



机器学习

第2章 模型评估与选择

朱桂祥 (9120201070@nufe.edu.cn)

南京财经大学信息工程学院

江苏省电子商务重点实验室

电子商务信息处理国家级国际联合研究中心

电子商务交易技术国家地方联合工程实验室

<https://github.com/zgx881205/Machine-Learning/tree/main>



南京财经大学

NANJING UNIVERSITY OF FINANCE & ECONOMICS

01 经验误差与过拟合

02 评估方法

03 性能度量

04 比较检验

01 经验误差与过拟合

02 评估方法

03 性能度量

04 比较检验

经验误差与过拟合

4

数据预处理



经验/数据

归纳/训练

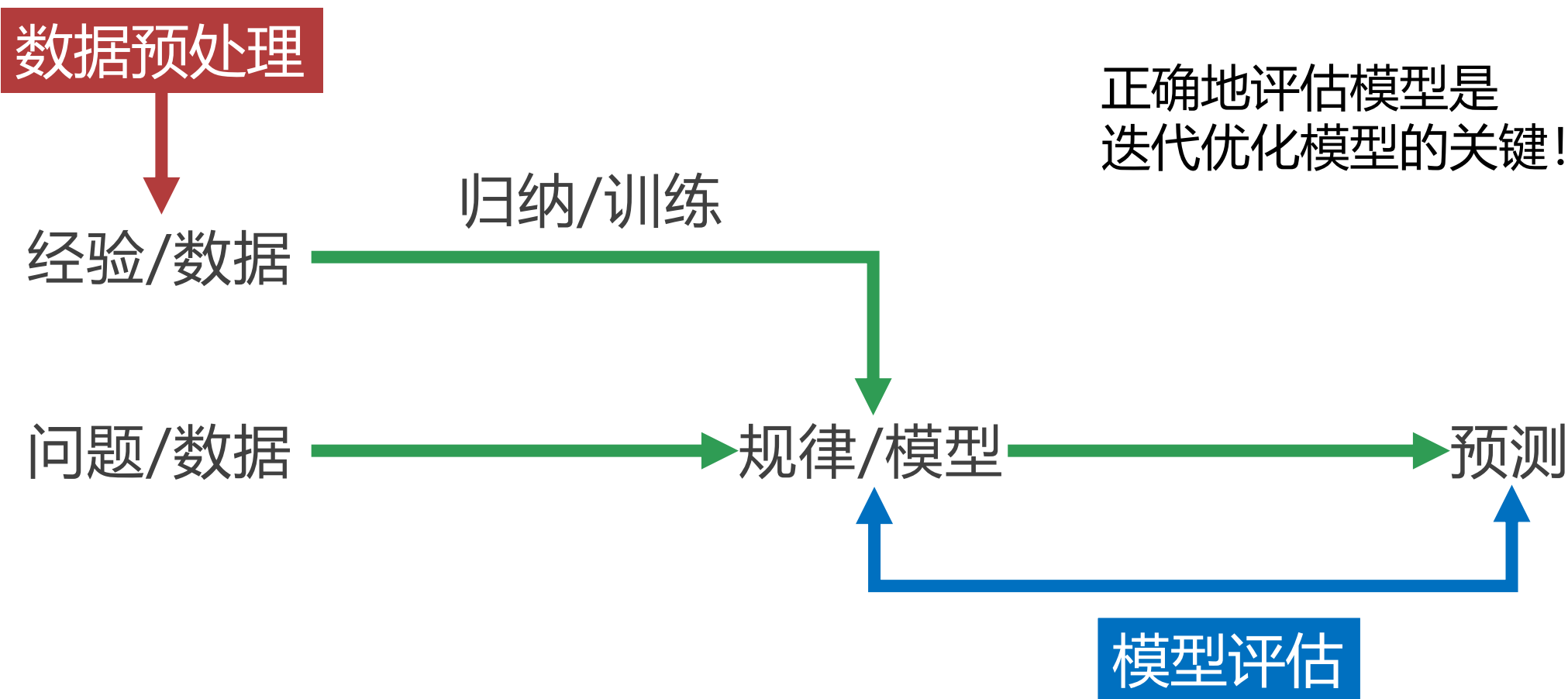
问题/数据

规律/模型

预测

正确地评估模型是
迭代优化模型的关键！

模型评估



经验误差与过拟合

