom{m}{\epsilon \hat{\times} m}\epsilon^{\hat{\epsilon}\times m}(1-\epsilon)^{m-\hat{\epsilon}\times m} \tag{1.2.26}$$

给定测试错误率，则解 $\partial P(\hat{\epsilon};\epsilon)/\partial\epsilon = 0$ 可知，$P(\hat{\epsilon};\epsilon)$ 在 $\epsilon = \hat{\epsilon}$ 时最大，$|\epsilon - \hat{\epsilon}|$ 增大时 $P(\hat{\epsilon};\epsilon)$ 减小。这符合二项分布，如下图所示，若 $\epsilon = 0.3$，则 10 个样本中测得 3 个被误分类的概率最大。

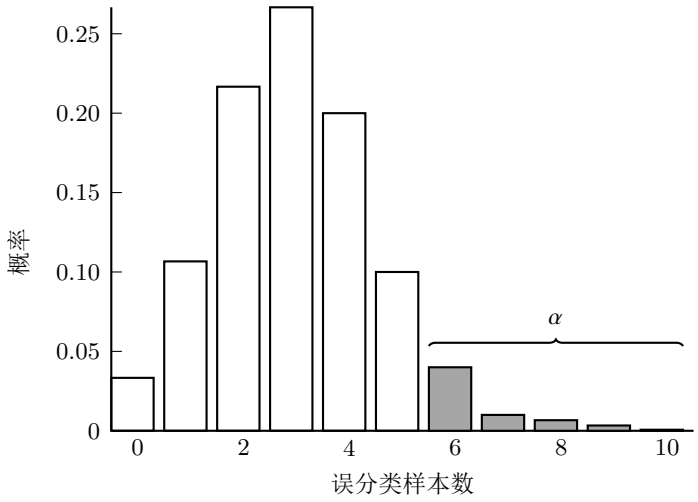

图 1.2.6　二项分布示意图

我们可使用 “二项检验” 来对 “$\epsilon \leqslant 0.3$”(即泛化误差是否不大于 0.3) 这样的假设进行检验。更一般地，考虑假设 “$\epsilon \leqslant \epsilon_0$”，则在 $1-\alpha$ 的概率内所能观测到的最大错误率如下式计算。这里 $1-\alpha$ 反映了结论的 “置信度”，直观地来看，相应于上图中的非阴影区域[9]。

$$\bar{\epsilon} = \min \epsilon \text{s.t.} \sum_{i=\epsilon_0 \times m+1}^{m} \binom{m}{i} \epsilon^i (1-\epsilon)^{m-i} < \alpha \tag{1.2.27}$$

此时若测试错误率 $\hat{\epsilon}$ 小于临界值 $\bar{\epsilon}$，则根据二项检验可得出结论：在 α 的显著度下，假设 “$\epsilon \leqslant \epsilon_0$” 不能被拒绝，即能以 $1-\alpha$ 的置信度认为，学习器的泛化错误率不大于 ϵ_0；否则该假设可被拒绝，即在 α 的显著度下可认为学习器的泛化错误率大于 ϵ_0。

在很多时候我们并非仅做一次留出法估计，而是通过多次重复留出法或是交叉验证法等进行多次训练/测试，这样会得到多个测试错误率，此时可使用 “t 检验”。假定我们得到了 k 个测试错误率，$\hat{\epsilon}_1, \hat{\epsilon}_2, \cdots, \hat{\epsilon}_k$，则平均测试错误率 μ 和方差 σ^2 为

$$\mu = \frac{1}{k} \sum_{i=1}^{k} \hat{\epsilon}_i \tag{1.2.28}$$

$$\sigma^2 = \frac{1}{k-1} \sum_{i=1}^{k} (\hat{\epsilon}_i - \mu)^2 \tag{1.2.29}$$

