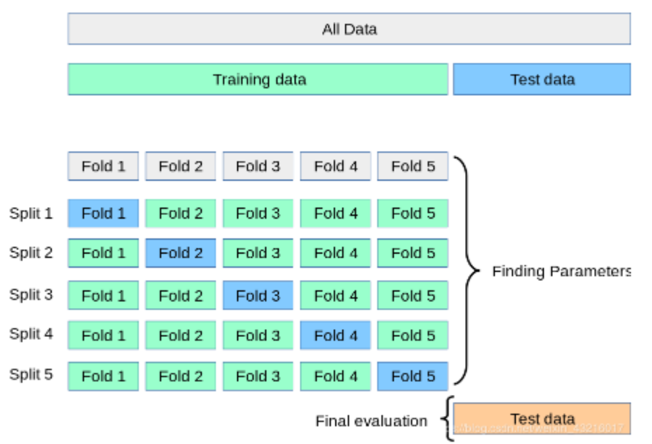
2022.3.9课后作业

统计学习导论第5题，第6题；对于数据Auto,

- (a)将数据集按照留出法分为训练集(比例0.8)和测试集(比例0.2)，对于函数$mpg = 40 - 0.15 \times horsepower$, 编写程序估计平方损失下的泛化误差,自行设定试验次数，进行泛化;
- (b)将数据集按照留p交叉验证的方式提取测试集，$K = 20$划分采取无放回抽样$p = 10$,对于函数$mpg = 40 - 0.15 \times horsepower$, 编写程序计算平方损失下的泛化误差;
- (c)建立一个二元变量，每加仑里程量（mpg01），1表示每加仑汽油该型号车所跑里程数（mpg）在3/4 分位数以上，0 表示每加仑汽油该型号车所跑里程数（mpg）在3/4分位数以下。将数据集按照留出法随机分为训练集(比例0.8)和测试集(比例0.2)，对于函数$P(mpg01 = 1) = \frac{\exp\{3.85-0.01\times weight\}}{1+\exp\{3.85-0.01\times weight\}}$ 编写程序估计平方损失下的泛化误差;
- (d)将数据集按照分层等比例分按照训练集(比例0.8)和测试集(比例0.2)，对于函数$P(mpg01 = 1) = \frac{\exp\{3.85-0.01\times weight\}}{1+\exp\{3.85-0.01\times weight\}}$ 编写程序估计平方损失下的泛化误差;
- (e)根据以上实验，结合教材，分析分层抽样和不同的抽样方式对泛化误差的影响;
- (f)书上的2.1.

调参和最终模型

- 不同的参数一般代表着不同的模型，特别是同一类的不同模型。一般参数分三种：
  1.一类是函数的参数，降低编程的难度，方便代码重用提高效率，比如，形参（左图中的参数表示颜色，线形）停止条件要给出估计的精度，参数传参时，可以通过位置参数传参或者关键字参数传参，位置可变参数可以接受多个可以接收多个实参，一个形参可以匹配任意个参数。
  2.一类是算法的参数，亦称“超参数”，如聚类所需要的簇数量k，人工选择一组候选。
  3.一类是模型的参数，如神经网络中每个节点的权重，回归中的每个变量的系数，让机器自己学习。
- 调参是机器学习的重点，也是决定模型性能的关键。一般调参过程中，会将训练数据再次划分为训练集和验证集（validation set）。具体包含关系如下：
  （数据集（训练数据（训练集）（验证集））（测试集））


All Data
Training data
Test data
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Split 1
Split 2
Split 3
Split 4
Split 5
Finding Parameters
Final evaluation
Test data

2.2 性能度量-回归问题和度量

- 回归问题的目标是在给定d维输入（input）变量x的情况下，预测一个或者多个连续目标（target）变量y 的值。
- 对于$D = \{(x_1, y_1), (x_N, y_N), \}$,要评估学习器$f$的性能;
- 回归中使用平方损失The regression loss-function: $L(y,f(x)) = (f(x) - y)^2$;
- 期望损失:

$$\mathbb{E}(L) = \int\int \{f(x) - y\}^2 p(x, y) \mathrm{d}x\mathrm{d}y$$

- 最小化$\mathbb{E}(L)$,得到：$2\int(f(x) - y)p(x, y)\mathrm{d}y = 0.$
- 求解出$f(x)$:

$$f(x) = \frac{\int yp(x, y)\mathrm{d}y}{p(x)} = \int yp(y|x)\mathrm{d}y = \mathbb{E}_y(y|x)$$

- 最优预测Solution: $f(x) = \int yp(y|x)\mathrm{d}t = \mathbb{E}_y[y|x]$;

- 等权错误率：

$$E(f,D)=\frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(f(x_i)\neq y_i);$$

- 等权精度：

$$ACC(f,D)=\frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(f(x_i)=y_i)=1-E(f,D);$$

- 一般错误率：

$$E(f,D)=\frac{1}{N}\int_{x\in D}\mathbb{I}(f(x\neq y))p(x)\mathrm{d}x;$$

- 一般精度：

$$ACC(f,D)=\frac{1}{N}\int_{x\in D}\mathbb{I}(f(x)=y)p(x)\mathrm{d}x;$$

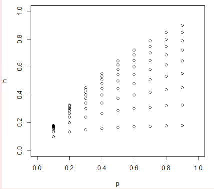
2.4 查准率和查全率

- 查准率P表示在所有被认为是正确的样例中，有多少个被正确分类的样例；
- 查全率R 表示在所有本身即是正确的样例中，有多少个被正确分类的样例。
- 对于二分类问题，分类结果表达为混淆矩阵（confusion matrix）

Table 2. Confusion Matrix.

