模型评估与选择

Group Study, ML: 2

周晓清 2020年6月12日

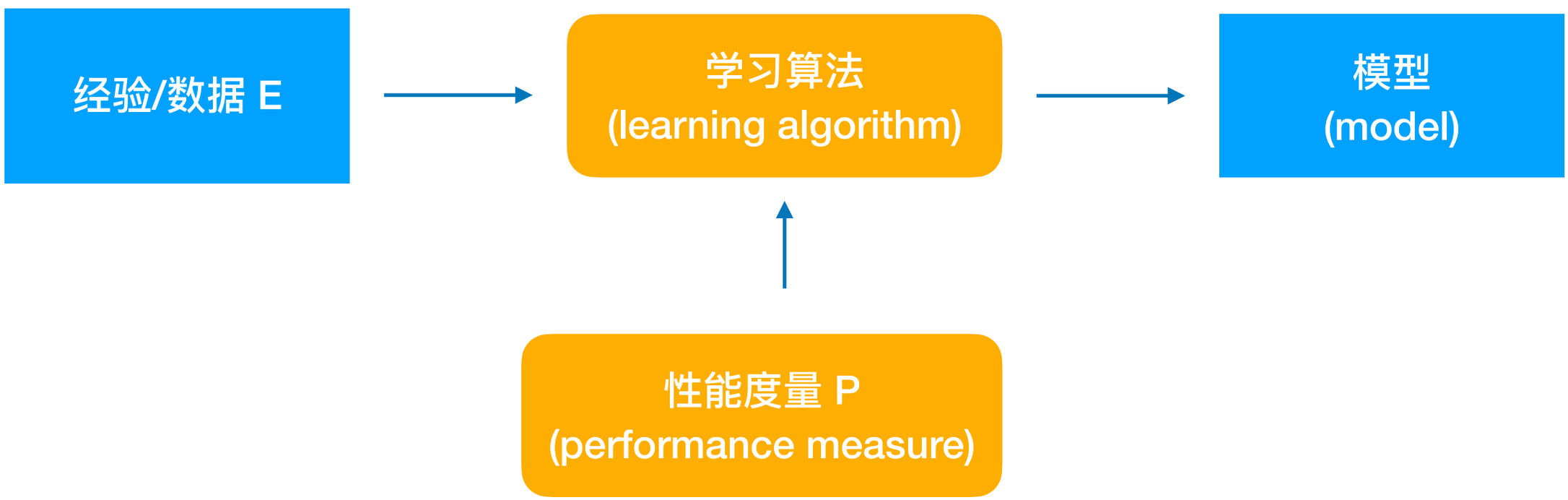
机器学习是人工智能发展的产物

	推理期 (1950s 至 pre-1970s)	**知识期** (mid-1970s 至 1980s)	**学习期** (1980s 至今)
观点	智能 = 逻辑推理能力	使机器拥有知识	使机器自己学习知识 **↑ 机器学习**
应用	证明数学定理	大量专家系统问世	计算机视觉、自然语言处理等“计算机应用”领域
缺陷	仅有逻辑推理能力无法实现人工智能	知识工程瓶颈	

机器学习的定义

“ 假设用性能度量 P 来评估计算机程序在某任务类 T 上的性能, 若一个程序通过利用经验 E 在 T 中任务上获得了性能改善, 则我们就说关于 T 和 P , 该程序对 E 进行了学习. ”

— Tom M. Mitchell, 1997



- **2.1 经验误差与过拟合**
- 2.2 评估方法
- 2.3 性能度量
- 2.4 比较检验
- 2.5 偏差与方差

经验误差与过拟合

- 机器学习的目标: 使学得的模型能很好地适用于“新样本”, 即使泛化误差(generalization error)最小.

- 相关术语

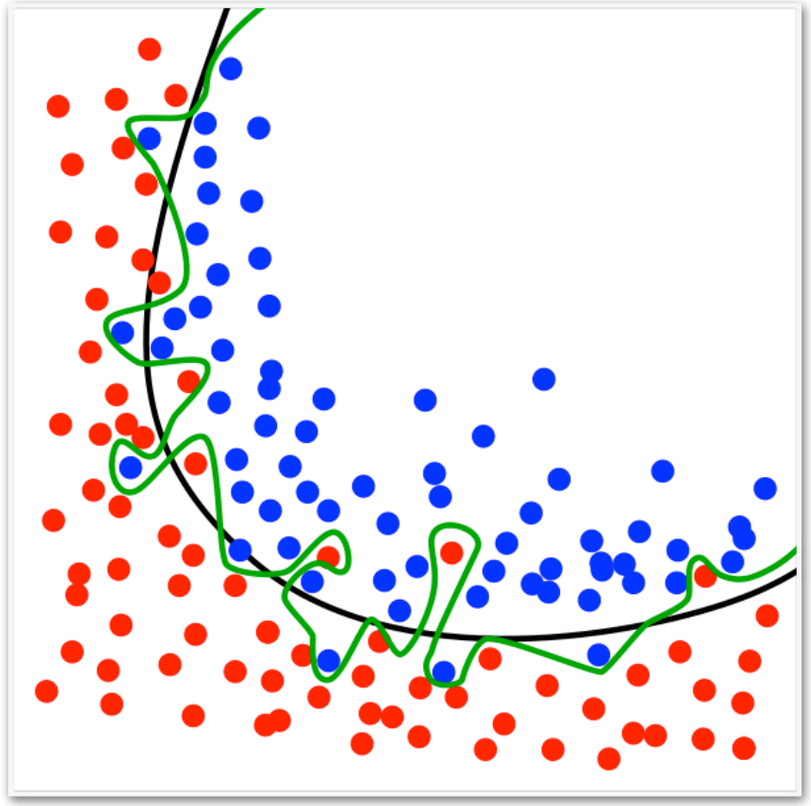
  $$R_{emp} = \frac{1}{m}\sum_{i=1}^{m} L(y_i, f(x_i))$$

  - 误差(error)
    学习器的预测输出与样本的真实输出之间的损失. 分为经验误差(empirical error)和泛化误差(generalization error).

  - 错误率(error rate)
    如果 m 个样本中有 a 个分类错误, 则错误率 E = a / m.

  - 精度(accuracy)
    精度 = 1 - 错误率 = 1 - a / m.

