《模式识别》读书报告 第一部分 概述

第一章 绪论

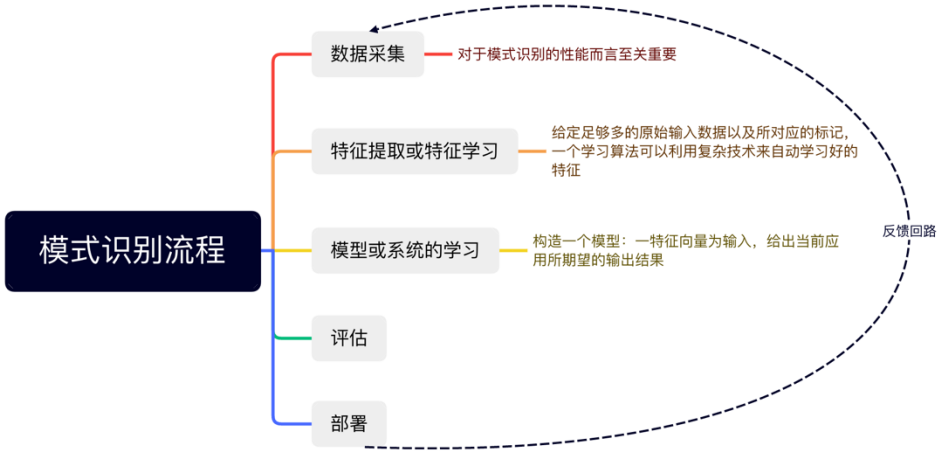
1.1 定义

模式识别在维基百科上的定义为：模式识别是机器学习的一个分支，尽管在某些情况可以被视为机器学习的同义词，而模式识别更加侧重于对规律的识别。

模式识别领域专注使用计算机算法来自动的发现数据的规律，并利用这些规律来采取行动，将数据划分到不同的类别中。

模式：指代不同应用中的各种有用的、有组织的信息。一些研究人员也使用“规律性”来作为模式的同义词。模式识别的方法或系统是不需要人类参与而依靠其自身自动完成的。

1.2 一个典型的模式识别流程：



1.3 模式识别 vs 机器学习

模式识别中的重要一步即模型学习通常会被当成一个机器学习任务，同时特征学习，也被称为表示学习。

传统机器学习算法更加关注抽象的模型学习部分，机器学习的研究人员的一个重要研究课题是关注算法的理论保证。

第二章 数学背景知识

2.1 奇异值分解 SVD

特征分解与奇异值分解密切相关，令 X 为一个 m*n 的矩阵，则 X 的 SVD 为：

$$X = U\Sigma V^T$$

其中 U 为一个 m*m 的矩阵，Σ是一个 m*n 的矩阵，其非对角线上的元素全为 0，V 是一个 n*n 的矩阵。

若存在标量σ和两个向量$u \in R^m$和$v \in R^n$(均为单位向量)同时满足下述两个等式：

$$Xv = \sigma u \ 并且 \ X^T u = \sigma v$$

则称σ是 X 的奇异值，u 和 v 分别为相应的左、右奇异向量。奇异值总是非负的。

SVD 可以找到所有的奇异值和奇异向量，U 的列被称为 X 的左奇异向量，V 的列被称为 X 的右奇异向量。矩阵 U 和 V 是正交的。Σ对角线元素为对应的奇异值。

X 的左奇异向量为XX^T的特征向量，X 的右奇异向量为X^TX的特征向量，X 的非零奇异值是XX^T和X^TX非零特征值的平方根。XX^T和X^TX非零特征值相同。