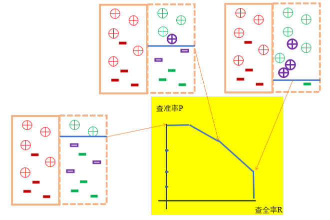
		预测结果	
		正例	负例
真实情况	正例	TP(真正例)	FN(假反例)
	负例	FP(假正例)	TN(真反例)

$$\text{precision} = \frac{TP}{TP+FP} \quad \text{recall} = \frac{TP}{TP+FN} \quad F1 = \frac{2\times\text{precision}\times\text{recall}}{\text{precision}+\text{recall}} \tag{1}$$



$P-R$曲线

分类器依次逐个将测试样例作为正例进行预测，则可以计算出每例的查准率和查全率。以查准率为纵轴y 轴、查全率为横轴x轴作图，就得到了查准率- 查全率曲线，简称“P-R曲线”.



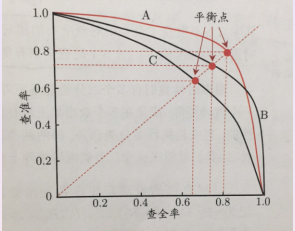


图 2.3 P-R曲线与平衡点示意图

- 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”，则可以断言后者的性能优于前者。例如上图中，学习器A的性能优于学习器C。
- 查准率和查全率是一对互为矛盾的度量，当查准率高时，查全率会偏低，当查全率高时，查准率会偏低，R 一定会是零吗？
- 平衡点(Break-Even Point,简称BEP)是一个用于比较学习器的性能度量，它是查准率= 查全率时的取值，如图中的BEP 是0.55。基于BEP值的高低可以比较出学习器的优劣。

$P-R$曲线2

$P(+)$大	$\longrightarrow$	$\longrightarrow$	$P(+)$小
+	+	-	-
+	+	+	-
+	-	+	-
+	+	-	-
P= 1,R= 0 FP=0	FP↑P↓ FN↓ R↑	FP↑P↓ FN=0，R↑	R=1

- 一些应用中，对查全率和查准率重视程度不一样，比如，推荐系统中，为更少打扰用户，被推荐的内容恰恰是用户感到有兴趣的，于是查准率很重要，而在治疗卵巢癌的手术时，为尽可能切除掉病灶，查全率更重要，根据查全率和查准率的不同偏好，使用F_β 来表达对查全率和查准率的不同偏好：

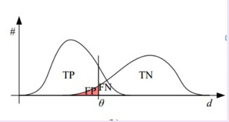
$$\frac{1}{\mathrm{F}_\beta}=\frac{1}{1+\beta^2}\times\left(\frac{1}{\mathrm{P}}+\frac{\beta^2}{\mathrm{R}}\right)$$

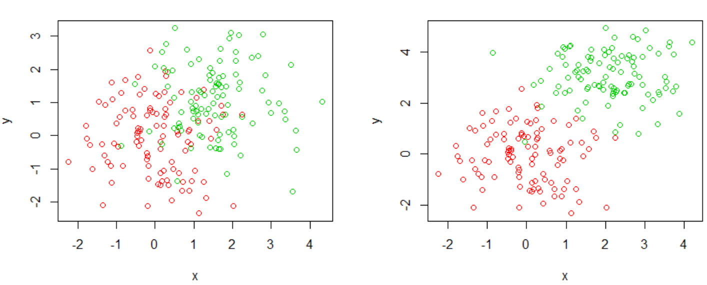
- 与算术平均$\frac{\mathrm{P}+\mathrm{R}}{2}$和几何平均$\sqrt{\mathrm{P}\times\mathrm{R}}$相比，调和平均更重视较小值。于是有：

$$\mathrm{F}_\beta=\frac{(1+\beta^2)\times\mathrm{P}\times\mathrm{R}}{\beta^2\times\mathrm{P}+\mathrm{R}}$$

思考题

$\beta > 1$，查全率有更大影响，$\beta < 1$,查准率有更大影响。$\beta = 1$，退化为F_1



- 对于图中的两个图，你认为如果用线性模型来划分两组数据，他们的P-R曲线会有怎样的不同？
- 

宏查准率、宏查准率、微查准率、微查全率

- 在前面的交叉验证法中对泛化误差的估计有很多，会有多次训练和测试，每次得到一个混淆矩阵，假设一共获得$(P_1, Q_1), (P_2, Q_2), \cdots, (P_K, Q_K)$ 个不同的查准率和查全率，
- 宏查全率，宏查准率:

$$\text{macro-P} = \frac{1}{K}\sum_{i=1}^{K} P_i; \quad \text{macro-R} = \frac{1}{K}\sum_{i=1}^{K} R_i;$$