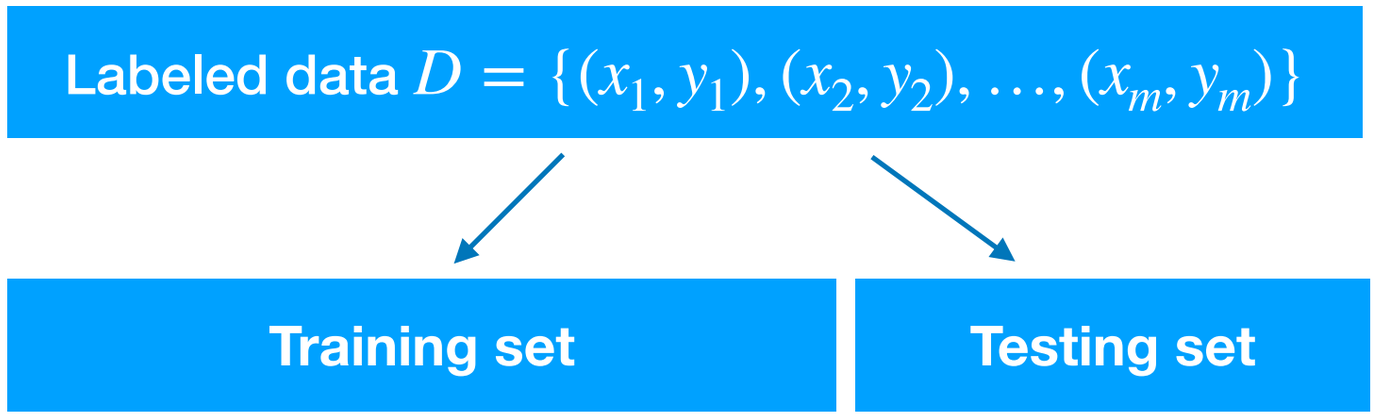
经验误差与过拟合



- 实际可行的做法: 使经验误差最小化.
- 经验误差越小越好吗?
  - 过拟合(overfitting): 把训练样本自身的特点当作了潜在样本会具有的一般性质, 导致泛化性能下降.
  - 违反归纳偏好. 奥卡姆剃刀原则: 若有多个假设与观察一致, 则选最简单的那个.
  - 过拟合无法彻底避免的原因: 噪声和计算复杂度.
- 结论: 经验误差由于过拟合现象的存在不适合作为评估标准.

- 2.1 经验误差与过拟合
- **2.2 评估方法**
- 2.3 性能度量
- 2.4 比较检验
- 2.5 偏差与方差

评估方法



- 使用测试集(testing set)来测试学习器对新样本的判别能力，以测试误差(testing error)作为泛化误差的近似.
- 产生训练集和测试集的原则:
  - 测试集从样本真实分布中独立同分布采样而得.
  - 测试集与训练集互斥.
  - 多次重复划分.

1 留出法(hold-out)

- 直接将数据集 D 划分为两个互斥集合.
  $D = S \cup T, S \cap T = \varnothing$

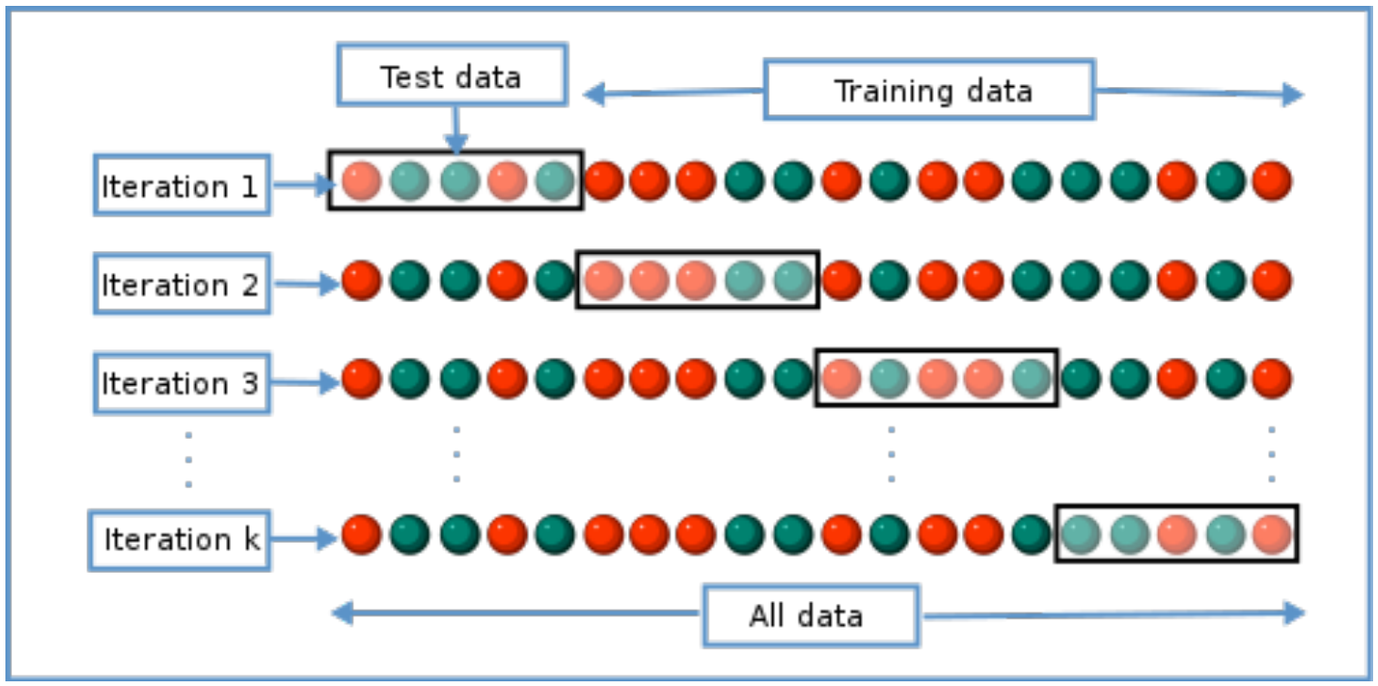
	样本个数	正例个数	反例个数
数据集 D	1000	500	500
训练集 S	700	350	350
测试集 T	300	150	150

- 采用若干次随机划分、重复进行实验评估后取平均值.
- 缺陷: 训练样本规模不同导致估计偏差.
  - 若 S 包含大多数样本, 模型接近于用 D 训练的模型, 但由于 T 比较小, 评估结果不够稳定准确. 即评估结果的方差较大.
  - 若 T 多包含一些样本，则 S 和 D 差别大, 被评估的模型和 D 训练出的模型相比可能有较大差别, 从而降低评估结果保真性(fidelity). 即评估结果的偏差较大.
  - 一般将大约 2/3 ~ 4/5 的样本用于训练.