2.2 优化与矩阵微积分

非正式地讲, 给定一个代价 (或目标) 函数 $f(\boldsymbol{x}) : \mathcal{D} \mapsto \mathbb{R}$, 数学优化的目标是在域 $\mathcal{D}$ 中找到 $\boldsymbol{x}^\star$, 使得对于任意 $\boldsymbol{x} \in \mathcal{D}$ 都有 $f(\boldsymbol{x}^\star) \leqslant f(\boldsymbol{x})$. 这类优化问题被称为最小化 (minimization) 优化问题, 通常记为

$$\min_{\boldsymbol{x}\in\mathcal{D}} f(\boldsymbol{x}) . \tag{2.67}$$

使得 $f(\boldsymbol{x})$ 最小的解 $\boldsymbol{x}^\star$ 是 f 的一个最小值解, 记为

$$\boldsymbol{x}^\star = \arg\min_{\boldsymbol{x}\in\mathcal{D}} f(\boldsymbol{x}) . \tag{2.68}$$

2.3 凸优化与凹优化

当 f(x)是一个域为R^d的凸函数，任何局部极小也是全局极小。凸最小化问题就是在一个凸集上最小化凸目标函数。在凸最小化中，任何局部极小必是全局极小。

琴生不等式:

令 f(x)为一个定义在凸集 S 上的凸函数，$x_1, x_2, \ldots, x_n$为 S 中的点。权重$w_1, w_2, \ldots, w_3$满足$w_i \geq 0$,以及$\sum_{i=1}^{n} w_i = 1$，琴生不等式表明

$$f\left(\sum_{i=1}^{n} w_i x_i\right) \leq \sum_{i=1}^{n} w_i f(x_i)$$

f(x)为一个定义在凸集 S 上的凹函数:

$$f\left(\sum_{i=1}^{n} w_i x_i\right) \geq \sum_{i=1}^{n} w_i f(x_i)$$

对于一个涉及多个变量的标量函数，如果它是连续且二次可微的，则其二阶偏导构成了一个方阵，称之为 Hessian 矩阵，若 Hessian 是半正定的，则这样一个函数是凸的。例如若 A 是半正定的，则 $f=x^TAx$ 是凸的。

第三章　模式识别系统概述

3.1 训练或学习

在形式化的描述下, 为了学习这种映射 (mapping), 我们可以获取 n 个实体对 $\boldsymbol{x}_i$ 和 y_i. 每一对 $(\boldsymbol{x}_i, y_i)$ 包含第 i 个训练样本 (training example) $\boldsymbol{x}_i$ (也被称为第 i 个训练示例, training instance) 及其所对应的标记 (label) y_i. 所有训练示例及其对应的标记的集合构成了训练集 (training set). 我们任务中的第一个阶段被称为训练 (training) 或学习 (learning) 阶段, 需要发现如何从任意样本 $\boldsymbol{x}$ 推演其标记 y.

3.2 测试或预测

在关于学习得到的映射 f 的第二种应用场景中, 会给定标记 $y_i, n+1 \leqslant i \leqslant n+m$, 即我们会知道标记的真实值 (groundtruth). 这样, 还可以对每个 $n+1 \leqslant i \leqslant n+m$, 通过比较 y_i 和 $\hat{y}_i$ 的差异来评估该映射有多准确.

如果我们学到的映射 f 的质量不太令人满意 (如准确率太低), 就需要提升其性能 (如通过设计或学习更好的特征, 或者通过学习更好的映射). 而当映射在测试中表现出令人满意的质量时, 就可以准备将它发送给用户了 (即在映射 f 的第一种无真实标记的应用场景中运行). 对学到的映射进行部署可以被视为任务的第三个阶段.

3.3 最近邻分类器

欧式距离

假设 $\boldsymbol{x} \in \mathbb{R}^d$ 和 $\boldsymbol{y} \in \mathbb{R}^d$ 是两个维度相同的向量，其欧几里德距离为

$$d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\| .$$

下面举例了一个用于人脸识别的简单最近邻算法

算法 3.1 一个用于人脸识别的简单最近邻算法

1: **输入：** 训练集 $(\boldsymbol{x}_i, y_i)$, $1 \leqslant i \leqslant n$. 第 i 个训练图像被转化为了向量 $\boldsymbol{x}_i$.

2: **输入：** 测试样本 $\boldsymbol{x}$, 由测试图像转化而来.