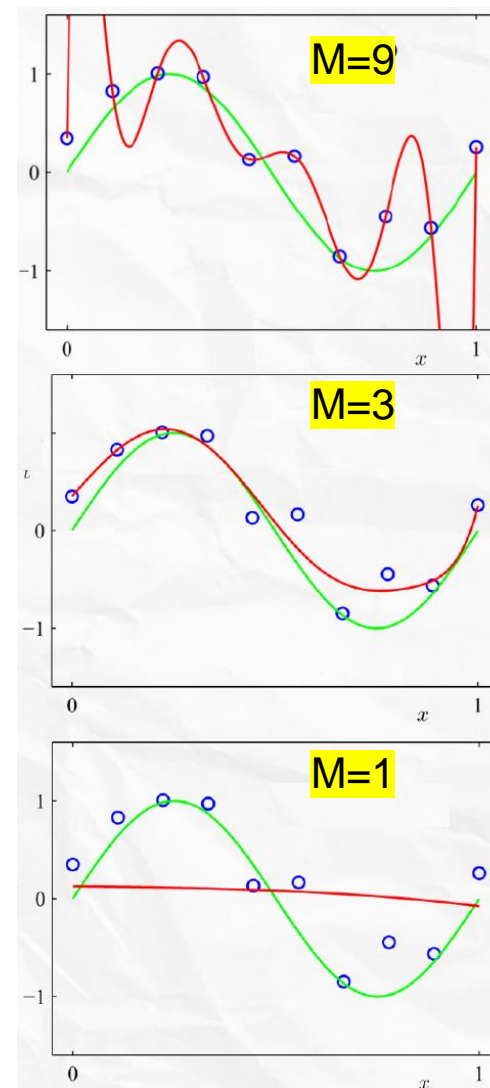
5

- 经验误差

训练数据 \longrightarrow 训练误差

- 测试误差/泛化误差

测试数据 \longrightarrow 测试误差



经验误差与过拟合

6

泛化误差：在“未来”样本上的误差

经验误差：在训练集上的误差，亦称“训练误差”

- 泛化误差越小越好
- 经验误差是否越小越好？

NO! 因为会出现“过拟合”(overfitting)

经验误差与过拟合

7



过拟合：学习器把训练样本本身特点当做所有潜在样本都会具有的一般性质。

欠拟合：训练样本的一般性质尚未被学习器学好。

过拟合、欠拟合的直观类比

经验误差与过拟合

8

- ✓ **过拟合**：学习器把训练样本学习的“太好”，将训练样本本身的特点当做所有样本的一般性质，导致泛化性能下降
 - ✓ 优化目标加正则项
 - ✓ early stop
- ✓ **欠拟合**：对训练样本的一般性质尚未学好
 - ✓ 决策树：拓展分支
 - ✓ 神经网络：增加训练轮数