2 交叉验证法(cross validation)

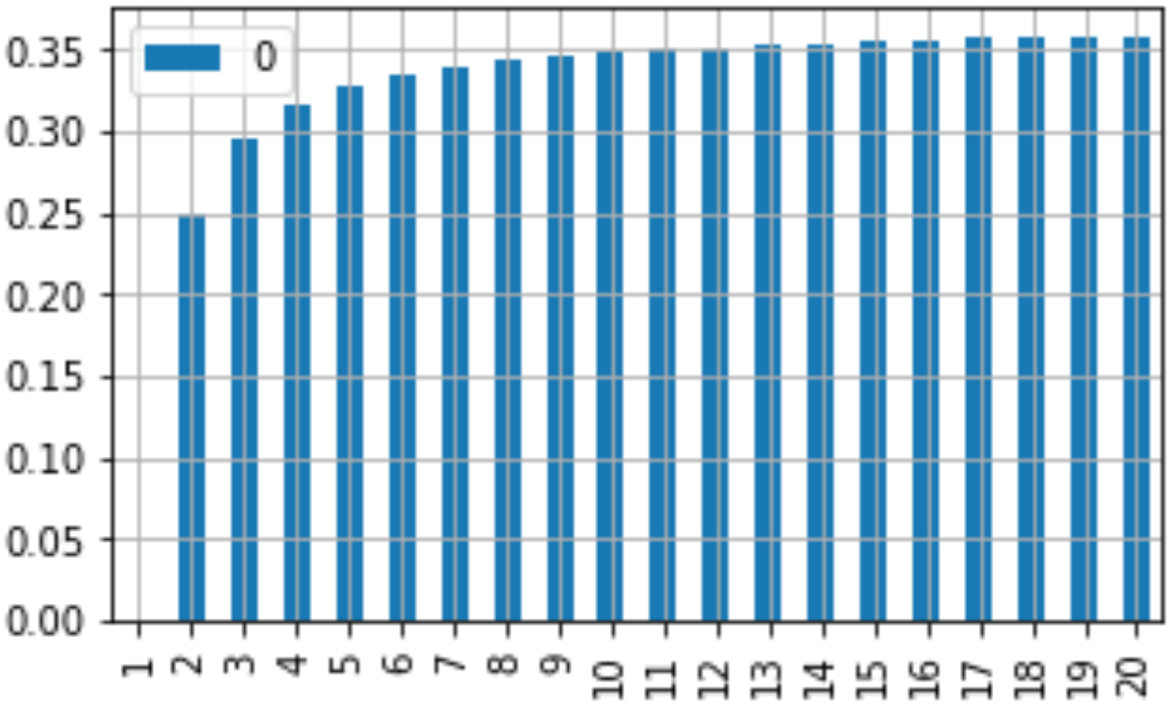
- k 折交叉验证: 先将数据集 D 划分为 k 个大小相似的互斥子集, 然后每次用 k-1 个子集的并集作为训练集, 余下的子集作为测试集.

$$D = D_1 \cup D_2 \cup \ldots \cup D_k, D_i \cap D_j = \varnothing (i \neq j)$$



- p 次 k 折交叉验证: 减少因样本划分不同而引入的差别, 共 p*k 次训练.
- 特例: 留一法(Leave-One-Out, LOO), k = m.
  - 优点: 被实际评估的模型与期望评估的用 D 训练出的模型很相似.
  - 缺陷: 计算复杂度高.

3 自助法(bootstrapping)



样本在 m 次采样中始终不被采到的概率:

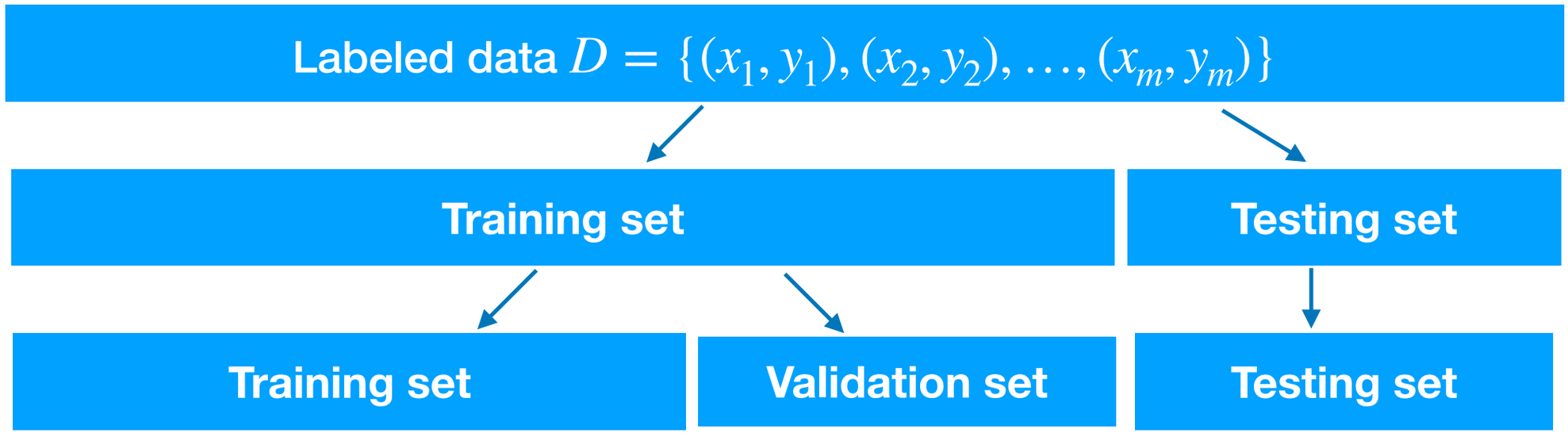
$$\lim_{m\to\infty}(1-\frac{1}{m})^m \to \frac{1}{e} \approx 0.367879$$

3 自助法(bootstrapping)

- 优点
  - 训练集与原样本集同规模.
  - 在数据集较小、难以有效划分训练/测试集时很有用.
  - 能从原始数据中产生多个不同的训练集, 对集成学习等方法有很大好处.
    - 集成学习: 使用多种学习算法来获得比单独使用任何单独的学习算法更好的预测性能。
- 缺陷
  - 改变了初始数据集的分布, 这会引入估计偏差.

