模型评估与选择

-----BY 2020.2 YANGZHONGXIU

内容

- 经验误差与过拟合
- 评估方法
- 性能度量
- 比较检验
- 偏差与方差

经验误差与过拟合

误差的几个概念：

错误率(error rate)：分类错误的样本数占样本总数的比例称为“错误率”。假设在m个样本中有a个样本分类错误，则错误率E=a/m；

精度(accuracy):1-错误率=精度；

误差(error)：机器学习的实际预测与样本的真实输出之间的差异称为“误差”。

训练误差(training error)/经验误差(empirical)：学习器在训练集上的误差成为“训练误差”。又称为“经验误差”。

泛化误差(generalization error):在新样本上的误差成为“泛化误差”。

我们最希望哪个误差最小？

经验误差与过拟合

过拟合(over fitting)：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提点当作了所有潜在的样本都会具有的一般性质，这样就会导致泛化能力下降，这种现象在机器学习中称为“过拟合”。

欠拟合(underfitting)：与过拟合相对应，指对训练样本的一般性质尚未学好，没有把样本的一半性质学完全，导致模型学习器正确率不高。

过拟合、欠拟合的直观类比

经验误差与过拟合

对于过拟合和欠拟合的几个认识：

- 导致过拟合(over fitting)原因是学习能力过于强大；导致欠拟合(underfitting)原因学习能力低下。
- 欠拟合可以通过加强学习能力来克服，过拟合无法彻底避免。

机器学习一直在与“过拟合”作斗争

评估方法

训练集与测试集划分方法：

通过实验测试来对学习器的泛化误差进行评估并进而做出选择。因此，需要使用“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。