01 经验误差与过拟合

02 评估方法

03 性能度量

04 比较检验

三个关键问题：

- 如何获得测试结果？  评估方法
- 如何评估性能优劣？  性能度量
- 如何判断实质差别？  比较检验

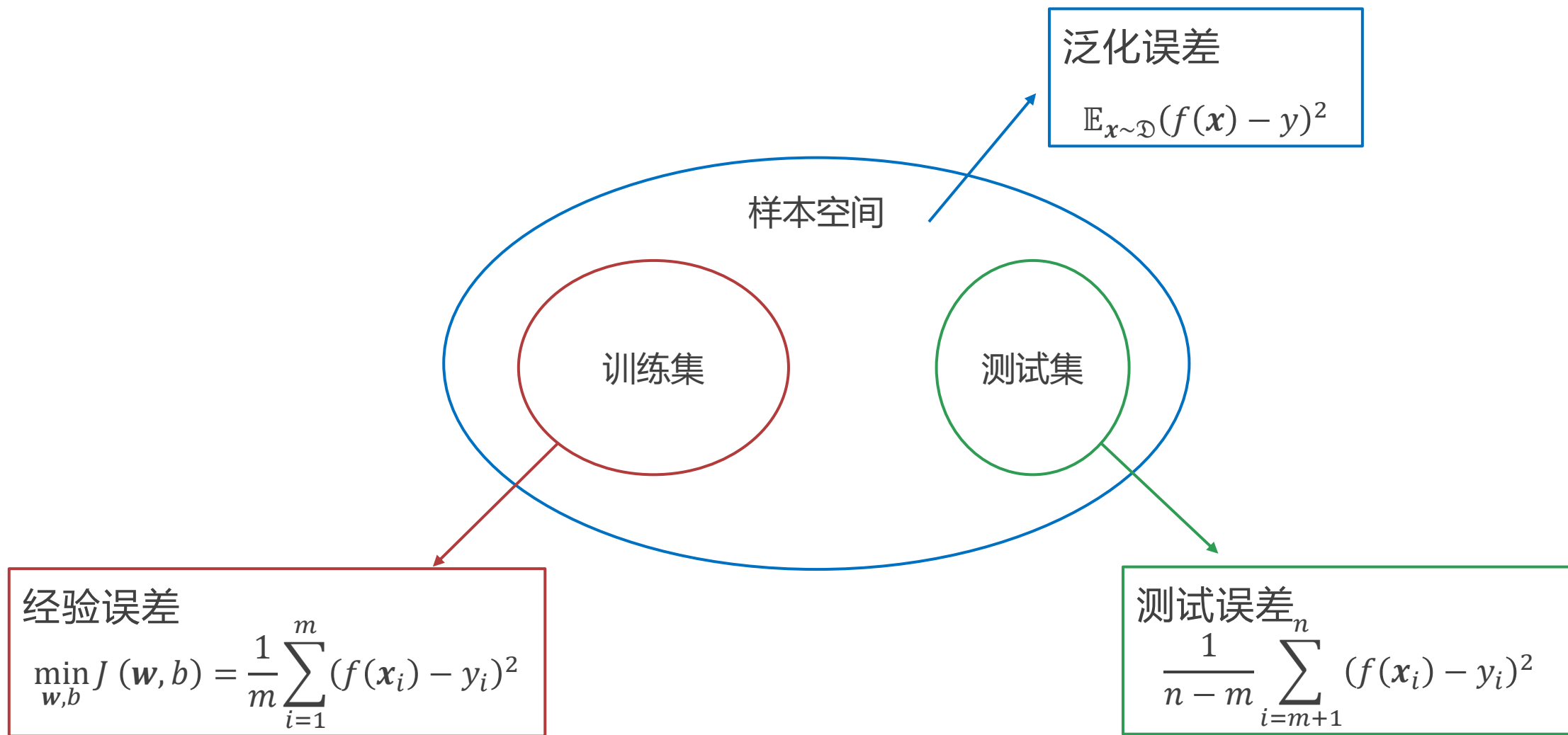
评估方法

11

- ✓ 现实任务中往往会对学习器的**泛化性能**、时间开销、存储开销、可解释性等方面的因素进行评估并做出选择。
- ✓ 我们假设测试集是从样本真实分布中独立采样获得，将测试集上的“**测试误差**”作为泛化误差的近似，所以测试集要和训练集中的样本尽量**互斥**。

评估方法

12



留出法

13

✓ 通常将包含个 m 样本的数据集 $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$ 拆分成训练集 S 和测试集 T 。