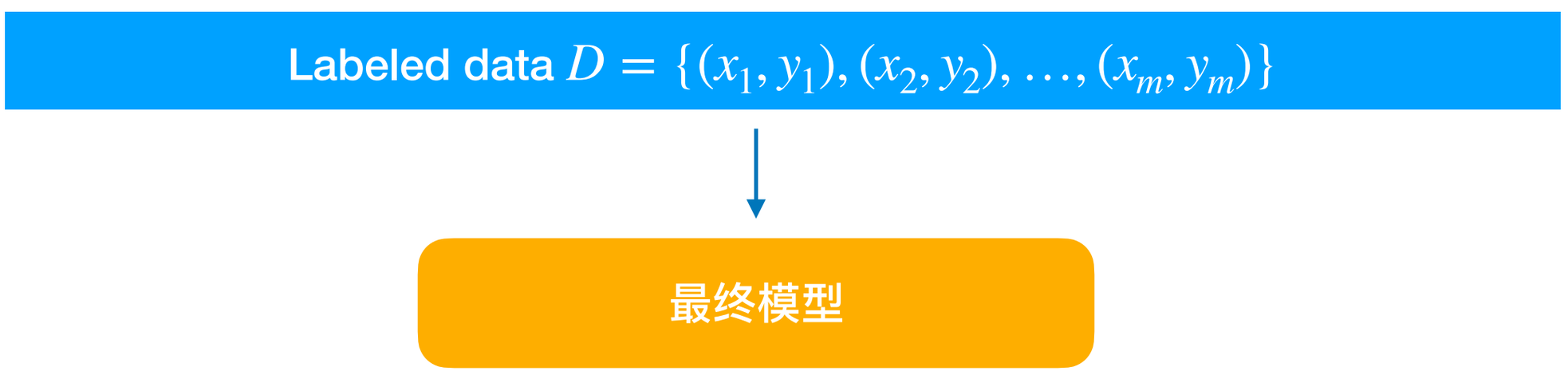
- 结论: 在初始数据量足够时, 留出法和交叉验证法更常用一些.

4 调参与最终模型



模型训练　　模型选择和调参　　评估实际泛化能力

学习算法和参数配置选定后，
用数据集 D 重新训练模型.

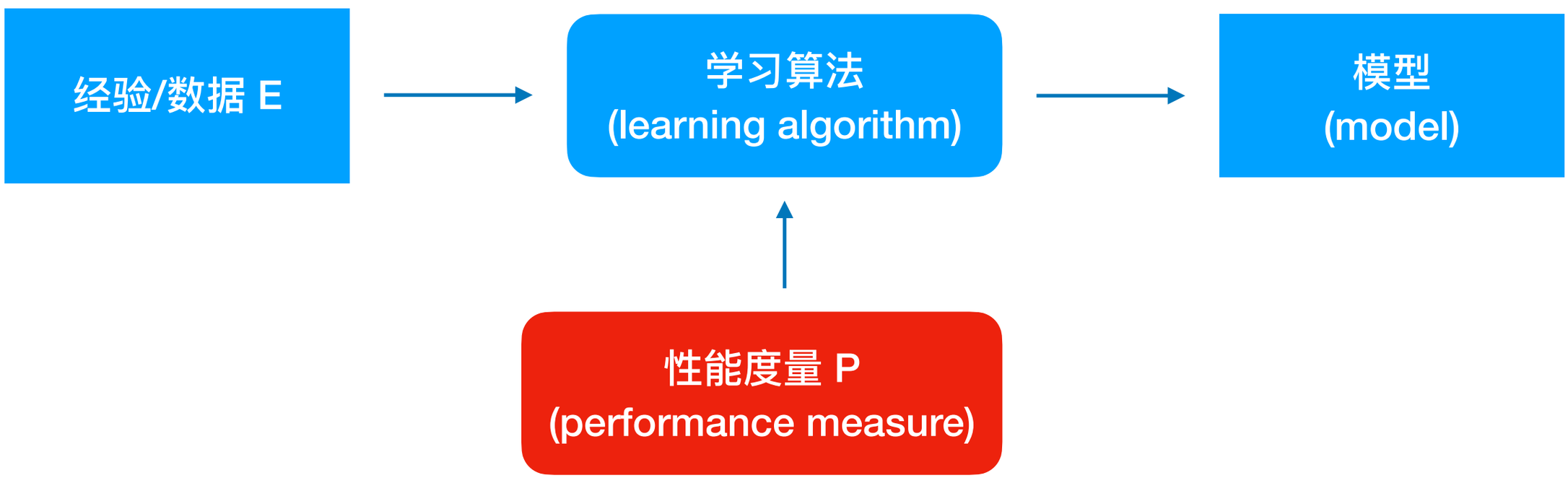


- 2.1 经验误差与过拟合
- 2.2 评估方法
- **2.3 性能度量**
- 2.4 比较检验
- 2.5 偏差与方差

机器学习的定义

“ 假设用性能度量 P 来评估计算机程序在某任务类 T 上的性能, 若一个程序通过利用经验 E 在 T 中任务上获得了性能改善, 则我们就说关于 T 和 P , 该程序对 E 进行了学习. ”

— Tom M. Mitchell, 1997



性能度量: 衡量模型泛化能力的评价标准

- 性能度量反应了任务需求.
- 对监督学习, 给定样例集 $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$, 其中 y_i 是样本 x_i 的真实标记. 性能度量将模型预测结果 $f(x)$ 与真实标记 y 进行比较.

监督学习	回归任务	• 均方误差
	分类任务	• 错误率与精度 • 查准率、查全率与F1 • ROC 与 AUC • 代价敏感错误率与代价曲线
	标注任务	• 精度均值(average precision, AP) • 平均精度均值(mean average precision, mAP)
无监督学习	聚类	

均方误差(mean squared error, MSE)

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

- 取平方, 以避免误差正负抵消.
- 除以 m, 以去除样本数量的影响.

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

- 开平方, 以保持原量纲.

错误率与精度

- 分类任务中最常用的两种性能度量, 既适用于二分类任务, 也适用于多分类任务.
  - 分类错误率 $E(f;D)=\frac{1}{m}\sum_{i=1}^{m}\mathbb{I}(f(x_i)\neq y_i)$
  - 精度 $acc(f;D)=\frac{1}{m}\sum_{i=1}^{m}\mathbb{I}(f(x_i)=y_i)=1-E(f;D)$

查准率、查全率与 F1

- 错误率和精度不能满足所有的任务需求.
  - 挑出的西瓜中有多少比例是好瓜? 所有好瓜中有多少比例被挑出来了?
- 混淆矩阵(confusion matrix): 分类预测结果的情形表示. 如对二分类问题, 将样例根据其真实类别与预测类别的组合划分为四种情形, 每种情形的样例数组成混淆矩阵.
  - TP + FP + TN + FN = 样例总数
- 正确预测位于对角线上, 错误预测由对角线之外的元素表示.

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(precision): $P = \frac{TP}{TP + FP}$
- 查全率(recall): $R = \frac{TP}{TP + FN}$

查准率、查全率与 F1

- 查准率和查全率是一对矛盾的度量, 不能仅用二者之一作为性能度量.
- P-R 图显示学习器在样本总体上的查全率、查准率.
  - P-R 曲线: 根据学习器的预测结果, 按正例可能性大小对样例进行排序, 并逐个把样本作为正例进行预测, 计算出当前的查全率、查准率.
- 平衡点(Break-Even Point, BEP): “查全率=查准率”时的取值.
  - BEP(A) > BEP(B), A 优于 B
  - 过于简化, 更常用的是 F1 度量.