如果要使得“测试误差”尽可贴近真实，测试样本尽量不要在训练集中出现，没有在训练过程中使用过。

假设有m个样例的数据集 D={ (x1,y1), (x2,y2),…, (xm,ym)}，如何产生测试集T与训练集S。

评估方法

- **留出法**

直接将数据集D划分为两个互斥的集合，其中一个作为训练集S，另一个作为测试集T。在S集上训练出模型后，用T来评估其测试误差，作为对泛化误差的估计。

问题：

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倒入额外的偏差而对最终结果产生影响。

解决方式：

即便在给定训练/测试集的样本比例后，仍存在多种方法对初始集D进行分割。一般要采用若干次随机划分，每次产生一个训练/测试集。留出法就是对多次得到的结果去平均值。

留出法中，一般将数据2/3~4/5的样本用于训练，其他用于测试。

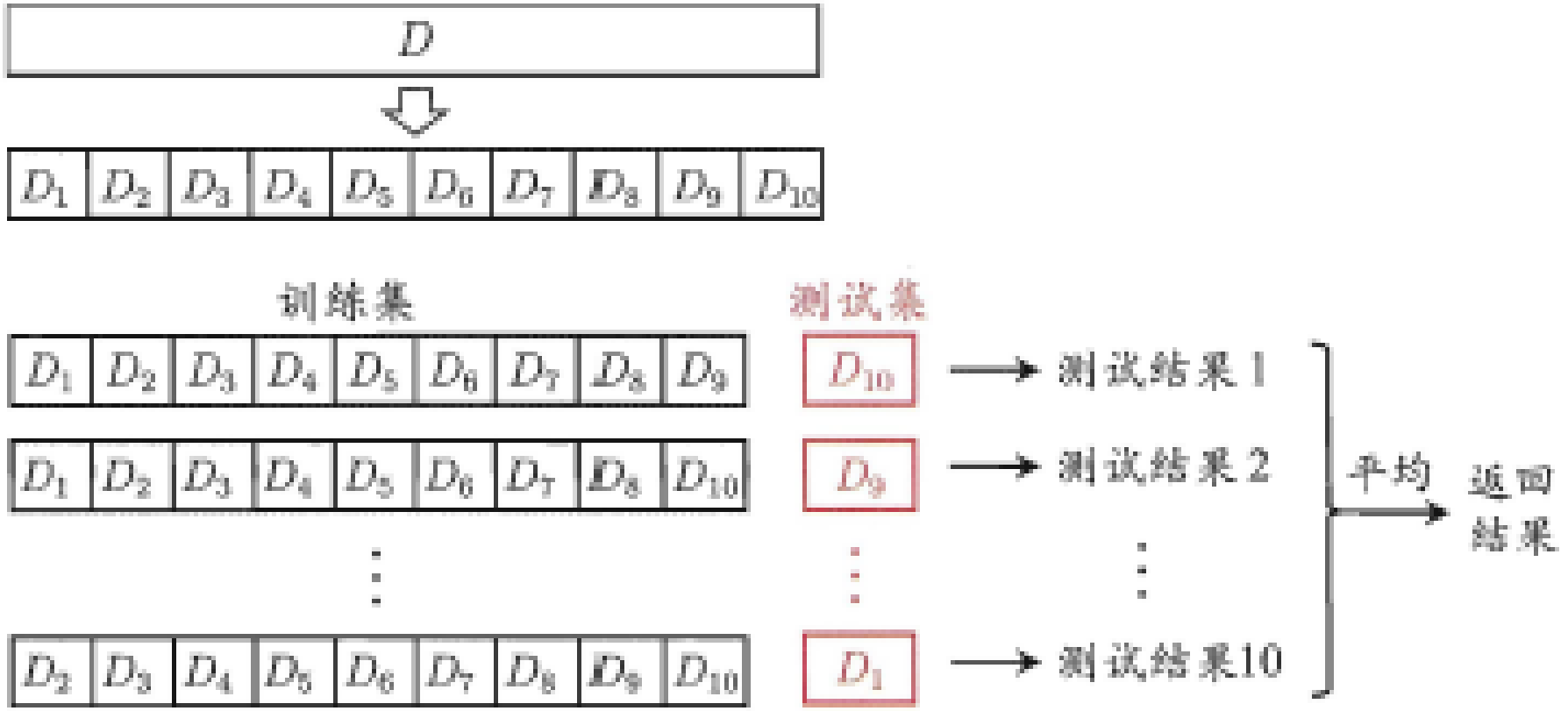
评估方法

- **交叉验证法（cross validation）**

先将数据集D划分为K个不同大小相似的互斥子集，每个子集尽量保持数据分布的一致性，即从D中通过分层采样得到。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，这样得到K个测试结果的平均。

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，因此交叉验证法又称为“K折交叉验证”

K折交叉验证也可以通过随机使用不同的划分重复P次，最终的苹果结果是这P次K折交叉验证结果的均值。


D
D1 D2 D3 D4 D5 D6 D7 D8 D9 D10
训练集
测试集
D1 D2 D3 D4 D5 D6 D7 D8 D9
D10
测试结果 1
D1 D2 D3 D4 D5 D6 D7 D8 D10
D9
测试结果 2
D2 D3 D4 D5 D6 D7 D8 D9 D10
D1
测试结果10
平均
返回结果

10折交叉验证示意图

评估方法

- **自助法（bootstrapping）**

实现过程：直接以自助采用（bootstrap sampling）为基础，即给定包含m个样本，将其拷贝放入D‘，然后再将该样本放回本初始数据集中，使得该样本下次采样时仍可能被踩到；这个过程重复执行m次后，得到包含m个样本的数据集D’，而另一部份样本不出现。

样本在m次采样中始终不被采到的概率为（1-1/m)^m

其极限表示为：

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

评估方法

调参与最终模型：

调参（parameter tuning）：在机器学习中，在进行模型评估与选择是，除了要对适用学习算法进行选择，还需要对算法参数进行设定，这就是“参数调节”，简称“调参”。

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)**。例如，在研究对比不同算法的泛化性能时，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**。

性能度量

概念：**衡量模型泛化能力的标准称为“性能度量(performance measure)”**。

在预测任务重，给定样例集$D=\{(x_1,y_1),(x_2,y_2),\ldots,(x_m,y_m)\}$，其中yi是$x_i$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(x)与真实标记y进行比较。

回归任务最常用的性能度量是“均方误差”(mean squared error)