✓ 留出法：

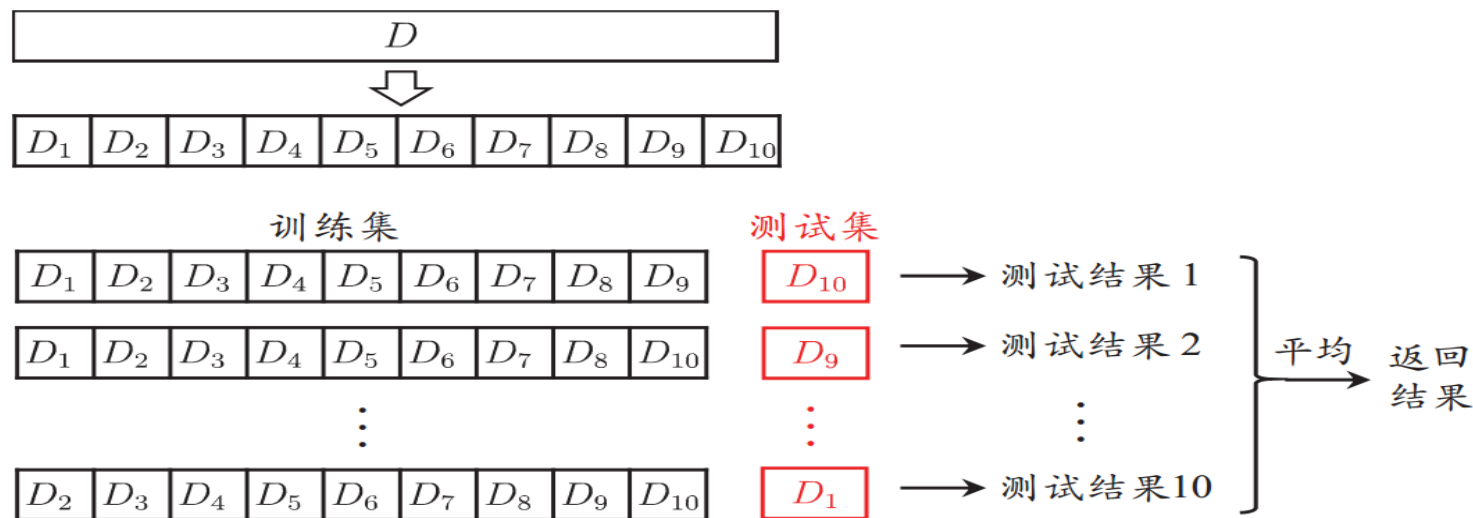
- ✓ 直接将数据集划分为两个互斥集合
- ✓ 训练/测试集划分要尽可能保持数据分布的一致性
- ✓ 一般若干次随机划分、重复实验取平均值
- ✓ 训练/测试样本比例通常为2:1~4:1



K-折交叉验证

14

- ✓ **交叉验证法**：将数据集划分为 k 个大小相似的**互斥**子集，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**，最终返回 k 个测试结果的均值， k 最常用的取值是10。



10 折交叉验证示意图

K-折交叉验证

15

- ✓ 与留出法类似，将数据集 D 划分为 k 个子集同样存在多种划分方式，为了减小因样本划分不同而引入的差别， k 折交叉验证通常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 折交叉验证结果的均值，例如常见的“10次10折交叉验证”。
- ✓ 假设数据集 D 包含 m 个样本，若令 $k = m$ ，则得到留一法：
 - ✓ 不受随机样本划分方式的影响
 - ✓ 结果往往比较准确
 - ✓ 当数据集比较大时，计算开销难以忍受

评估方法-调参

16

✓ 参数

✓ 算法的参数：一般由人工设定，亦称“超参数”

学习率 η

✓ 模型的参数：一般由学习确定

模型参数 w, b



评估方法

17

训练集：课上知识

验证集：课后习题

测试集：期末考试



训练集相当于上课学知识

验证集相当于课后的的练习题，用来纠正和强化学到的知识

测试集相当于期末考试，用来最终评估学习效果

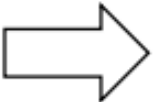

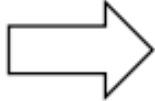
01 经验误差与过拟合

02 评估方法

03 性能度量

04 比较检验

三个关键问题：

- 如何获得测试结果？  评估方法
- 如何评估性能优劣？  性能度量
- 如何判断实质差别？  比较检验

性能度量

20

- ✓ 性能度量是**衡量模型泛化能力**的评价标准，反映了任务需求；使用不同的性能度量往往会导致不同的评判结果。
- ✓ 在预测任务中，给定样例集 $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$ ，评估学习器 f 的性能也即**把预测结果 $f(\mathbf{x})$ 和真实标记 y 比较**。