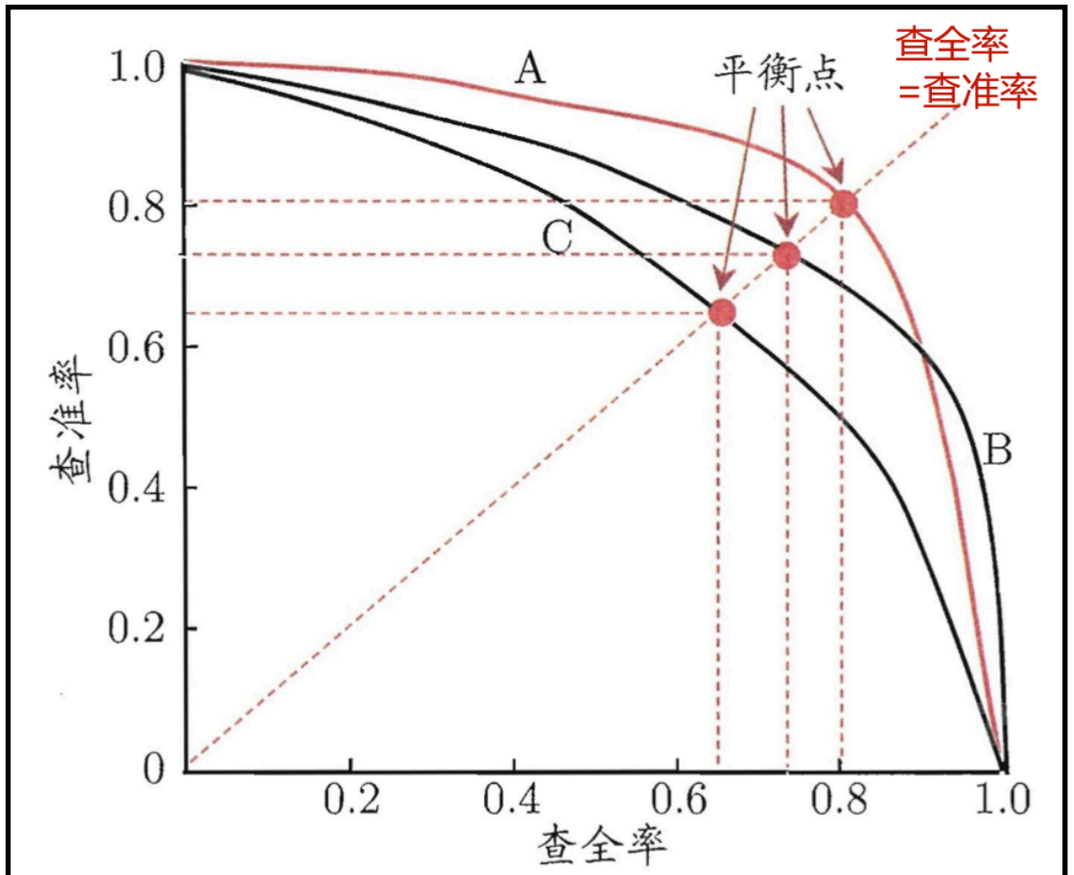


图 2.3 P-R曲线与平衡点示意图

- 学习器 A 优于学习器 C
- 学习器 B 优于学习器 C.
- 学习器 A 和学习器 B ?

查准率、查全率与 F1

- F1 是查准率和查全率的调和平均:

$$\frac{2}{F_1} = \frac{1}{P} + \frac{1}{R}, \quad F_1 = \frac{2 \times P \times R}{P + R} = \frac{2 \times TP}{m + TP - TN}$$

- 若对查准率和查全率有不同偏好, 使用查准率和查全率的加权调和平均:

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

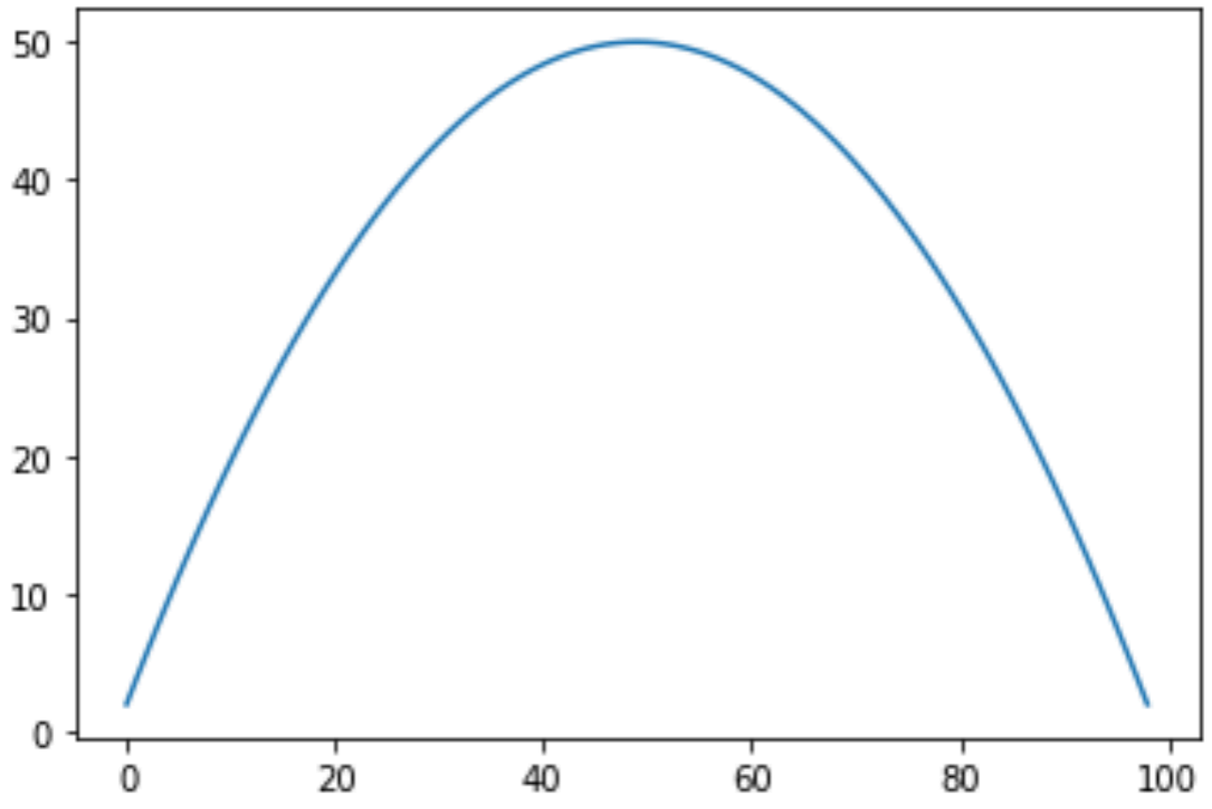
- $\beta > 0$ 度量了查全率对查准率的相对重要性:
  - $\beta = 1$ 退化为标准的 F1;
  - $\beta > 1$ 查全率有更大影响;
  - $\beta < 1$ 查准率有更大影响.