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

更一般，对于数据分布 D 和概率密度函数p(x)，均方误差可描述为：

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

性能度量——分类器性能度量

错误率与精度

对样本集D，错误率定义为

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

精度则定义为

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

性能度量——分类器性能度量

更一般的，对于数据分布Ð 和概率密度函数p(.)， 错误率与精度可分别描述为

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

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

性能度量——分类器性能度量

查准率，查全率与Ｆ１

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 与查全率 R 分别定义为

$$P = \frac{TP}{TP + FP},$$

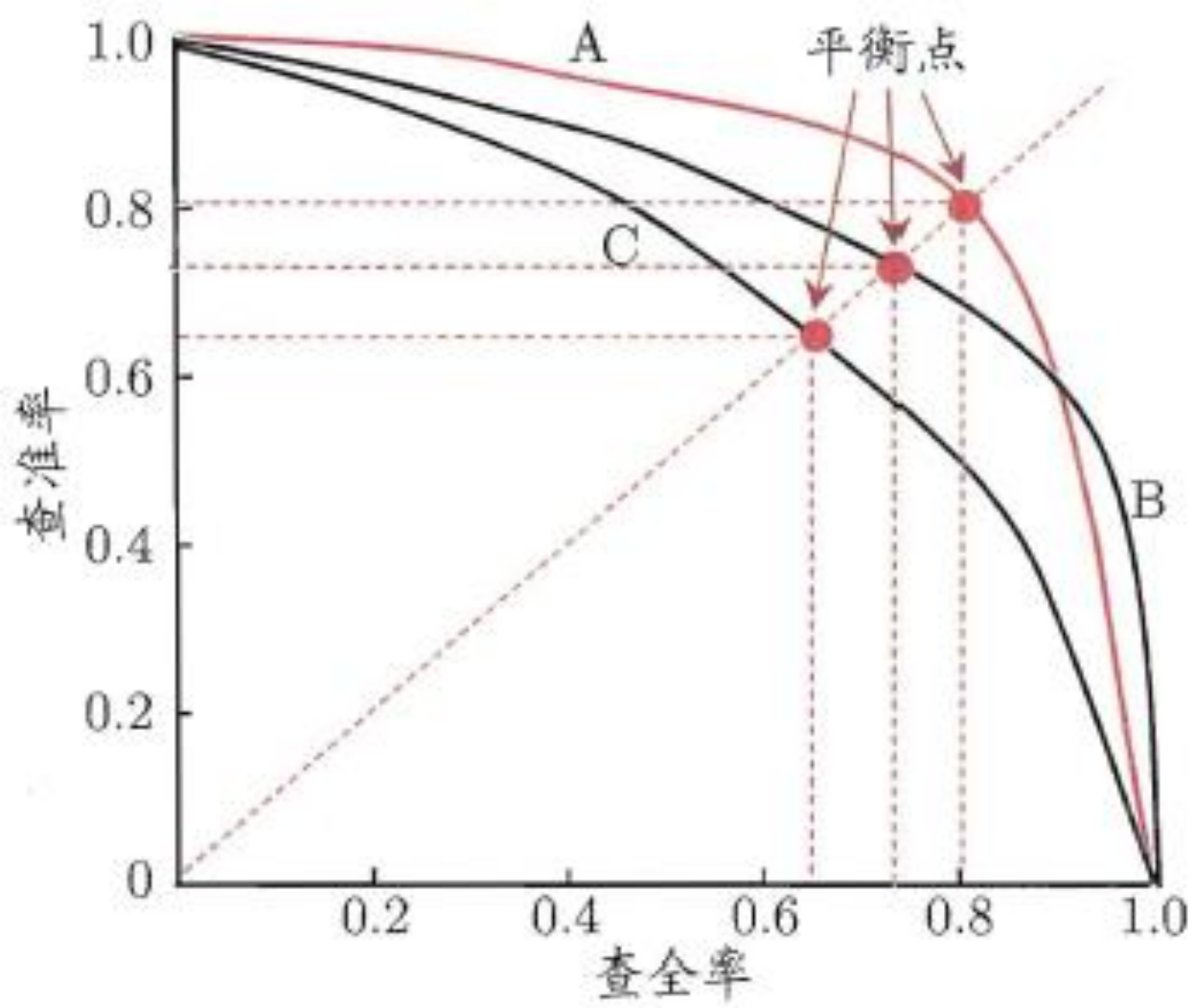
$$R = \frac{TP}{TP + FN}.$$

性能度量——分类器性能度量

P –R曲线：

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；查全率高时，查准率往往偏低。

在很多情况下，可根据学习器的预测结果对样例进行排序，排在前面的学习器认为“最可能”是正例样本，趴在最后的则是学习器认为“最不可能”是正样例的样本。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。一查全率为横轴、查准率为纵轴，得到查准率－查全率曲线，即为：Ｐ－Ｒ曲线。