3: 在训练集中查找最近邻的下标.

$$nn = \arg\min_{1 \leqslant i \leqslant n} \|\boldsymbol{x} - \boldsymbol{x}_i\| . \tag{3.3}$$

4: **输出：** 返回预测结果

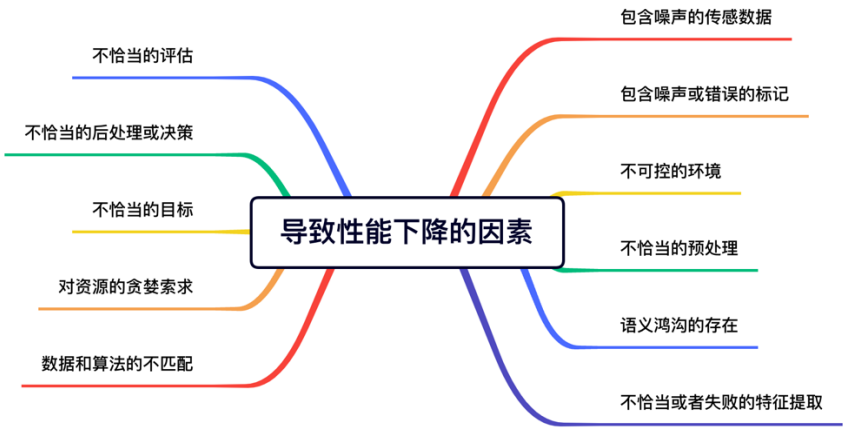
$$y_{nn} . \tag{3.4}$$

3.4 K-NN

基本思想：先计算分类样本与已知类别的训练样本之间的距离或者相似度，找到距离或者相似度与待分类样本数据的最近的 k 个邻居；在根据这些邻居所属的类别判断分类样本所属的类别。