查准率、查全率与 F1

- 为什么使用调和平均?

$$\frac{2}{F_1} = \frac{1}{P} + \frac{1}{R}, \quad F_1 = \frac{2 \times P \times R}{P + R} = \frac{2 \times TP}{m + TP - TN}$$

```
harmonic_means = [2*x*(100-x)/100 for x in range(1, 100)]
plot = pd.DataFrame(harmonic_means).plot.line(legend=False)
```



x	100-x	算数平均	调和平均
50	50	50	50
40	60	50	48
30	70	50	42
20	80	50	32

查准率、查全率与 F1

- 在 n 个二分类混淆矩阵上综合考察查准率和查全率:
  - 如多次训练/测试的结果、多分类任务的两两混淆矩阵.
  - 先在各混淆矩阵上分别计算查准率和查全率, 再取平均值:

$$macroP = \frac{1}{n}\sum_{i=1}^{n} P_i \qquad macroR = \frac{1}{n}\sum_{i=1}^{n} R_i$$

$$macroF1 = \frac{2 \times macroP \times macroR}{macroP + macroR}$$

  - 或先将各混淆矩阵的对应元素进行平均, 得到 $\overline{TP}, \overline{FP}, \overline{TN}, \overline{FN}$, 再基于这些平均值计算:

$$microP = \frac{\overline{TP}}{\overline{TP} + \overline{FP}} \qquad microR = \frac{\overline{TP}}{\overline{TP} + \overline{FN}}$$

$$microF1 = \frac{2 \times microP \times microR}{microP + microR}$$

ROC 与 AUC

- ROC (Receiver Operating Characteristic, 受试者工作特征) 曲线 —— 衡量样本预测的排序质量.
  - 排序质量本身的质量好坏, 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, 或者说, “一般情况下”泛化性能的好坏.
  - 与 P-R 曲线使用查准率、查全率为横、纵轴不同. ROC 曲线的纵轴是“真正例率” (True Positive Rate, TPR), 横轴是“假正例率” (False Positive Rate, FPR):

$$TPR = \frac{TP}{TP + FN} = \frac{TP}{m^+}$$

$$FPR = \frac{FP}{TN + FP} = \frac{FP}{m^-}$$

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)

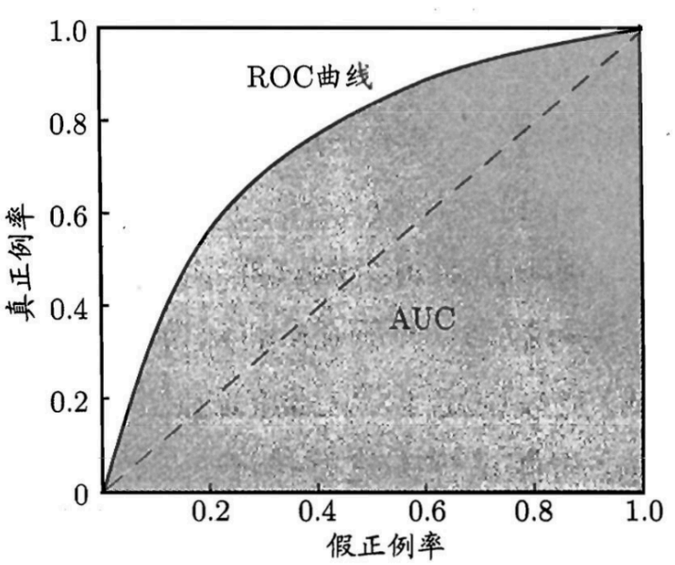
- ROC 曲线和 P-R 曲线
  - P-R 曲线中, 由一次训练结果(查全率、查准率)得到曲线上的一个点.
  - ROC 曲线中, 由一次训练结果(测试样本根据概率预测结果排序)得到了整条曲线.

ROC 与 AUC

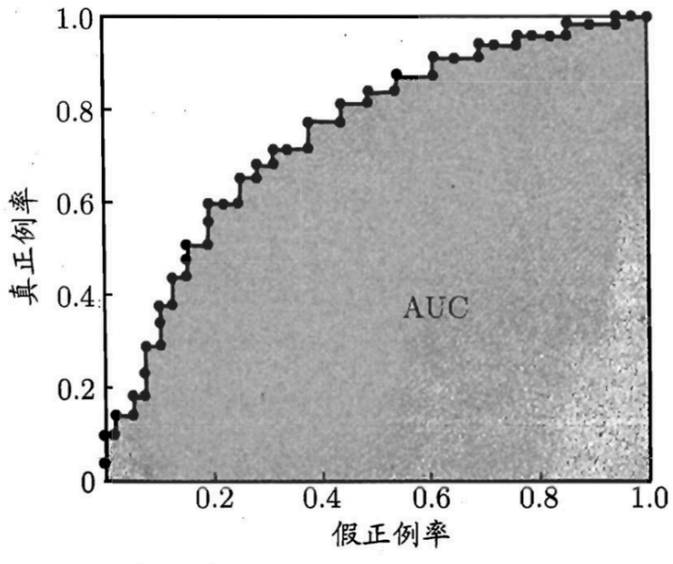
- 例: 绘制 ROC 曲线
  - $TPR = \frac{TP}{TP + FN} = \frac{TP}{m^+}$
  - $FPR = \frac{FP}{TN + FP} = \frac{FP}{m^-}$

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)

ROC 与 AUC



(a) ROC 曲线与 AUC



(b) 基于有限样例绘制的 ROC 曲线与 AUC

+1)

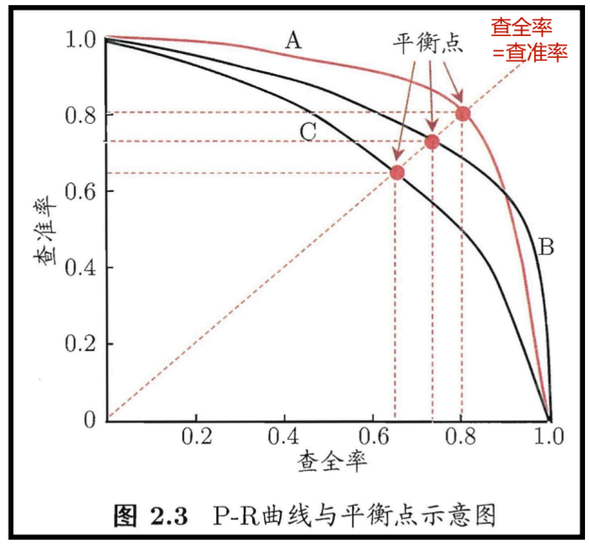


图 2.3 P-R曲线与平衡点示意图

- AUC(Area Under ROC Curve), 越大越好:

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

- 排序损失, 对应曲线之上的面积:

$$l_{rank} = \frac{1}{m^+m^-}\sum_{x^+\in D^+}\sum_{x^-\in D^-}(I(f(x^+) < f(x^-)) + \frac{1}{2}I(f(x^+) = f(x^-)))$$

  - 考虑每一对正反例, 若正例的预测值小于反例, 则记 1 个“罚分”, 若相等, 则记 0.1 个“罚分”.