平衡点（Break-Even Point）：

BEP,查准率＝查全率时的取值。

如何通过P-R曲线评价学习器优劣：

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”，则可断言后者性能优于前者。如果发生交叉，则对曲线所包含的面积进行比较。

性能度量

Ｆ１度量：

Ｆ１是基于查准率与查全率的调和平均，即：

1/F1 = 1/2 (1/P + 1/R)

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

性能度量

F_β度量：

F_β是查准率与查全率的加权调和平均，即：

$$1/F_\beta = 1/(1+\beta^2)\ (1/P + \beta^2/R)$$

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

β >0度量了查全率与准确率的相对重要性。 B＝ 1 退化为Ｆ 1，
β>1查全率有更大影响； β<1时查准率有更大影响。

性能度量

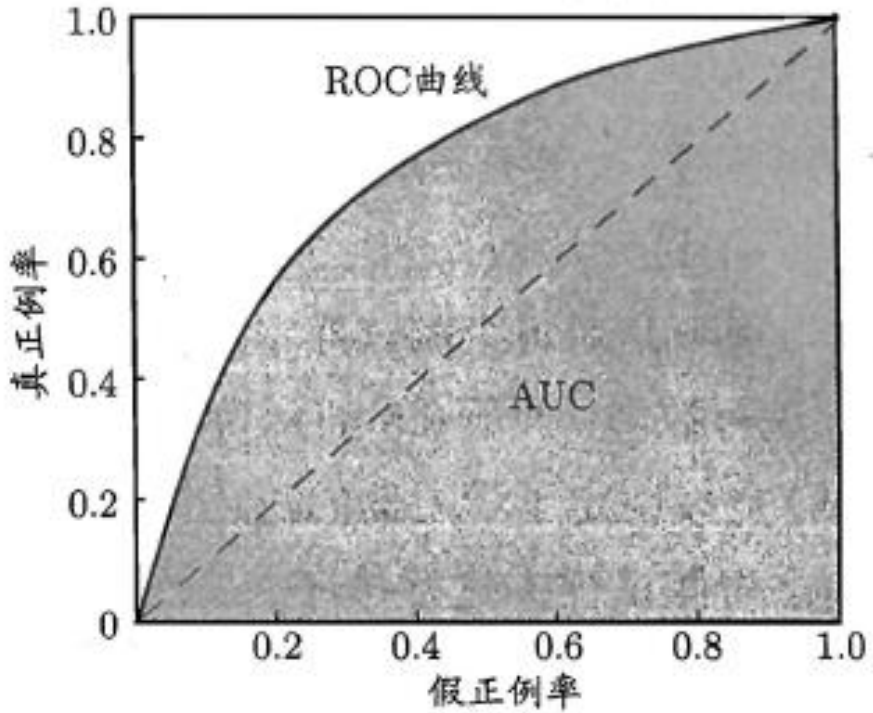
ROC与AUC

ROC是研究学习器泛化性能的有力工具，AUC即为ROC曲线下的面积（Area under ROC Curve）。

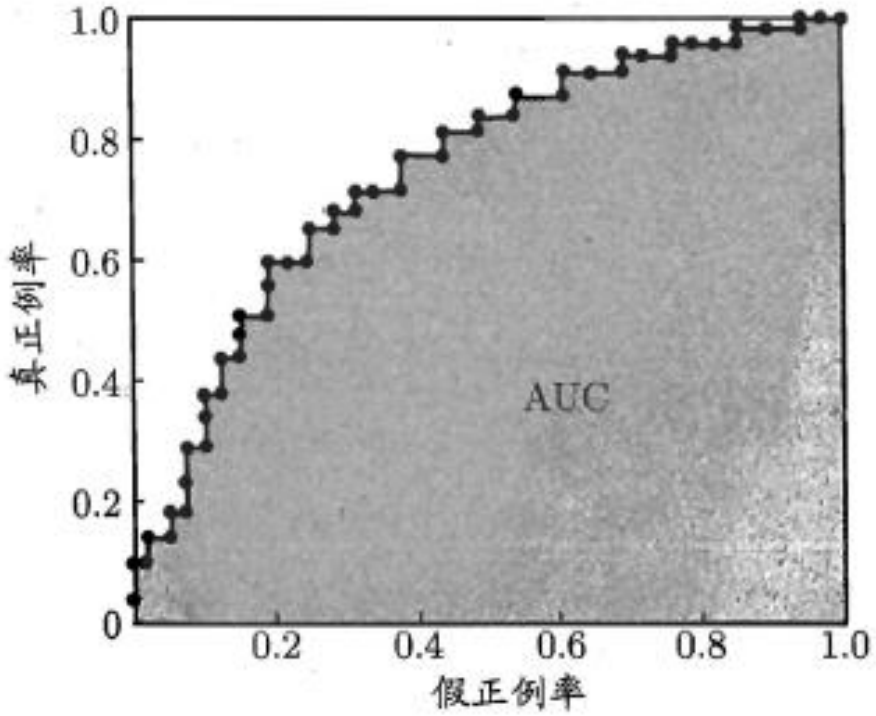
ROC全称“受试者工作特征曲线”，根据学习器的预测结果对样例进行排序，按此顺序逐个把样本作为正例进行预测，每次计算出两个重要的值“真正例率(TPR)”、“假正例率(FPR)”，分别以他们为横纵坐标。