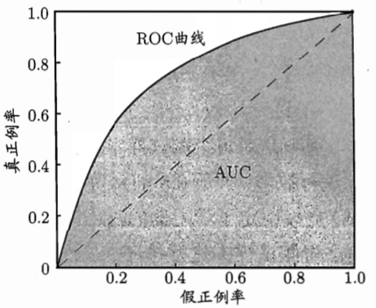
$$\text{macro-F1} = \frac{2 \times \text{macro-P} \times \text{macro-R}}{\text{macro-P} + \text{macro-R}}$$

- 微查全率，微查准率:

$$\text{micro-P} = \frac{\overline{TP}}{\overline{TP} + \overline{FP}}; \quad \text{micro-R} = \frac{\overline{TP}}{\overline{TP} + \overline{FN}};$$

$$\text{micro-R1} = \frac{2 \times \text{micro-R} \times \text{micro-R}}{\text{micro-P} + \text{micro-R}}$$

ROC(Receiver Operating Characteristic)与AUC



(a) ROC 曲线与 AUC

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

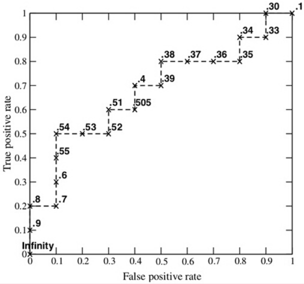
- Equal Error rate

如何绘制ROC与AUC

- 假设已经得出一系列样本被划分为正类的概率，然后按照大小排序，下表是一个示例，表中共有20 个测试样本，"Class"一栏表示每个测试样本真正的标签（*p*表示正样本，*n*表示负样本），"Score"表示每个测试样本属于正样本的概率。

Inst#	Class	Score	Inst#	Class	Score
1	**p**	.9	11	**p**	.4
2	**p**	.8	12	**n**	.39
3	**n**	.7	13	**p**	.38
4	**p**	.6	14	**n**	.37
5	**p**	.55	15	**n**	.36
6	**p**	.54	16	**n**	.35
7	**n**	.53	17	**p**	.34
8	**n**	.52	18	**n**	.33
9	**p**	.51	19	**p**	.30
10	**n**	.505	20	**n**	.1

- 接下来，从高到低，依次将“Score”值作为阈值threshold，当测试样本属于正样本的概率大于或等于这个threshold 时，我们认为它是正样本，否则为负样本。举例来说，对于图中的第4 个样本，其“Score”值为0.6，那么样本1，2，3，4都被认为是正样本，因为它们的“Score”值都大于等于0.6，而其他样本则都认为是负样本。每次选取一个不同的threshold，我们就可以得到一组FPR 和TPR，即ROC 曲线上的一点。这样一来，我们一共得到了20 组FPR和TPR的值，将它们画在ROC曲线的结果如下图：



- 设当前点为(x,y),当测试样本属于正样本时，下一个点是$(x,y+\frac{1}{N^+})$, 否则就是$(x+\frac{1}{N^-},y)$

ROC与AUC

- AUC表示ROC曲线下的面积，用于比较学习器的性能优劣，一般面积更大的学习器的性能更为优良。AUC 可估算为

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

- 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N^+ 个正例和N^- 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

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

  它刻画了正例得分没有高出负例得分的那部分损失，l_{rank}对应的是ROC曲线以上的面积

- $\mathrm{AUC} = 1 - l_{rank}$;

ROC曲线的基本性质1/2

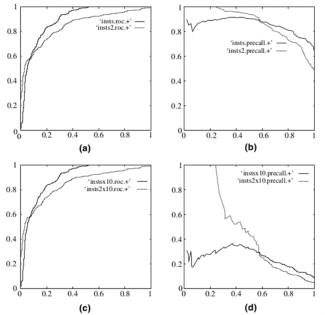
- 性质1：任何ROC曲线必定经过原点和(1,1)
  证明：当阈值大于所有样本的预测值时，所有的样本都会被归为负类，这时正例数为0，因此FPR和TPR都为0，对应于ROC 曲线的原点，此时的分类方法是最“保守的”。当阈值小于所有样本的预测值时，所有的样本都会被归为正类，此时负例数为0，因此$TN = FN = 0$ 进而FPR和TPR都为1，对应于ROC曲线的(1,1)点。
- 性质2：位于上方的ROC曲线分类效果好于下方的ROC曲线
  证明：在给定样本分布的情况下，假设有两个分类器A和B，A的ROC曲线高于另一个分类器B的ROC曲线，说明在相同的假正例率的条件下，A 的真正例率大于B，因此分类器A要优于分类器B。