代价敏感错误率

- 犯不同类型错误往往会造成不同损失, 需要为错误赋予 “非均等代价” (unequal cost).
- 以二分类任务为例, 根据任务的领域知识设定 “代价矩阵” (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

- 重要的是代价比值而非绝对值.

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)

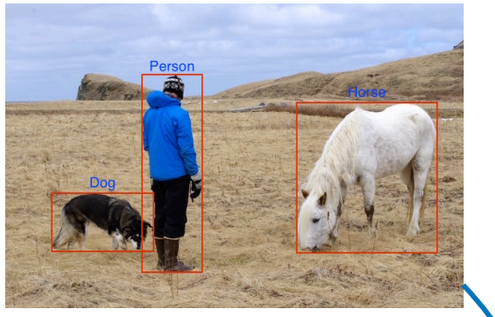
混淆矩阵

- 所希望的不再是最小化错误次数, 而是最小化“总体代价” (total cost), 如 “代价敏感” (cost-sensitive) 错误率:

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

AP 与 mAP

- Object detection

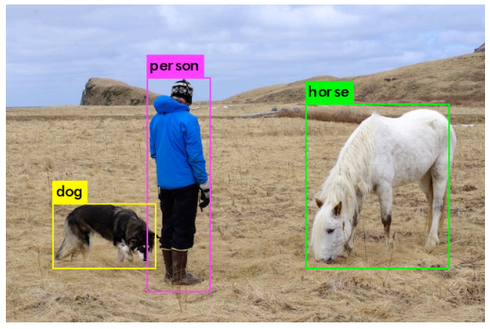


Human visualisation of the ground truth

The actual image

Model

Confidence threshold



Results from our model

LoU threshold

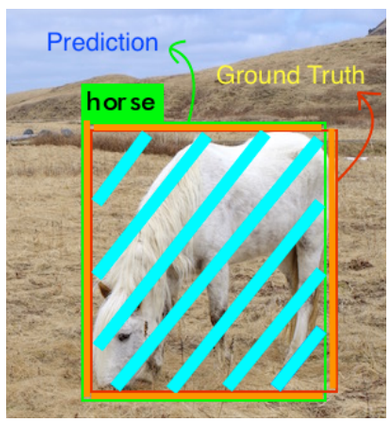
TP	FN
FP	TN

Precision
Recall

- Structure prediction

Class	X coordinate	Y coordinate	Box Width	Box Height
Dog	100	600	150	100
Horse	700	300	200	250
Person	400	400	100	500

- IoU - Intersection over Union



$$IoU = \frac{\text{Intersection}}{\text{Union}}$$

Measuring Object Detection models-mAP-What is Mean Average Precision?

AP 与 mAP

- AP (average precision): the mean precision at a set of eleven equally spaced recall levels [0, 0.1, 0.2, …, 1].
  - The precision at each recall level r is interpolated by taking the maximum precision measured for a method for which the corresponding recall exceeds r.
- mAP (mean average precision): the Mean of all the Average Precision values across all classes.

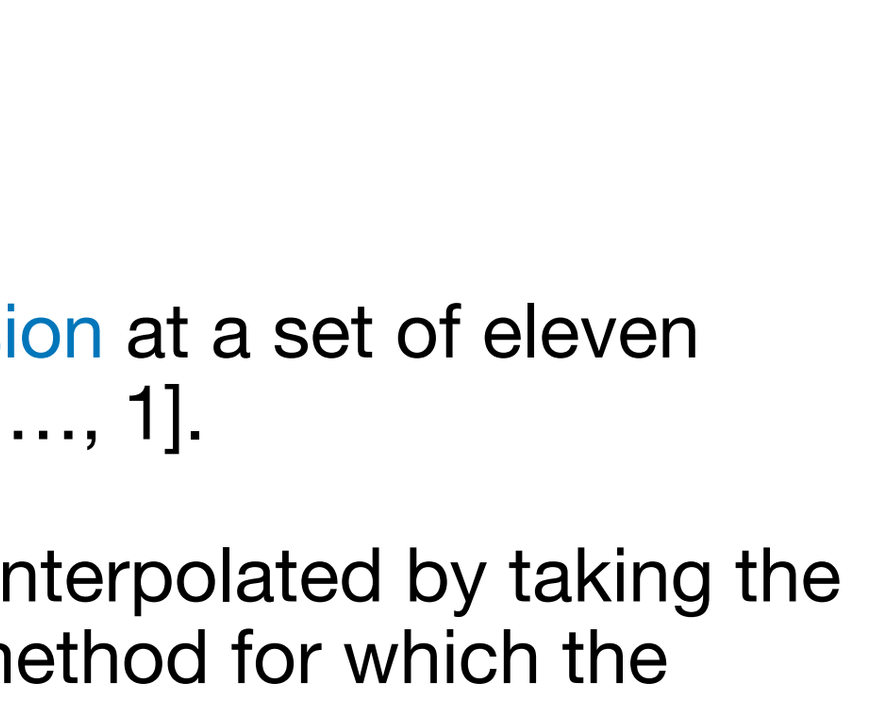


图 2.3 P-R曲线与平衡点示意图

Measuring Object Detection models-mAP-What is Mean Average Precision?

- 2.1 经验误差与过拟合
- 2.2 评估方法
- 2.3 性能度量
- **2.4 比较检验**
- 2.5 偏差与方差

比较检验: 如何比较不同学习器的性能

- 直接取得性能度量的值然后“比较大小”吗？❌ 性能度量是统计量.
  - 通过评估方法得到的是测试集上的性能, 未必等于泛化性能.
  - 测试集上的性能与测试集选择有关.
  - 机器学习算法本身有随机性.
- 统计假设检验 (hypothesis test) 为学习器性能比较提供了重要依据.
  - 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, 以及这个结论的把握有多大.
- 为便于讨论, 本节以错误率为性能度量, 用 ϵ 表示.

假设检验

- 假设: 关于总体的论断或命题. 此处指对学习器泛化错误率 ϵ 分布的某种判断或猜想, 例如 "$\epsilon = \epsilon_0$".