性能度量

21

- ✓ 回归任务最常用的性能度量是 “均方误差”

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

✓ 分类任务最常用的性能度量：

✓ **错误率**：分错样本占样本总数的比例

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

✓ **精度**：分对样本占样本总数的比率

$$acc(f; D) = \frac{1}{m} \sum_{i=1}^m \mathbb{I}(f(\mathbf{x}_i) = y_i)$$

性能度量

23

- ✓ Web搜索等场景中经常需要衡量预测出来的正例中正确的比率或者正例被预测出来的比率，此时查准率和查全率比错误率和精度更适合。



预测用户感兴趣的信息



预测用户不感兴趣的信息

性能度量

24

- ✓ 统计真实标记和预测结果的组合可以得到 “**混淆矩阵**”。

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)

查准率 $P = \frac{TP}{TP + FP}$

查全率 $R = \frac{TP}{TP + FN}$

性能度量

预测用户感兴趣的信息-正例



FP

TP

TP

TP

FP

预测用户不感兴趣的信息-反例



TN

TN

TN

FN

TN

真实情况	预测结果	
	正例	反例
正例	3	1
反例	2	4

性能度量

26

- ✓ 查准率和查全率通常是一堆**矛盾**的度量
 - ✓ 为了提高查准率，可以只查出最具把握的结果，但难免漏掉其他好的结果，使得查全率较低
 - ✓ 为了提高查全率，可以返回所有结果，但查准率会降低

我们希望检索结果Precision越高越好，同时Recall也越高越好，但事实上这两者在某些情况下有矛盾的。比如极端情况下，我们只搜索出了一个结果，且是准确的，那么Precision就是100%，但是Recall就很低；而如果我们把所有结果都返回，那么比如Recall是100%，但是Precision就会很低。

因此在不同的场合中需要自己判断希望Precision比较高或是Recall比较高。

P-R曲线

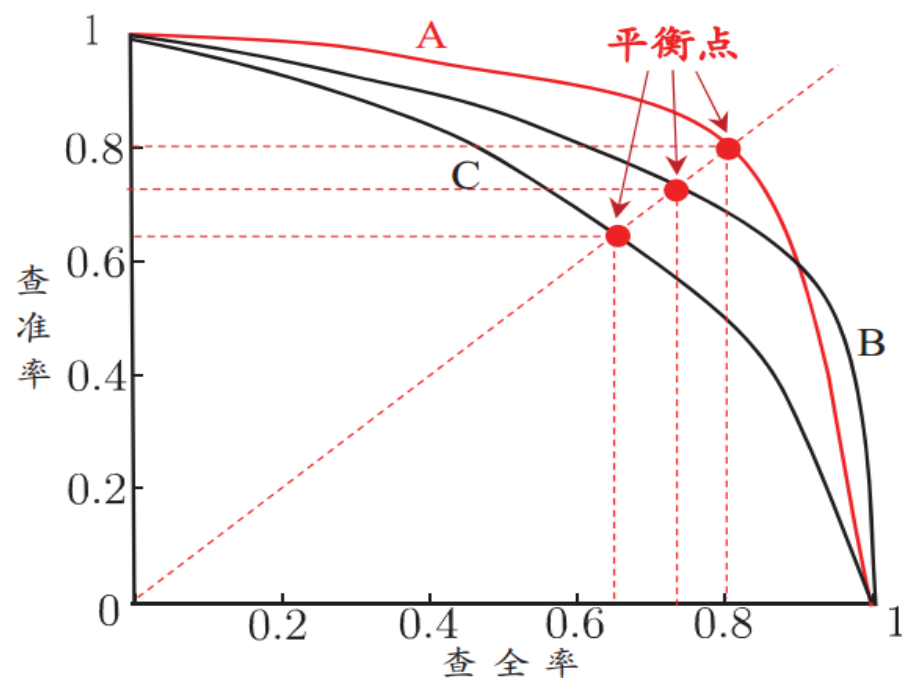
28

- ✓ 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，则可以得到查准率-查全率曲线，简称“P-R曲线”