ROC曲线的基本性质2/2

- 性质3：在样本有限的情况下，ROC曲线为由水平线和竖直线构成的折线
  可以将所有样本的预测值由大到小排序。然后把分类器的阈值设为最大，即所有样本都被归为反例，这时对应于ROC曲线的原点。然后，从高到低依次将阈值设为每个样本的预测值，即依次把每个样本划分为正例。设前一个标记点的坐标为(x, y)，若当前为真正例，则对应标记点的坐标为$(x, y+1/m^+)$, 若当前为假正例(负例)，则对应标记点的坐标为$(x+1/m^-, y)$, m^+, m^-分别表示样本中正例和反例的总数,这就证明了该性质。
- 性质4：$y=x$曲线对应于随机猜测分类器在无穷多样本下的极限
  证明：对于随机猜测的情况，被归为正例的样本中有一半是猜对的有一半是猜错的，被归为负例的样本也一样。假设总共有M个样本归为正类，N个样本归为负类，
  则$TN=FN=N/2, TP=FP=M/2,$，因此$TPR=FPR=\frac{M}{M+N}$，
  这对应于$y=x$ 曲线。
- 性质5：任何一个有效的分类器都位于$y=x$上方
  证明：由于任何有效的分类器都优于随机分类器，由性质2和性质3可直接得到性质4。

ROC曲线的好特性

- 性质6：ROC曲线对样本的分布不敏感 当测试集中的正负样本的分布不对等的时候，ROC曲线能够保持不变。在实际的数据集中经常会出现样本类不平衡，即正负样本比例差距较大，而且测试数据中的正负样本也可能随着时间变化。下图是ROC 曲线和Presision-Recall 曲线的对比：



- 在上图中，(a)和(c)为Roc曲线，(b)和(d)为Precision-Recall曲线。(a)和(b)展示的是分类其在原始测试集(正负样本分布平衡)的结果，(c)(d) 是将测试集中负样本的数量增加到原来的10倍后，分类器的结果，可以明显的看出，ROC曲线基本保持原貌，

		预测结果	
		正例0	负例1
真实情况	正例0	0	$Cost_{01}$
	负例1	$Cost_{10}$	0

$Cost_{10} = C(+|-)$表示实际为反例但预测成正例的代价;$Cost_{01} = C(-|+)$表示实际为正例但是预测为反例的代价

- $$\mathbb{E}(f, D_i, cost) = \frac{1}{N}\left(\sum_{x_i \in D^+} \mathbb{I}(f(x_i) \neq y_i) cost_{01} + \sum_{x_i \in D^-} \mathbb{I}(f(x_i) \neq y_i) cost_{10}\right)$$
- 非均衡代价下，ROC曲线不能直接反映出学习器的期望总代价，于是出现了一个新的曲线“代价曲线”
- 分析理解：

$$Cost_{norm} = \frac{Cost_{10} * (1-p) * FPR + Cost_{01} * p * FNR}{Cost_{10} * (1-p) + Cost_{01} * p}$$

这里$E(Cost) = Cost_{10} * P(Cost = 1|负例)P(负例) + Cost_{01} * P(Cost = 0|正例)P(正例)$

代价敏感错误率与代价曲线2/2

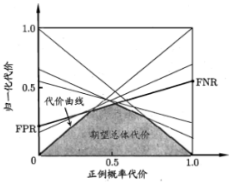
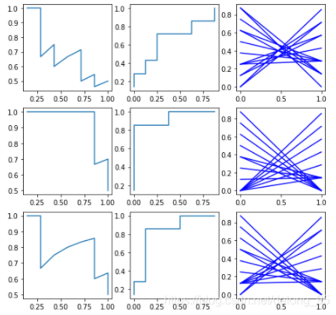


图 2.5 代价曲线与期望总体代价

- 期望总体代价越小，则模型的泛化能力越强；
- 与ROC曲线的区别与练习：ROC主要考量均等代价，代价敏感曲线主要考量非均等代价。两者都是衡量某一学习器在不同场景下的综合表现情况，而不是单一场景。ROC通过阈值变化来体现不同场景，即高阈值表现了重视查准率的场景，低阈值则重视查全率的场景。代价敏感曲线则是通过P值，即正例的先验概率即原本正例占比的变化来体现不同场景。代价敏感曲线上方直线是根据不同决策阈值下做出的（含有参数P，固定参数），横轴P值确定时，即确定了一种情景，直线对应的点体现了P这种情景下在该阈值下的代价。因此当画出不同阈值下的直线时，某一P总能找到最小代价进而找出对应的决策阈值。这样每一个P都能对应一个在学习器下的最下代价，进而下方面积就是概率和代价的积分则为期望：该学习器的整体期望或整体表现。

如下9幅图为3个模型对同一个样本的分类结果。第1行为$model_1$的$P-R$图、ROC曲线、代价曲线。第2行为$model_2$，第3行为$model_3$。3 个$mode$对比,判断哪个模型比较理想?