3.5 丑陋的细节

由任何处理不当的细节所引起的精度下降可能要远远大于由不良学习算法所造成的性能下降。



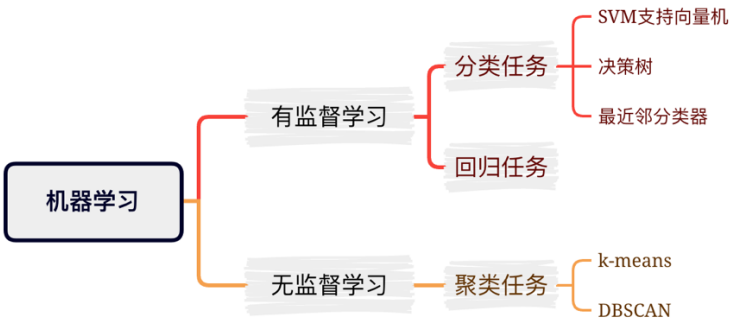
3.6 假设与简化

独立同分布假设

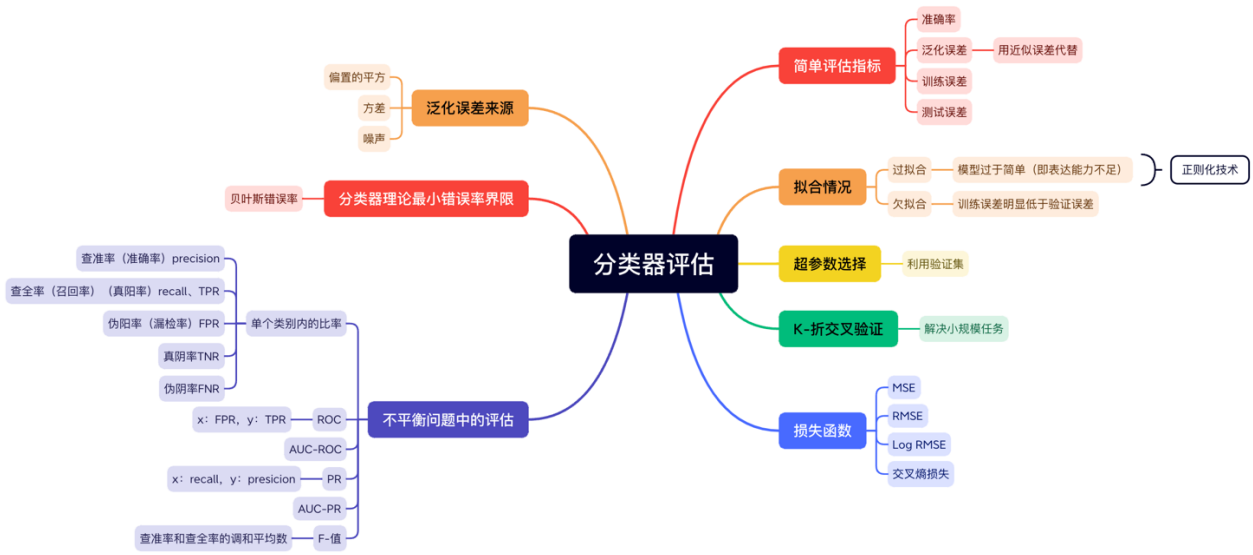
独立同分布 i.i.d.是假设训练和测试数据均来自一个完全相同的潜在分布 p(x,y);

独立同分布假设的另一个含义表明，潜在分布 p(x,y)不会发生改变。

遵循 i.i.d.假设会要求环境处于可控状态。



第四章　评估



4.1 简单情形的准确率和错误率

准确率(accuracy)是在分类任务中被广泛使用的评估指标。错误率(error rate)就是样本中被错误分类的百分比。

泛化误差(generalization error)是我们考虑服从潜在分布的所有可能样本时候的期望错误率。

$$Err = \mathbb{E}_{(x,y)\sim p(x,y)}[\llbracket f(x) \neq y \rrbracket]$$

近似误差(approximate error)，由于 iid 假设，我们可以用一个样本集合 D，来近似泛化误差。只要 D 服从 iid 假设从潜在分布中采样得到，就有：

$$Err \approx \frac{1}{|D|}\sum_{(x,y)\in D}\llbracket f(x) \neq y \rrbracket$$

4.2 训练与测试误差(training and test error)

$$Err \approx \frac{1}{|D_{train}|}\sum_{(x,y)\in D_{train}}\llbracket f(x) \neq y \rrbracket$$

$$Err \approx \frac{1}{|D_{test}|}\sum_{(x,y)\in D_{test}}\llbracket f(x) \neq y \rrbracket$$

因为训练误差不是泛化误差的可靠估计，因此我们需要用到测试集。

4.3 过拟合与欠拟合

如果模型不能降低训练误差，这可能意味着模型过于简单（即表达能力不足）， 无法捕获试图学习的模式。 此外，由于我们的训练和验证误差之间的泛化误差很小， 我们有理由相信可以用一个更复杂的模型降低训练误差。 这种现象被称为欠拟合（underfitting）。

另一方面，当我们的训练误差明显低于验证误差时要小心，这表明严重的过拟合（overfitting）。

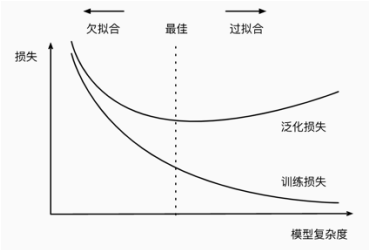


图4.4.1 模型复杂度对欠拟合和过拟合的影响

4.4 使用验证集来选择超参数

与可以在训练数据上学习到的参数不同，超参数通常是不可学习的。超参数通常与映射 f 的容量或者表示能力有关。

在学习方法中，超参数至关重要。错误的改变某个超参数的值可能会显著降低分类器的分类准确度。一种被广泛采用的方法是利用验证集，验证集、测试集、训练集彼此不相交。例如，对于一个分类器，我们可以获得多个验证误差，每个超参数对应一个误差，导致最小验证误差的那组超参数会被我们选中。