平衡点是曲线上“查准率=查全率”时的取值，可用来用于度量P-R曲线有交叉的分类器性能高低

注意：A的P-R曲线完全包住C，则学习器A要优于学习器C；

A的P-R曲线与B交叉，则难以一般性断言孰优孰劣。



P-R曲线与平衡点示意图

<https://www.jianshu.com/p/61dae5cd2420>

F1 度量

29

- ✓ 比P-R曲线平衡点更常用的是F1度量:

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

- ✓ 比F1更一般的形式

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

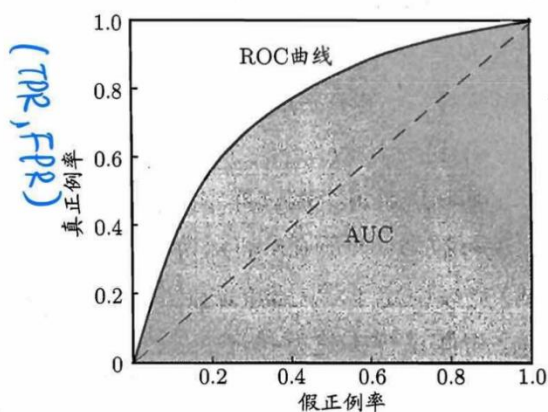
$\beta > 1$ 偏重查全率(逃犯信息检索)

$\beta < 1$ 偏重查准率(商品推荐系统)

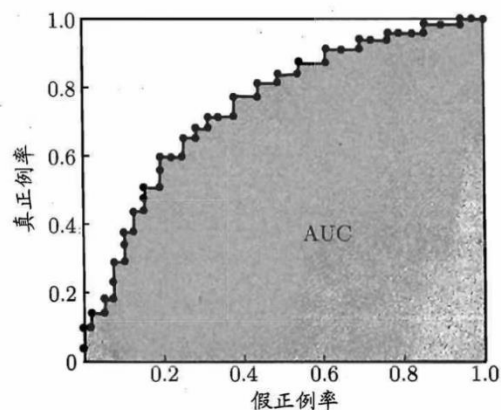
ROC与AUC

30

- 若某个学习器的ROC曲线被另一个学习器的曲线“包住”，则后者性能优于前者；否则如果曲线交叉，可以根据ROC曲线下面积大小进行比较，也即AUC值。



(a) ROC 曲线与 AUC



(b) 基于有限样例绘制的 ROC 曲线与 AUC

假设ROC曲线由 $\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ 的点按序连接而形成 ($x_1 = 0, x_m = 1$)，则：

AUC可估算为：

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

AUC衡量了样本预测的排序质量。

代价敏感错误率

31

- ✓ 现实任务中不同类型的错误所造成的后果很可能不同，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**”。
- ✓ 以二分类为例，可根据领域知识设定“代价矩阵”，如 $cost_{ij}$ 表示将第 i 类样本预测为第 j 类样本的代价。损失程度越大， $cost_{01}$ 与 $cost_{10}$ 值的差别越大。

代价敏感错误率

32

- ✓ 在非均等代价下，不再最小化错误次数，而是最小化“总体代价”，则“代价敏感”错误率相应的为：

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




01 经验误差与过拟合

02 评估方法

03 性能度量

04 比较检验

三个关键问题：

- 如何获得测试结果？  评估方法
- 如何评估性能优劣？  性能度量
- 如何判断实质差别？  比较检验

在某种度量下取得评估结果后，是否可以直接比较以评判优劣？

NO!

因为：

- 测试性能不等于泛化性能
- 测试性能随着测试集的变化而变化
- 很多机器学习算法本身有一定的随机

机器学习 “概率近似正确”

比较检验

36

统计假设检验：为学习器性能提供了重要依据

✓ 两学习器比较

- ✓ 交叉验证t检验（基于成对t检验）
 - ✓ k折交叉验证：5x2交叉验证
- ✓ McNemar检验（基于列联表，卡方检验）

✓ 多学习器比较

- ✓ Friedman+Nemenyi
 - ✓ Friedman检验（基于序值，F检验；判断是否都相同）
 - ✓ Nemenyi后续检验（基于序值，进一步判断两两差别）



谢谢观赏 下节课见