作业

1. （03-21）请根据例题中的测试得分绘制ROC曲线，$P-R$曲线
2. （03-29）请根据例题中的测试得分绘制均衡代价曲线(注：$cost_{10}=1, cost_{01}=1$)。如果该数据是一个银行追查网络钓鱼攻击的一个学习器的训练数据，漏网的代价(FN)是误报代价(FP)的5倍（$cost_{10}=1, cost_{01}=5$）那么请绘制代价曲线，如果是3倍，代价曲线会怎样变化，请比较三种不同代价下的代价曲线有什么变化？
3. （03-29）如下数据是16次实验中的误分类结果，请比较测试误差率是否小于30%($\epsilon\leq 30\%$)，实验中每次测试集$n=10$的误分类数量(mc) 如下，凭借这些数据使用t检验对改算法的泛化误差给出判断？请给出检验问题的表述，计算统计量，计算统计量的$p-$值，给出结论。

3	4	1	2	0	1	2	3
1	2	4	0	0	3	4	5

比较检验（比较什么？）

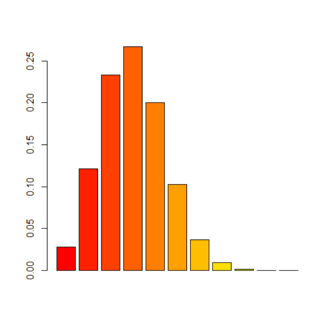
- 比较的是学习器的泛化性能，但性能度量观测（$\epsilon_1, \cdots, \epsilon_K$）可能和泛化性能（$\epsilon$）并不相同；
- 测试集上的性能与测试集本身的选择有很大关系，测试集不同，结果可能也会发生改变，场景不同结果可能也不同;
- 很多机器学习算法具有很大的随机性，同一份测试集多次测试结果也可能不同(假设参数相同)；统计假设检验为进行学习算法的性能比较提供了重要依据。基于假设检验的结果可以推断出:若在测试集上观察到学习算法L_A比L_B好，则L_A 的泛化性能是否在统计意义上优于L_B，以及这个结论的把握有多大。
- 假设是对学习器泛化错误率分布的某种判断或猜想，用测试错误率估计泛化错误率，以检查学习器性能。

问题1 当问起一个概念学习算法的泛化误差，多数人的回答是$\epsilon \leq 30\%$，如果这在行业中成为一种共识。一项研究中16次算法实验中每次测试集$n = 10$的误分类数量(mc) 如下，凭借这些数据可以对这个泛化误差给出怎样的判断？

3	4	1	2	0	1	2	3
1	2	4	0	0	3	4	5

问题1 当问起一个概念学习算法的泛化误差，多数人的回答是$\epsilon \leq 30\%$，如果这在行业中成为一种共识。一项研究中16次算法实验中每次测试集$n = 10$的误分类数量(mc) 如下，凭借这些数据可以对这个泛化误差给出怎样的判断？

3	4	1	2	0	1	2	3
1	2	4	0	0	3	4	5



- 检验分析：假设实验次数为K,每一次试验会产生一个泛化误差的估计$\hat{\epsilon}_k$,这个$\hat{\epsilon}_k, k = 1, \cdots, K$ 可视作ϵ 的一次估计，泛化误差为ϵ在一次测试中产生$\hat{\epsilon}$的可能性$P(\hat{\epsilon}_k|\epsilon) = dbinom(mc|n_k, \epsilon) = \binom{n_k}{mc}\epsilon^{mc_k}(1-\epsilon)^{n_k - mc_k}$,
这里$\hat{\epsilon}_k = \frac{mc_k}{n_k}$
- 检验过程：给出一个待比较的泛化误差ϵ_0,可信度$\alpha > 0$,取$k^* = \text{argmax}\hat{\epsilon}_k$
- 找出最大的错误率和错误率所在的那组实验，对这组错误率实施以下检验,
- 设立假设: $H_0 : \epsilon \le \epsilon_0 \quad H_1 : \epsilon > \epsilon_0$;
- 计算概率

$$\mathbb{P}(mc > mc_{k^*}|\epsilon_0) = \sum_{i=mc_{k^*}+1}^{n_{k^*}} \binom{n_{k^*}}{i} \epsilon_0^i (1-\epsilon_0)^{n-i}$$

- 如果$\mathbb{P}(mc > mc_{k^*}|\epsilon_0, n_k) < \alpha$拒绝零假设，可以在$1-\alpha$的显著性下认为学习器的泛化误差是大于$\epsilon_0$ 的。
- 【例题】本例中，错误率最高的那一类是误分类数达到5的那一组，计算$1 - pbinom(5, 10, 0.3) = 0.047$, 如果取$\alpha = 0.05$, 可以认为拒绝零假设，认为整个泛化误差是超过0.30 的。