验证集也可以被用于检测模型是否发生过拟合，若模型发生了过拟合，则验证误差和测试错误率都会发生增大。

4.5 K-折交叉验证

当训练数据稀缺时，我们甚至可能无法提供足够的数据来构成一个合适的验证集。 这个问题的一个流行的解决方案是采用 K-折交叉验证。这里，原始训练数据被分成 K 个不重叠的子集。 然后执行 K 次模型训练和验证，每次在 K-1 个子集上进行训练，并在剩余的一个子集（在该轮中没有用于训练的子集）上进行验证。最后，通过对 K 次实验的结果取平均来估计训练和验证误差。

算法 4.1 错误率的交叉验证估计

1: **输入**: k, D_{train}, 一个学习算法 (已指定好超参数的值).
2: 将训练集随机地划分为 k 折, 使其满足公式 (4.6)— 公式 (4.8).
3: **FOR** $i = 1, 2, \ldots, k$ **DO**
4: 构建一个由 $D_1, \ldots, D_{i-1}, D_{i+1}, \ldots, D_k$ 中所有样本组成的新的训练集.
5: 使用该新训练集和学习算法学习一个分类器 f_i.
6: 计算 f_i 在 D_i 上的错误率, 记为 Err_i.
7: **END FOR**
8: **输出**: 返回如下由交叉验证估算的错误率

$$\text{Err}_{\text{CV}} = \frac{1}{k}\sum_{i=1}^{k}\text{Err}_i\,. \tag{4.9}$$

4.6 最小化代价/损失

基于上述对错误率的分析，我们将学习任务形式化为一个最小化错误率的优化问题。给定包含 n 个样本的训练集(x_i, y_i),(1<=i<=n)。映射 f 的训练错误率可以简单表示为:

$$\min \frac{1}{n}\sum_{i=1}^{n}[\![f(x_i) \neq y_i]\!]$$

由于指示函数是非线性的，难以优化，因此我们会使用新的对优化友好的损失函数来替代难以优化的损失函数。

我们使用如下的均方误差损失函数 MSE：

$$\min \frac{1}{n}\sum_{i=1}^{n}(f(x_i) \neq y_i)^2$$

4.7 正则化

正则化是处理过拟合的常用方法：在训练集的损失函数中加入惩罚项，以降低学习到的模型的复杂度。

令$R(f)$表示 f 上的一个正则项，MSE 最小化现在变为：

$$\min \frac{1}{n}\sum_{i=1}^{n}(f(x_i) \neq y_i)^2 + \lambda R(f)$$

其中超参数λ是一个权衡参数，若$\lambda = 0$，表示忽略正则化直接训练。若$\lambda > 0$,则表示将正则化代价添加进来。

4.8 代价矩阵

在很多情况下，简单的准确率或错误率可能效果不佳。大多数的不平衡的学习任务是代价敏感的：犯不同类型的错误所导致的损失也是不平衡的。

如下定义了一个 2*2 的代价矩阵

$$\begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix} = \begin{bmatrix} 0 & 10 \\ 1000 & 10 \end{bmatrix}$$

其中c_{ij}s 真实标记 y 为 i 且预测标记 f(x)为 j 的情况下产生的代价。

- 最小化错误率可以被视为代价最小化的特例，其代价矩阵为 $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.
- 但是，在很多实际应用中很难确定适当的 c_{ij} 值.
- 代价矩阵可以很容易地扩展到多类别问题. 在 m 类分类问题中，代价矩阵的大小为 $m \times m$, c_{ij} 表示真实标记为 i 却被预测为 j 的代价.