$$\mathrm{TPR} = \frac{TP}{TP + FN},$$

$$\mathrm{FPR} = \frac{FP}{TN + FP}.$$



(a) ROC 曲线与 AUC



(b) 基于有限样例绘制的 ROC 曲线与 AUC

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-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 Under ROC Curve) ，

AUC 可估算为：

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

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十个正例和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：

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

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

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， loss函数对应的是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:

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

代价敏感错误率与代价曲线

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" (unequa1 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" (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

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， $cost_{ii} = 0$; 若将第0 类判别为第1 类所造成的损失更大，则 $cost_{01} > cost_{10}$； 损失程度相差越大，$cost_{01}$ 与$cost_{10}$ 值的差别越大。

代价敏感错误率与代价曲线

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 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

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

代价敏感错误率与代价曲线

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，

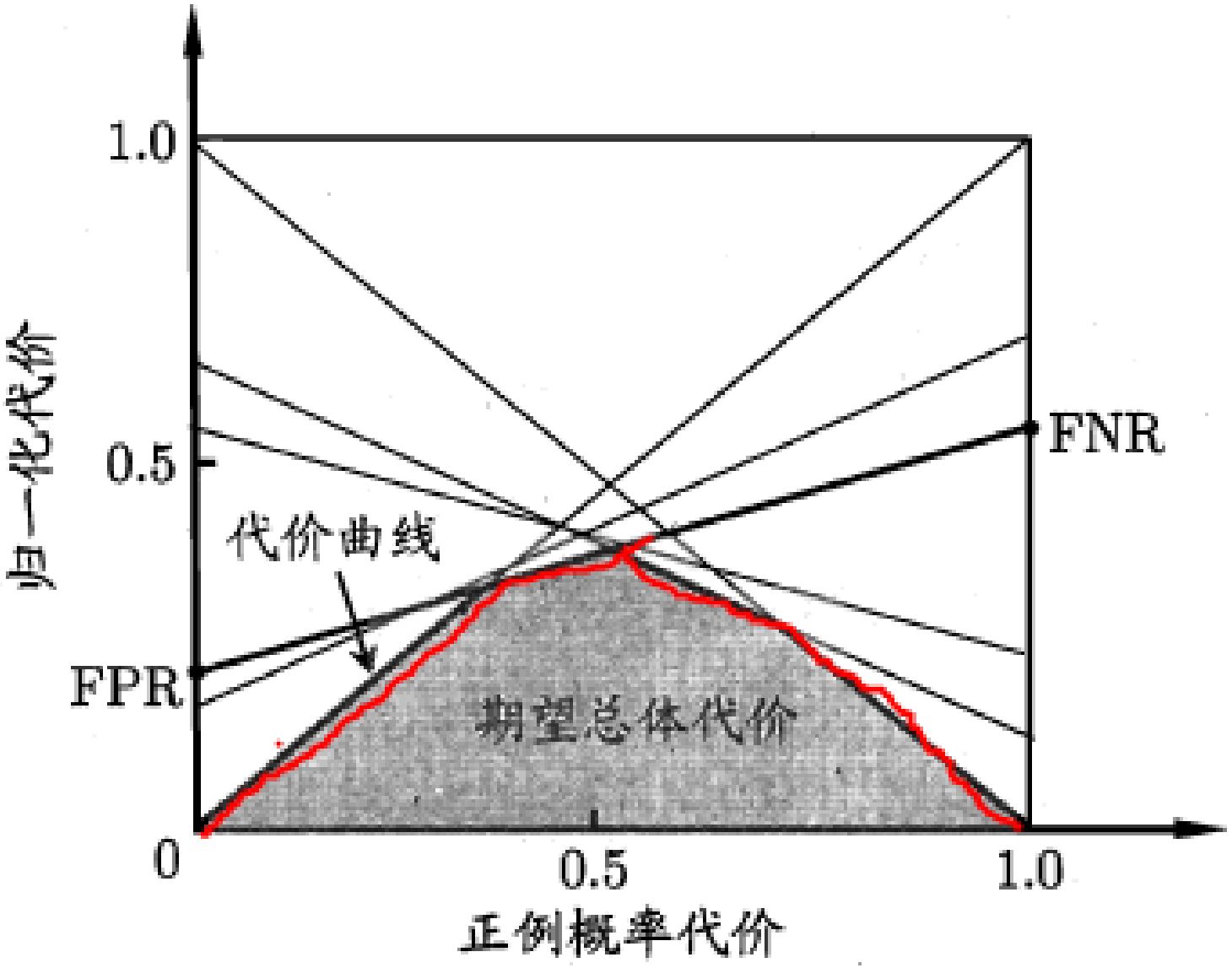
$$P(+)cost = \frac{p \times cost_{01}}{p \times cost_{01} + (1-p) \times cost_{10}},$$