多次留出法以及交叉验证中的t检验

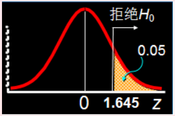
- 设立假设: $H_0 : \epsilon \leq \epsilon_0 \quad H_1 : \epsilon > \epsilon_0$;
- （K次）训练/测试，会产生多个测试错误率$\epsilon_1, \cdots, \epsilon_K$,假设要比较的泛化误差率是$\hat{\mu}$, 假设样本方差$\hat{\sigma}^2$ 如下：

$$\hat{\mu}_\epsilon = \frac{1}{K}\sum_{k=1}^{K}\epsilon_k \quad \hat{\sigma}^2 = \frac{1}{K-1}\sum_{k=1}^{K}(\epsilon_k - \hat{\mu}_\epsilon)^2.$$

- 将K个测试误差看作泛化误差率ϵ_0的独立采样，定义检验统计量

$$T = \frac{\sqrt{K}(\hat{\mu}_\epsilon - \epsilon_0)}{\hat{\sigma}}.$$

- T服从自由度为$K-1$的t分布，检验准则为：p值为$P(T > t_\alpha(K-1)) < \alpha$时拒绝零假设，认为测试误差率$> \epsilon_0$;



交叉验证$t-$检验

- 基本原理：对两个学习器A和B，使用k折交叉验证法分别得到k个测试错误率，如果两个学习器性能相同，则使用相同训练/测试集时测试错误率应该相同.设A的误差率$\epsilon_1^A, \cdots, \epsilon_k^A$，设$B$ 的误差率$\epsilon_1^B, \cdots, \epsilon_k^B$，
- 设立假设: $H_0 : \epsilon^A = \epsilon^B \quad H_1 : \epsilon^A \neq \epsilon^B$;
- 首先求两个学习器的k个测试错误率的差，即$\Delta_k = \epsilon_k^A - \epsilon_k^B$，若

$$T_\Delta = |\frac{\sqrt{k} \times \mu_\Delta}{\sigma_\Delta}| < t_{\alpha/2,k-1}$$

,则认为两个学习器性能相同,否则，认为两个学习器性能存在显著性差别，而且学习错误率较小的那个学习器的性能较优。

-

5×2交叉验证$t-$检验

- 要进行有效的t检验，一个重要前提是测试误差率均为泛化误差的独立采样，然而由于样本有限，使用交叉验证等实验估计方法时，不同轮次的训练样本会有一定程序的重叠，这就使得测试误差率实际上并不独立，而且会导致过高估计假设成立的概率
- 解决方案：采用5×2交叉验证法[Dieterich1998],每次2折交叉验证之前用随机数将数据打乱，使得5 次交叉验证的数据划分不重复，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，分别求差，记为Δ_k^1, Δ_k^2, 为缓解测试错误率的非独立性，仅计算第1次2 折交叉验证的两个结果的平均值$\mu = 0.5(\Delta_i^1 + \Delta_i^2)$,但是对每次2折实验的结果都计算方差$\sigma_i^2 = (\Delta_i^1 - \mu) + (\Delta_2^1 - \mu)$. 用统计量
- $T = \frac{\mu}{\sqrt{0.2 \sum_{i=1}^{5} \sigma_i^2}}$服从自由度5的$t$分布，取双边检验临界值$t_{\alpha/2,5} = 2.5706(\alpha = 0.05)$.进行拒绝性检验Machine Learning. In David Hemmendinger, Anthony Ralston and Edwin Reilly (Eds.), The Encyclopedia of Computer Science, Fourth Edition, Thomson Computer Press.

McNemar检验

Table 1.两学习器分类差别列联表

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

- 若假设A,B学习器性能相同，则应由$e_{01}=e_{10}$，那么$|e_{01}-e_{10}|$ 应服从正态分布。McNemar 检验考虑变量$\chi^2=\frac{(|e_{01}-e_{10}|)^2-1}{e_{10}+e_{01}}$.
和$\chi^2_{0.975}(1)=pchisq(0.975,1)=5.023$ 进行比较，大于它拒绝。
- 【例题】A和B两种算法对同一组数据集进行测试，A算法判出91个正例(65%)，B算法判出77 名正例(55%)，A和B 两法一致的正例56 名(40%)，问哪种方法学习性能更高？

		算法 A 正确	算法 A 错误	合计
算法 B	正确	56	35	91
	错误	21	28	49

- $1-pchisq(10.73,1)=0.0010$p值很小，于是拒绝零假设，认为两个算法性能有差异。

Friedman检验和Nemenyi检验

- 有多个数据集多个学习器进行比较时使用，对各个算法在各个数据集上对测试性能排序，对平均序值计算χ^2和F 检验,并进行临界值检验。
- 假定有B个数据集上比较K个算法，令R_j表示第j个算法在D个数据集的秩和（这里忽略秩打结的情况），根据秩方差分析原理（非参数统计，王星2014），构造统计量

$$Q_{\chi^2} = \frac{12}{BK(K+1)} \sum_{k=1}^{K} R_j^2 - 3B(K+1)$$

当$K \times B$比较大的时候，上述Q统计量服从自由度为$K-1$的χ^2分布，可以分析是不是几个算法性能有差异。

在一组数据集上对多个算法进行比较

表 2.5 算法比较序值表

数据集	算法 A	算法 B	算法 C
D_1	1	2	3
D_2	1	2.5	2.5
D_3	1	2	3
D_4	1	2	3
平均序值	1	2.125	2.875