- 假设检验: 根据样本, 按照一定规则判断所做假设的真伪, 并作出接受还是拒绝假设的决定.

- 区间估计: 对于一个未知量(泛化错误率 ϵ), 得到近似值(测试错误率 $\hat{\epsilon}$)后, 还需进一步估计其误差. 区间估计即估计出一个范围, 并告知这个范围内包含该未知量真值的可信程度.

- 置信区间: 对给定的 α, 如果两个统计量 ϵ_1 和 ϵ_2 满足
  $P\{\epsilon_1 < \epsilon < \epsilon_2\} = 1 - \alpha$,
  则称随机区间 (ϵ_1, ϵ_2) 为 ϵ 的置信度为 1 - α 的置信区间.

假设检验

- 单个学习器泛化性能的假设检验
  - 二项检验 (binomial test)
  - t 检验 (t-test)
- 两学习器比较
  - 交叉验证 t 检验 (基于成对 t 检验)
  - McNemar 检验 (基于列联表, 卡方检验)
- 多学习器比较
  - Friedman 检验 (基于序值, F 检验)
  - Nemenyi 后续检验 (基于序值, 进一步判断两两差别)

二项检验: 单个学习器泛化性能的假设检验

- 在 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率为 $\hat{\epsilon}$ 的概率为:

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

  - 泛化错误率为 ϵ 的学习器在一个样本上犯错的概率是 ϵ.
  - 测试错误率为 $\hat{\epsilon}$ 意味着在 m 各测试样本中恰有 $\hat{\epsilon}\times m$ 个被误分类.

- 例: 假设 "$\epsilon \le \epsilon_0$", 则在给定置信度 1 - α 内所能观测到的最大错误率(即置信上界)为:

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

  此时若测试错误率 $\hat{\epsilon}$ 小于临界值 $\bar{\epsilon}$, 则在 α 的显著度下, 假设 "$\epsilon \le \epsilon_0$" 不能被拒绝.
  即能以 1 - α 的置信度认为, 学习器的泛化错误率不大于 ϵ_0 .

t 检验: 单个学习器泛化性能的假设检验

- 通过多次重复留出法或交叉验证法进行多次训练/测试, 会得到多个测试错误率, 此时可使用 “t 检验”.

- t 分布: 设 X 和 Y 相互独立, 且 X ~ N(0, 1), Y ~ χ^2(n), 则称随机变量 $T = \dfrac{X}{\sqrt{Y/n}}$ 服从自由度为 n 的 t 分布.

- 假定我们得到了 k 个测试错误率, $\hat{\epsilon}_1, \hat{\epsilon}_2, \ldots, \hat{\epsilon}_k$, 则平均测试错误率 μ 和方差 σ^2 为 $\mu = \dfrac{1}{k}\sum_{i=1}^{k}\hat{\epsilon}_i, \quad \sigma^2 = \dfrac{1}{k-1}\sum_{i=1}^{k}(\hat{\epsilon}_i - \mu)^2$.

  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 $\tau_t = \dfrac{\sqrt{k}(\mu - \epsilon_0)}{\sigma}$ 服从自由度为 k - 1 的 t 分布.

交叉验证 t 检验: 两学习器比较

- 对两个学习器 A 和 B, 使用 k 折交叉验证法得到测试错误率分别为 $\epsilon_1^A, \epsilon_2^A, \ldots, \epsilon_k^A$ 和 $\epsilon_1^B, \epsilon_2^B, \ldots, \epsilon_k^B$, 则可用 k 折交叉验证 “成对 t 检验” (paired t-test) 来进行比较检验.

- 基本思想:
  - 对每对结果求差 $\Delta_i = \epsilon_i^A - \epsilon_i^B$; 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 $\Delta_1, \Delta_2, \ldots, \Delta_k$ 来对 “学习器 A 与 B 性能相同” 这个假设做 t 检验.

McNemar 检验: 两学习器比较

- 对二分类问题, 使用留出法不仅可以估计出学习器 A 和 B 的测试错误率, 还可获得两学习器分类结果的差别, 如 “列联表” (contingency table):

算法B	算法A	
	正确	错误
正确	e00	e01
错误	e10	e11

- 基本思想:
  - 若假设两学习器性能相同, 则应有 $e_{01} = e_{10}$, 那么 $|e_{01} - e_{10}|$ 应当服从正态分布.
  - McNemar 检验考虑变量 $\tau_{\chi^2} = \dfrac{(|e_{01} - e_{10}|)^2}{e_{01} + e_{10}}$ 服从自由度为 1 的 χ^2 分布.

- 2.1 经验误差与过拟合
- 2.2 评估方法
- 2.3 性能度量
- 2.4 比较检验
- **2.5 偏差与方差**

偏差-方差分解: 为什么具有这样的性能

- 偏差-方差分解 (bias-variance decomposition): 分析误差从何而来.
- 以回归任务为例:
  - 学习算法的期望预测为 $\bar{f}(\mathbf{x}) = \mathbb{E}_D[f(\mathbf{x}; D)]$
  - 期望误差与真实标记的差别称为偏差, 即 $bias^2(\mathbf{x}) = (\bar{f}(\mathbf{x}) - y)^2$
  - 使用样本数相同的不同训练集产生的方差为 $var(\mathbf{x}) = \mathbb{E}_D[(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2]$
  - 噪声为 $\epsilon^2 = \mathbb{E}_D[(y_D - y)^2]$
- 泛化误差可分解为 $E(f; D) = \mathbb{E}_D[(f(\mathbf{x}; D) - y_D)^2] = bias^2(x) + var(x) + \epsilon^2$, 即偏差、方差与噪声之和.

x	测试样本	yD	x 在数据集 D 中的标记
y	x 的真实标记	f(x; D)	由 D 学得模型 f 在 x 上的预测输出

偏差-方差分解: 为什么具有这样的性能

- 偏差 $bias^2(\mathbf{x}) = (\bar{f}(\mathbf{x}) - y)^2$
  - 学习算法本身的拟合能力
- 方差 $var(\mathbf{x}) = \mathbb{E}_D[(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2]$
  - 数据扰动所造成的影响
- 噪声 $\epsilon^2 = \mathbb{E}_D[(y_D - y)^2]$
  - 学习问题本身的难度
- 结论: 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.

偏差-方差窘境 (bias-variance dilemma)

- 一般来说, 偏差与方差存在冲突:
  - 训练不足时, 学习器拟合能力不强, 训练数据扰动不足以使学习器产生显著变化, 偏差主导.
  - 随着训练程度加深, 学习器拟合能力逐渐增强, 方差逐渐主导.
  - 训练程度充足后, 学习器拟合能力非常强, 训练数据发生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, 则发生过拟合.