4.9 贝叶斯决策理论

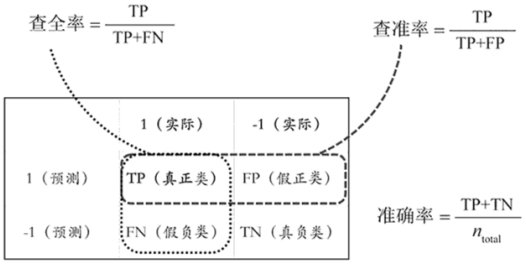
贝叶斯决策理论在概率意义上最小化代价函数。假设数据是一个随机向量，其联合概率密度为 Pr(x,y)，贝叶斯决策理论寻求最小化风险，它是关于联合密度的期望损失

$$\sum_{x,y} c_{y,f(x)} \Pr(x, y)$$

风险是在真实的潜在概率分布上的平均损失，可以写为：

$$\mathbb{E}_{(x,y)}[c_{y,f(x)}]$$

4.10 不平衡问题中的评估



样本总数:

$$TOTAL = TP + FN + FP + TN$$

正类样本总数:

$$P = TP + FN$$

负类样本总数:

$$N = FP + TN$$

准确率:

$$Acc = \frac{TP + TN}{TOTAL}$$

错误率:

$$Err = \frac{FP + FN}{TOTAL} = 1 - Acc$$

查准率:

$$查准率 = \frac{TP}{TP + FP}$$

真阳率（灵敏度）（查全率）:

$$TPR = \frac{TP}{TP + FN} = \frac{TP}{P}$$

伪阳率（漏检率）:

$$FPR = \frac{FP}{N}$$

真阴率:

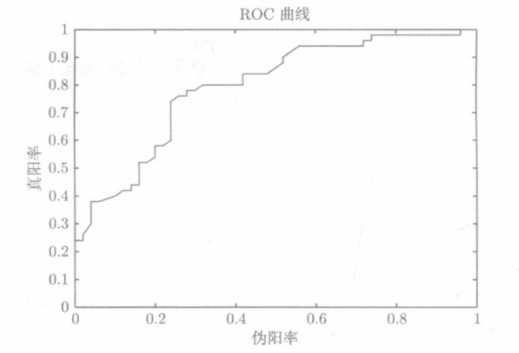
$$TNR = \frac{TN}{N}$$

伪阴率:

$$FNR = \frac{FN}{P}$$

4.11 受试者工作特征曲线 ROC

受试者工作特征曲线将 FPR 和 FNR 的变化过程记录到曲线中。X 轴为伪阳率，Y 轴为真阳率，TPR=1-FNR，且曲线是非减的。

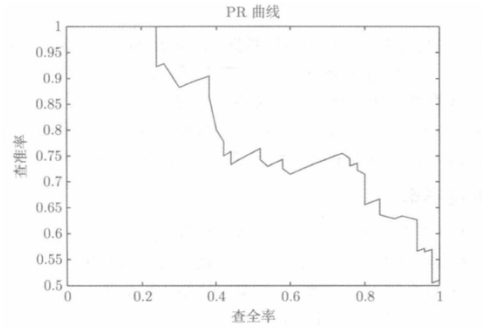


可以使用 ROC 曲线下的面积 AUC-ROC 来作为该评估的单值评估指标。

最好的 ROC 曲线应是从(0,0)到(0,1)的垂直线段和(0,1)到(1,1)的水平线段组成，对应于此的 ROC 曲线的分类器可以对所有的测试样本正确分类。它的 AUC-ROC 曲线是 1。

4.12 PR 曲线

与 ROC 曲线类似，我们可以调整分类器或检索系统的阈值来生成很多对（查准率、查全率），这些值形成了查准率-查全率曲线（PR 曲线）。然后我们可以用 AUC-PR 来作为评估标准。



PR 曲线和 ROC 曲线的一个区别是 PR 曲线不再是非递减的，AUC-PR 指标比 AUC-ROC 更具区分性。

4.13 F 值

$$F = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

F 值是查准率和查全率的调和平均数，它介于 0 和 1 之间。较高的 F 值表示更好的分类器。

$$F = \frac{2TP}{2TP + FP + FN}$$

F 值的扩展定义：

$$F\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{\beta^2 \cdot precision + recall}$$