Nemanyi后续检验

- ```
  > alcp=c(1,1,1,1,2,2.5,2,2,3,2.5,3,3)
  > group=c(1,2,3,4,1,2,3,4,1,2,3,4)
  > treat=c(1,1,1,1,2,2,2,2,3,3,3,3)
  > friedman.test(alcp,treat,group)

          Friedman rank sum test

  data:  alcp, treat and group
  Friedman chi-squared = 7.6, df = 2, p-value = 0.02237
  ```

- Friedman检验的后续分析1F检验：当要比较的组数很多的时候（K比较大）

$$F = \frac{(B-1)Q_{\chi^2}}{B(K-1)-Q_{\chi^2}},$$

F服从自由度为$(K-1)(B-1)$的F分布

- 【例题】B=4,K=3, $F=(B-1)*7.6/(B*(K-1)-7.6)=57$,显著大于临界值$F_{0.05}((K-1),(K-1)(B-1))=qf(0.95,2,2*3)=$ [1]5.143253,于是拒绝零假设，认为几组学习器之间的测试误差性能具有很大差异。
  ```

Nemanyi后续检验

- 当Friedman检验发现多个学习器之间性能存在差异时，想知道哪些算法性能之间差异较大，需要进行平均秩之差的Nemanyi 后续检验。
- $$CD = q_{\alpha}\sqrt{\frac{K(K+1)}{6B}}$$
- 【例题】$CD = 2.344 * sqrt(K * (K+1)/(6 * B)), CD = 1.657$
- 比较各组平均秩$\Delta_{1,2} = 1.125 < CD, \Delta_{2,3} = 0.75, \Delta_{1,3} = 1.775 > CD$
- 于是，有关各组算法之间性能的结论是：在显著性水平$\alpha = 0.05$之下，算法A与算法C 的差异是比较大的，其他算法之间没有发现明显差异。

偏差与方差(1)

$$
\begin{aligned}
E(f;D) =& \mathbb{E}[(f(x;D) - y_D)^2] \\
=& \mathbb{E}_D\left[(f(x;D) - \mathbb{E}(y|x) + \mathbb{E}(y|x) - y_D)^2\right] \\
=& \mathbb{E}_D\left[(f(x;D) - \mathbb{E}(y|x)\right]^2 + \mathbb{E}\left[\mathbb{E}(y|x) - y_D)^2\right] \\
& + \mathbb{E}_D[2(f(x;D) - \mathbb{E}(y|x))(\mathbb{E}(y|x) - y_D)] \\
=& \mathbb{E}_D\left[(f(x;D) - \mathbb{E}(y|x)\right]^2 + \mathbb{E}\left[\mathbb{E}(y|x) - y_D)^2\right] \\
=& \mathbb{E}_D\left[(f(x;D) - \mathbb{E}(y|x)\right]^2 + \mathbb{E}\left[\mathbb{E}(y|x) - y + y - y_D)^2\right] \\
=& \mathbb{E}_D\left[(f(x;D) - \mathbb{E}(y|x)\right]^2 + \mathbb{E}\left[(\mathbb{E}(y|x) - y)^2\right] + \mathbb{E}\left[(y - y_D)^2\right] \\
& + 2\mathbb{E}_D\left[(\mathbb{E}(y|x) - y)(y - y_D)\right] \\
=& \mathbb{E}_D\left[(f(x;D) - \mathbb{E}(y|x)\right]^2 + \left[(\mathbb{E}(y|x) - y)^2\right] + \mathbb{E}_D\left[(y - y_D)^2\right]
\end{aligned}
$$

- 于是

$$E(f;D) = bias^2(x) + var(x) + \epsilon^2$$

- 泛化误差分解为偏差、方差与噪声之和。

偏差与方差(2)

- 偏差-方差分解是解释算法泛化性能的一种重要工具。偏差-方差分解试图对学习算法的期望泛化错误率进行拆解。
- 泛化误差可分解为：偏差，方差与噪声之和。
- 偏差度量了学习算法的期望预测与真实结果的偏离程度，即刻画了学习算法本身的拟合能力。
- 方差度量了同样大小的训练集的变动所导致的学习性能的变化，即刻画了数据扰动所造成的影响。
- 噪声表达了在当前任务上任何学习算法所能达到的期望泛化误差的下界,即刻画了学习问题本身的难度
- 泛化性能是由学习算法的能力，数据的充分性以及学习任务本身的难度所共同决定的。

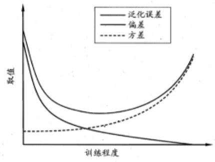


图 2.9 泛化误差与偏差、方差的关系示意图

第二章总结

- 本章主要基于NFL，学习器性能很难判断，经验误差无法代表泛化误差进行训练，否则难免不出现过度拟合；
- 需要合适的评估方法来给出泛化误差的近似值，划分训练、测试集，估计出不同的泛化估计；
- 用哪些性能指标来计算测试误差（错误率、查准率，查全率，F1,ROC,AUC,代价敏感曲线）；