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

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

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

其中FPR 是式(2.19) 定义的假E例率， FNR == 1 - TPR 是假反例率.代价曲线的绘制很简单: ROC 由线上每一点对应了代价平面上的一条线段7，设ROC 曲线上点的坐标为(TPR， FPR) ，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 曲线土的每个点转化为代价平面上的一条线段，然后取所有线段的下界，围成的自积即为在所有条件下学习器的期望总体代价，如图


1.0
0.5
FPR
0
0.5
1.0
FNR
代价曲线
期望总体代价
归一化代价
正例概率代价

偏差与方差

比较检验

学习器比较困难的几个重要因素：

- 目标任务泛化能力的比较，实际获得的是测试集上的性能，二者结果未必相同
- 测试集上的性能与测试集本身的选择有很大关系
- 很多机器学习算法本身有一定的随机性，其结果不一定相同。

统计假设检验(hypothesis test)为学习器性能比较提供了重要依据。

比较检验（略）

学习器比较困难的几个重要因素：

- 目标任务泛化能力的比较，实际获得的是测试集上的性能，二者结果未必相同
- 测试集上的性能与测试集本身的选择有很大关系
- 很多机器学习算法本身有一定的随机性，其结果不一定相同。

统计假设检验(hypothesis test)为学习器性能比较提供了重要依据。

比较检验（略）

假设检验

偏差与方差

"偏差方差分解" (bias-variance decomposition)是解释学习算法泛化性能的一种重要工具。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;