考虑到这 k 个测试错误率可看作泛化错误率 ϵ_0 的独立采样，则变量

$$\tau_t = \frac{\sqrt{k}(\mu - \epsilon_0)}{\sigma} \tag{1.2.30}$$

服从自由度为 $k-1$ 的 t 分布。

对假设 “$\mu = \epsilon_0$” 和显著度 α，我们可以计算出当测试错误率均为 ϵ_0 时，在 $1-\alpha$ 概率内能观测到的最大错误率，即临界值。这里考虑双边假设，如下图所示，两边各有 $\alpha/2$ 的面积；假定阴影部分范围分别为 $(-\infty, t_{-\alpha/2}]$ 和 $[t_{\alpha/2}, \infty]$，若 τ_t 位于临界值范围 $[t_{-\alpha/2}, t_{\alpha/2}]$ 内，则

[9] s.t. 是 subject to 的缩写

不能拒绝假设 “$\mu = \epsilon_0$”，即可认为泛化错误率为 ϵ_0，置信度为 $1-\alpha$；否则可拒绝该假设，即在该显著度下可认为泛化错误率与 ϵ_0 有显著不同。α 常用的取值有 0.05 和 0.1。

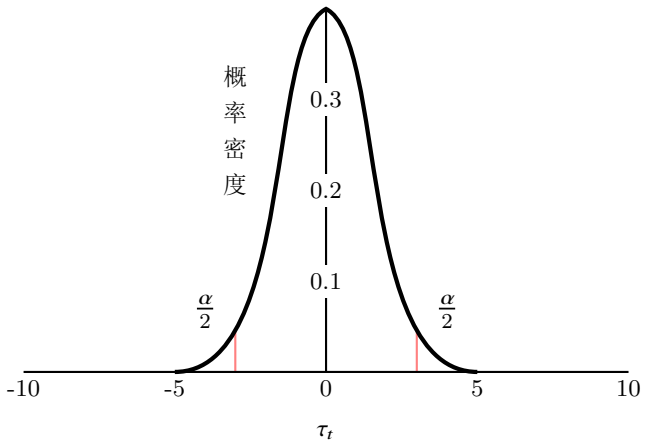

图 1.2.7　t 分布示意图

上面介绍的两种方法都是对单个学习器泛化性能的假设进行检验，而在现实任务中，更多时候我们需对不同学习器的性能比较，虾米那介绍适用于此类情况的假设检验方法。

2.4.2 交叉验证 t 检验

对于两个学习器 A 和 B,若我们使用 k 折交叉验证法得到的测试错误率分别为 $\epsilon_1^A, \epsilon_2^A, \cdots, \epsilon_k^A$ 和 $\epsilon_1^B, \epsilon_2^B, \cdots, \epsilon_k^B$，其中 $\epsilon_i^A \epsilon_i^B$ 是在相同的第 i 折训练/测试集上得到的结果，则可用 k 折交叉检验 “成对 t 检验” 来进行比较检验。这里的思想是：若两个学习器的性能相同，则它们使用相同的训练/测试集得到的测试错误率应相同，即 $\epsilon_i^A = \epsilon_i^B$。

具体地说，对 k 折交叉验证产生的 k 对测试错误率：先对没对结果求差，$\Delta_i = \epsilon_i^A - \epsilon_i^B$；若两个学习器性能相同，则差值均值应为零。因此，可根据差值 $\Delta_1, \Delta_2, \cdots, \Delta_k$ 来对 “学习器 A 与 B 性能相同” 这个假设做 t 检验，计算出插值的均值 μ 和方差 σ^2，在显著度 α 下，若变量

$$\tau_t = |\frac{\sqrt{k}\mu}{\sigma}| \tag{1.2.31}$$

小于临界值 $t_{\alpha/2,k-1}$，则假设不能被拒绝，即认为两个学习器的性能没有显著差别；否则可认为两个学习器的性能有显著差别，且平均错误率较小的学习器性能较优。这里 $t_{\alpha/2,k-1}$ 是自由度为 $k-1$ 的 t 分布上尾部累积分布为 $\alpha/2$ 的临界值。

欲进行有效的假设检验，一个重要的前提是测试错误率均为泛化错误率的独立采样。然而，通常情况下由于样本有限，在使用交叉验证等实验估计方法时，不同轮次的训练集会有一定程度的重叠，这就使得测试错误率实际上并不独立，会导致过高估计假设成立的概率。为缓解这一问题，可采用 “5×2 交叉验证” 法。