根据β的取值可以决定查准率和查全率哪个指标更重要。

4.14 贝叶斯错误率

有 m 个类，第 i 个类表示为 y=i 。对于任意实例 x，我们将其真实标记置为：

$$y^* = \arg\max \Pr(x, y)$$

以使得不可避免的错误最小化。对应于 x，其错误为：

$$\sum_{y \neq y^*} \Pr(x, y)$$

如果我们对所有可能的实例$x \in \mathbb{x}$重复上述分析的话，则任意分类器的泛化误差都有一个下界，为

$$\sum_{x \in \mathbb{x}} \sum_{y \neq y^*(x)} \Pr(x, y) = 1 - \sum_{x \in \mathbb{x}} \Pr(x, y^*(x))$$

其中$y^*(x) = \arg\max \Pr(x, y)$是对于 x 使得 Pr（x,y）最大化的类别索引。上述公式定义了贝叶斯错误率，这是任意分类器可以达到的最小的错误率的理论极限。

4.15 偏置-方差分解

对于回归的偏置-方差分解为

$$\mathbb{E}[(y - f)^2] = (F - \mathbb{E}[f])^2 + \mathbb{E}[(f - \mathbb{E}[f])^2] + \sigma^2$$

这表明对于任意样本 x 的泛化误差可能来自三部分：偏置的平方、方差、噪声。

4.16 对评估结果的信心

影响置信度的一个因素是测试或验证集的大小。

为什么要取平均?

令 E 为与错误率对应的随机变量、如果$E1$，$E2$，…，Ek是 E 的 k 次采样，并且是在 i.i.d. 采样的训练和测试集上计算得到的（具有相同的训练和测试集大小），错误率通常被建模为正态分布，即$E \sim N(\mu, \sigma^2)$。很容易验证其平均值

$$\bar{E} = \frac{1}{k} \sum_{j=1}^{k} E_j \sim N(\mu, \frac{\sigma^2}{k})$$

也就是说，多个独立错误率的平均值会使方差减少到 k 分之一，这一事实意味着取平均可以减少错误率估计的方差。

为什么要报告样本标准差?

如果同时知道$\bar{E}$和σ，我们就可推断出关于$\bar{E}$的置信度。虽然我们不知道泛化误差μ，但通过$\bar{E}$和σ可以获得置信度较高 95%的估计值。也就是说我们知道μ位于:

$$[\bar{E} - \frac{2\sigma}{\sqrt{k}}, \bar{E} + \frac{2\sigma}{\sqrt{k}}]$$

但是实际中，我们不知道真实的总体标准差，它被 k 次划分所计算的样本标准差代替，但是，使用样本标准差后的分布具有封闭形式，即 t-分布。其次$\bar{E}$在公式中为一个随机变量。在实际中我们需要替换为再一次实验中从 k 个划分所计算的样本均值$\bar{e}$。

$$[\bar{e} - \frac{2\sigma}{\sqrt{k}}, \bar{e} + \frac{2\sigma}{\sqrt{k}}]$$

4.17 比较两个分类器

算法 4.2 用于比较两个分类器的配对 t-检验

输入: 两个分类器 f_1 与 f_2, 其在 n 个数据集上的错误率估计为 e_{ij} $(i\in\{1,2\}, 1\leqslant j\leqslant n)$.
选择显著性水平 α (常用值为 0.05 和 0.01).
找到临界值 $c_{n-1,\alpha/2}$.
对于 $1\leqslant j\leqslant n$, $x_j \leftarrow e_{1j} - e_{2j}$.
$\bar{x} \leftarrow \dfrac{1}{n}\sum_{j=1}^{n} x_j$, $s \leftarrow \sqrt{\dfrac{1}{n-1}\sum_{j=1}^{n}(x_j-\bar{x})^2}$.
$t \leftarrow \dfrac{\bar{x}}{s/\sqrt{n}}$.
IF $|t| > c_{n-1,\alpha/2}$ **THEN**
    我们认为 f_1 与 f_2 具有不同的错误率. (以 α 的显著性水平拒绝零假设.)
ELSE
    我们不认为 f_1 与 f_2 之间存在显著差异.
END IF