噪声则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。

偏差与方差

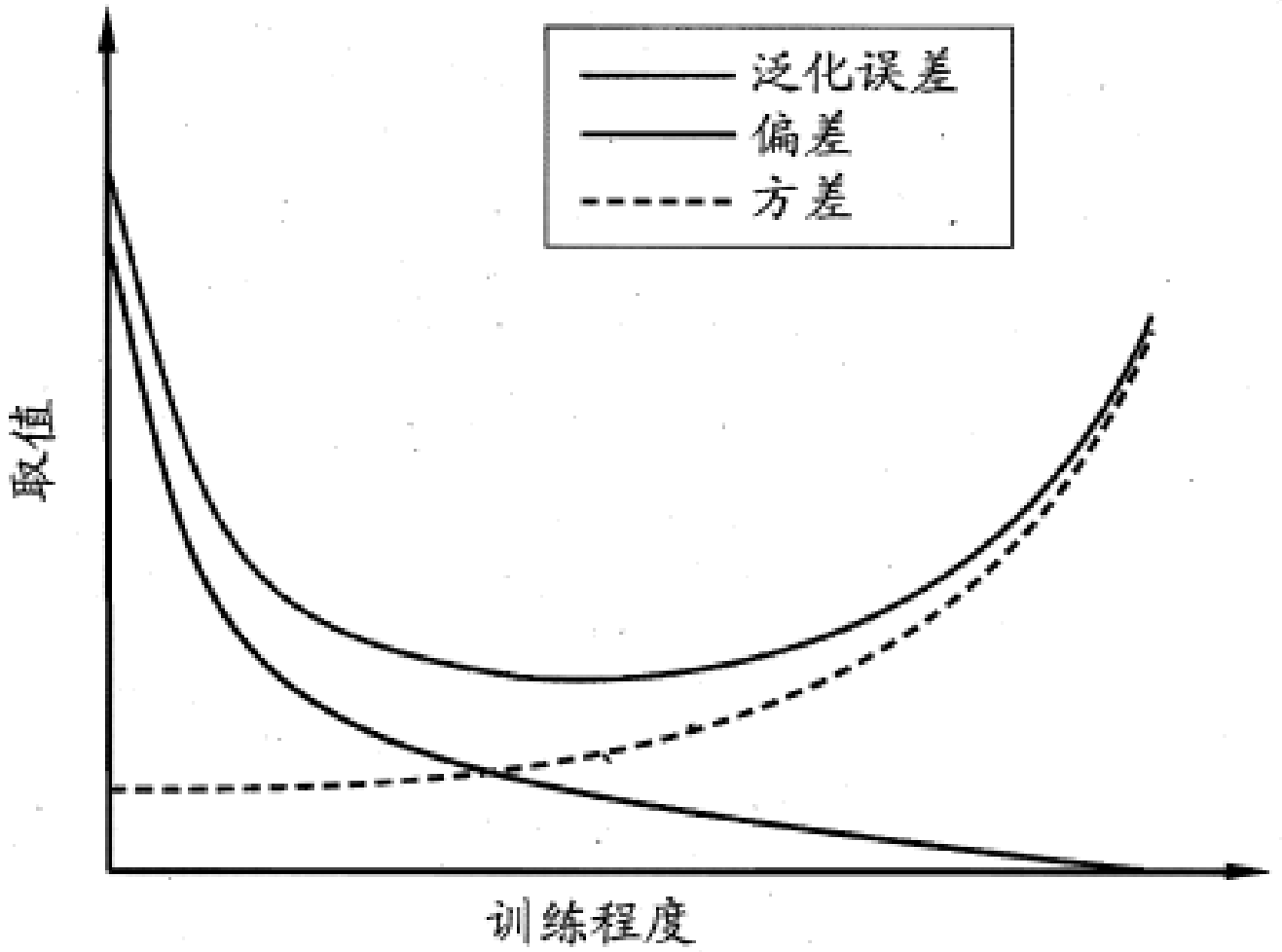
偏差一方差分解说明：泛化性能是由学习算法的能力、数据的充分性以及学习任务本身的难度所共同决定的。给定学习任务，为了取得好的泛化性能，则需使偏差较小，即能够充分拟合数据，并且使方差较小，即使得数据扰动产生的影响小。

偏差与方差是有冲突的，这称为偏差一方差窘境(bias-variance dilemma).

偏差与方差

图 给出了一个示意图给定学习任务，假定我们能控制学习算法的训练程度，则在训练不足时，学习器的拟合能力不够强，训练数据的扰动不足以便学习器产生显著变化，此时偏差主导了泛化错误率;

随着训练程度的加深，学习器的拟合能力逐渐增强，训练数据发生的扰动渐渐能被学习器学到，方差逐渐主导了泛化错误率;在训练程度充足后，学习器的拟合能力已非常强，训练数据发生的轻微扰动都会导致学习器发生显著变化，若训练数据自身的、非全局的特性被学习器学到了，则将发生过拟合.



泛化误差与偏差、方差关系示意图

本章小结

- 误差的几个概念，重点：过拟合和欠拟合
- 评估的集中方法，重点：测试集与训练集如何划分，调参的概念
- 性能度量，重点：回归任务与分类任务性能度量，分类性能度量P-R曲线、ROC、AUC曲线
- 偏差与方差，重点：泛化性能与偏差、方差关系。

本章概念性比较多，难理解。应付之道：多看，多捉摸。