- 计算完误差以后如何比较不同学习器的性能?
- 比较完性能后如何理解性能上的差异。

3.29课后作业

- 在一个钓鱼问题中（数据fishing.xls），感兴趣的是具备怎样的条件可以钓到鱼，我们的数据分成训练集和测试集（由标签变量定义），在训练集上建立了两个模型，分别获得了测试集和训练集的预测评分($E(Y|X)$ 的估计)，要求编写程序，实现以下分析内容，做出数据分析报告：
  - 请对训练数据和测试数据分别对模型1和模型2制作P-R 图，ROC 曲线图，用经验AUC 估算面积，判断整体预测效果，如果按照实验结果给出一个符合实际的预测阈值，请说明这个阈值选取的依据；
  - 在给出的阈值上，输出混淆矩阵。
  - 如果将钓不到鱼的游客判为能钓到鱼，这个损失将会使钓鱼中心的声誉受损更大，将其设为将能钓到鱼的游客判为钓不到鱼的损失的2倍，请根据新的非平衡代价损失制作两个模型的测试数据的代价曲线，并给出比较分析结果。
- 在一个汉字识别问题中（数据手写体数字数据.xls），我们用5折交叉验证法将数据分成5 折训练集和测试集，获得了3个0-1分类模型
  - 编写程序计算宏、微查全率和查准率，根据经验选择合适的阈值。
  - 编写程序求解宏期望代价曲线($Macro_{FPR}, Macro_{FNR}$)，微期望代价曲线($Micro_{FPR}, Micro_{FNR}$)并画图表示,比较三种模型的代价曲线。

3.29课后作业1

- 在一个图像去噪研究中，作者研究了四种算法$K-SVD$,$K-SVD-N$,NLM和$K-SVD-N-NL$的实验效果，分别在5 组测试图像上获得信噪比评分PSNR,该信噪比数值越高表示方法越有效，请完成以下检验：

表 1　去噪图像的 PSNR 值比较　dB

测试图像	算法	$\sigma=20$	$\sigma=30$	$\sigma=40$	$\sigma=50$	$\sigma=60$	$\sigma=70$	$\sigma=80$	$\sigma=90$	$\sigma=100$
Lena	K-SVD	30.42	27.77	25.66	23.94	22.51	21.28	20.21	19.25	18.40
	K-SVD-N	30.69	28.33	26.48	25.16	23.97	23.05	22.17	21.30	20.43
	NLM	30.61	28.56	27.03	25.72	24.60	23.65	22.85	22.19	21.62
	K-SVD-N-NL	**31.57**	**29.96**	**28.57**	**27.51**	**26.75**	**25.91**	**25.47**	**24.92**	**24.36**
Barbara	K-SVD	29.25	27.01	25.04	23.34	21.90	20.70	19.67	18.76	17.96
	K-SVD-N	29.42	27.10	25.24	23.72	22.45	21.48	20.63	19.97	19.33
	NLM	29.26	26.98	24.76	23.43	22.39	21.56	20.89	20.34	19.88
	K-SVD-N-NL	**29.74**	**28.00**	**26.61**	**25.37**	**24.40**	**23.61**	**23.00**	**22.47**	**22.04**
Boat	K-SVD	29.09	26.89	25.00	23.40	22.04	20.88	19.85	18.95	18.13
	K-SVD-N	29.37	27.22	25.62	24.23	23.03	22.12	21.19	20.59	19.66
	NLM	28.50	26.41	24.98	23.93	23.08	22.36	21.75	21.21	20.75
	K-SVD-N-NL	29.64	28.01	26.84	25.79	25.01	24.36	23.79	23.30	22.82
Cameraman	K-SVD	26.96	25.82	24.39	22.98	21.70	20.52	19.50	18.59	17.75
	K-SVD-N	27.52	26.20	24.74	23.45	22.29	21.34	20.46	19.70	19.15
	NLM	**28.75**	26.56	24.84	23.56	22.57	21.75	21.05	20.43	19.89
	K-SVD-N-NL	27.87	**26.83**	**25.86**	**24.99**	**24.15**	**23.38**	**22.74**	**22.14**	**21.61**
Peppers256	K-SVD	29.18	26.94	25.01	23.34	21.92	20.71	19.59	18.64	17.79
	K-SVD-N	29.49	27.33	25.57	24.03	22.93	21.98	21.01	20.28	19.48
	NLM	28.95	26.57	24.74	23.19	21.89	20.84	20.01	19.33	18.78
	K-SVD-N-NL	**29.76**	**28.09**	**26.74**	**25.58**	**24.56**	**23.87**	**23.08**	**22.46**	**21.85**

  - 在Cameraman数据上，分别比较四种算法，请问哪一种算法的信噪比高于20%。
  - 在$\sigma=20$上，四种算法是否有显著差异？