5×2 交叉验证是做 5 次 2 折交叉验证，在每次 2 折交叉验证之前随机将数据打乱，使得 5 次交叉验证中的数据划分不重复。对两个学习器 A 和 B，第 i 次 2 折交叉验证将产生两对测试错误率，我们对他们分别求差，得到第 1 折上的插值 Δ_i^1 和第 2 折上的插值 Δ_i^2。为缓解测

试错误率的非独立性，我们仅计算第 1 次 2 折交叉验证的两个结果的平均值 $\mu = 0.5(\Delta_1^1 + \Delta_1^2)$，但对每次 2 折实验的结果都计算出其方差 $\sigma_i^2 = \left(\Delta_i^1 - \frac{\Delta_i^1+\Delta_i^2}{2}\right) + \left(\Delta_i^2 - \frac{\Delta_i^1+\Delta_i^2}{2}\right)$。变量

$$\tau_t = \frac{\mu}{\sqrt{0.2\sum_{i=2}^{5}\sigma_i^2}} \tag{1.2.32}$$

服从自由度为 4 的 t 分布，其双边检验的临界值 $t_{\alpha/2,5}$，当 $\alpha = 0.05$ 时为 2.776，$\alpha = 0.1$ 时为 2.132。

2.4.3 McNemar 检验

对二分类问题，使用留出法不仅可估计出学习器 A 和 B 的测试错误率，还可以获得两学习器分类结果的差别。

表 1.3　两学习器分类差别列联表

算法 B	算法 A	
	正确	错误
正确	e_{00}	e_{01}
错误	e_{10}	e_{11}

若我们做的假设是两学习器性能相同，则应有 $e_{01} = e_{10}$，那么变量 $|e_{01} = e_{10}|$ 应当服从正态分布，McNemar 检验考虑变量

$$\tau_{\chi^2} = \frac{(|e_{01} - e_{10}| - 1)^2}{e_{01} + e_{10}} \tag{1.2.33}$$

服从自由度为 1 的 χ^2 分布，即标准正态分布变量的平方。给定显著度 α，当以上变量值小于临界值 χ_α^2 时，不能拒绝假设，即认为两学习器的性能没有显著差别；否则拒绝假设，认为有显著差别，且平均错误率较小的那个学习器性能较优。自由度为 1 的 χ^2 检验的临界值当 $\alpha = 0,05$ 时为 3.8415，$\alpha = 0.1$ 时为 2.7055。

II 附录

一、主要符号表

表 2.1 主要符号表

符号	意义	符号	意义
x	标量	$\vec{x}$	向量
x	变量集	$\mathbf{A}$	矩阵
$\mathbf{I}$	单位阵	$\mathcal{X}$	样本空间或状态空间
$\mathcal{D}$	概率分布	D	数据样本 (数据集)
$\mathcal{H}$	假设空间	H	假设集
$\mathcal{E}$	学习算法	$(\cdot,\cdot,\cdot)$	行向量
$(\cdot;\cdot;\cdot)$	列向量	$(\cdot)^T$	向量转置
$\{\cdots\}$	集合	$\lvert\{\cdots\}\rvert$	集合中元素的个数
$\lvert\lvert\cdot\rvert\rvert_p$	L_p 范数，p 缺省时为 L_2 范数	$P(\cdot), P(\cdot\lvert\cdot)$	(条件) 概率质量函数
$p(\cdot), p(\cdot\lvert\cdot)$	(条件) 概率密度函数	$\sup(\cdot)$	上确界
$\mathbb{I}(\cdot)$	指数函数，$\cdot$ 为真假分别取 1,0	$\text{sign}(\cdot)$	符号函数, 分别取-1,0,1
$\mathbb{E}_{\cdot\sim D}[f(\cdot)]$	函数 $f(\cdot)$ 对 $\cdot$ 在分布 $\mathcal{D}$ 下的数学期望；意义明确时省略测 $\mathcal{D}\cdot$		