三、线性模型

主讲教师：周志华

对率回归

以对率函数为联系函数：

$$y = \frac{1}{1+e^{-z}} \quad \text{变为} \quad y = \frac{1}{1+e^{-(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b)}}$$

即：

$$\ln \frac{y}{1-y} = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$$

几率(odds)，反映了 $\boldsymbol{x}$ 作为正例的相对可能性

“对数几率”
(log odds, 亦称 logit)

“对数几率回归”(logistic regression)
简称“对率回归”

- 无需事先假设数据分布
- 可得到“类别”的近似概率预测
- 可直接应用现有数值优化算法求取最优解

注意：它是
分类学习算法！

求解思路

若将 y 看作类后验概率估计 $p(y=1 \mid \boldsymbol{x})$，则

$$\ln \frac{y}{1-y} = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b \quad \text{可写为} \quad \ln \frac{p(y=1 \mid \boldsymbol{x})}{p(y=0 \mid \boldsymbol{x})} = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$$

于是，可使用“极大似然法” → 第7章
(maximum likelihood method)

给定数据集 $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$

最大化“对数似然”(log-likelihood)函数

$$\ell(\boldsymbol{w}, b) = \sum_{i=1}^{m} \ln p(y_i \mid \boldsymbol{x}_i; \boldsymbol{w}, b)$$

求解思路

令 $\boldsymbol{\beta}=(\boldsymbol{w};b)$, $\hat{\boldsymbol{x}}=(\boldsymbol{x};1)$，则 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b$ 可简写为 $\boldsymbol{\beta}^{\mathrm{T}}\hat{\boldsymbol{x}}$

再令 $p_1\left(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta}\right)=p\left(y=1\mid\hat{\boldsymbol{x}};\boldsymbol{\beta}\right)=\dfrac{e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}}{1+e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}}$

$$p_0\left(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta}\right)=p\left(y=0\mid\hat{\boldsymbol{x}};\boldsymbol{\beta}\right)=1-p_1\left(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta}\right)=\frac{1}{1+e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}}$$

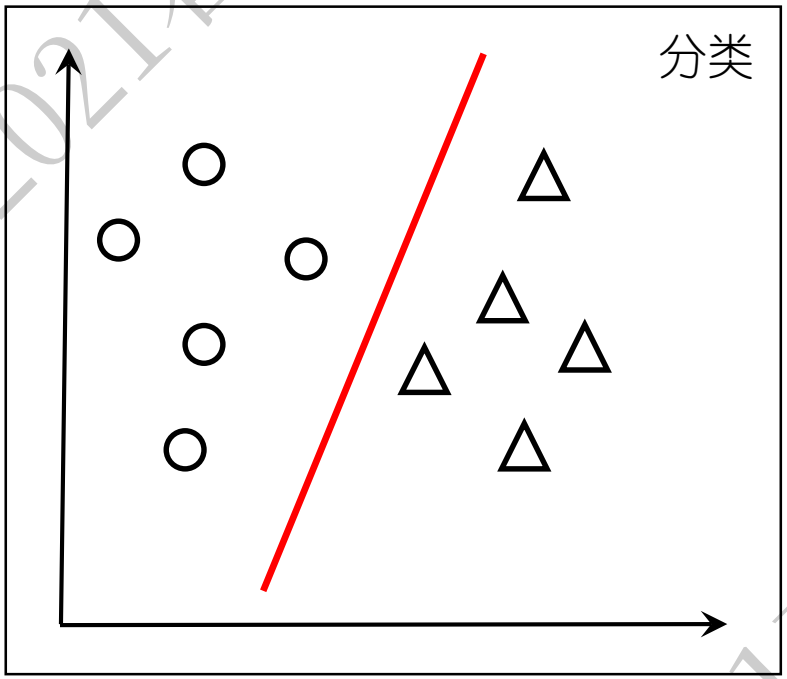
则似然项可重写为 $p\left(y_i\mid\boldsymbol{x}_i;\boldsymbol{w}_i,b\right)=y_ip_1\left(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta}\right)+\left(1-y_i\right)p_0\left(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta}\right)$

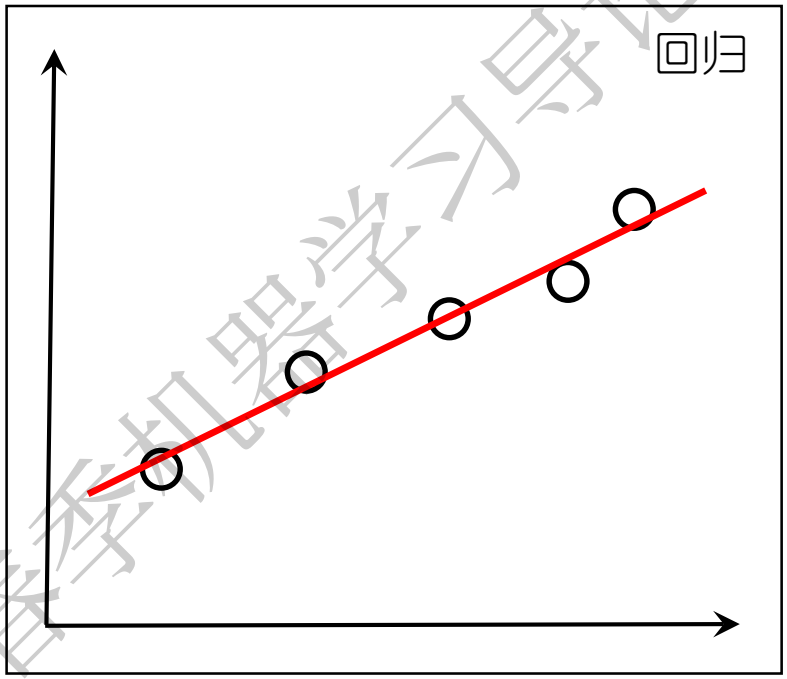
于是，最大化似然函数 $\ell(\boldsymbol{w},b)=\sum_{i=1}^{m}\ln p(y_i\mid\boldsymbol{x}_i;\boldsymbol{w},b)$

等价为最小化 $\ell\left(\boldsymbol{\beta}\right)=\sum_{i=1}^{m}\left(-y_i\boldsymbol{\beta}^{\mathrm{T}}\hat{\boldsymbol{x}}_i+\ln\left(1+e^{\beta^{\mathrm{T}}\hat{\boldsymbol{x}}_i}\right)\right)$

高阶可导连续凸函数，可用经典的数值优化方法
如梯度下降法/牛顿法 [Boyd and Vandenberghe, 2004]

线性模型做“分类”



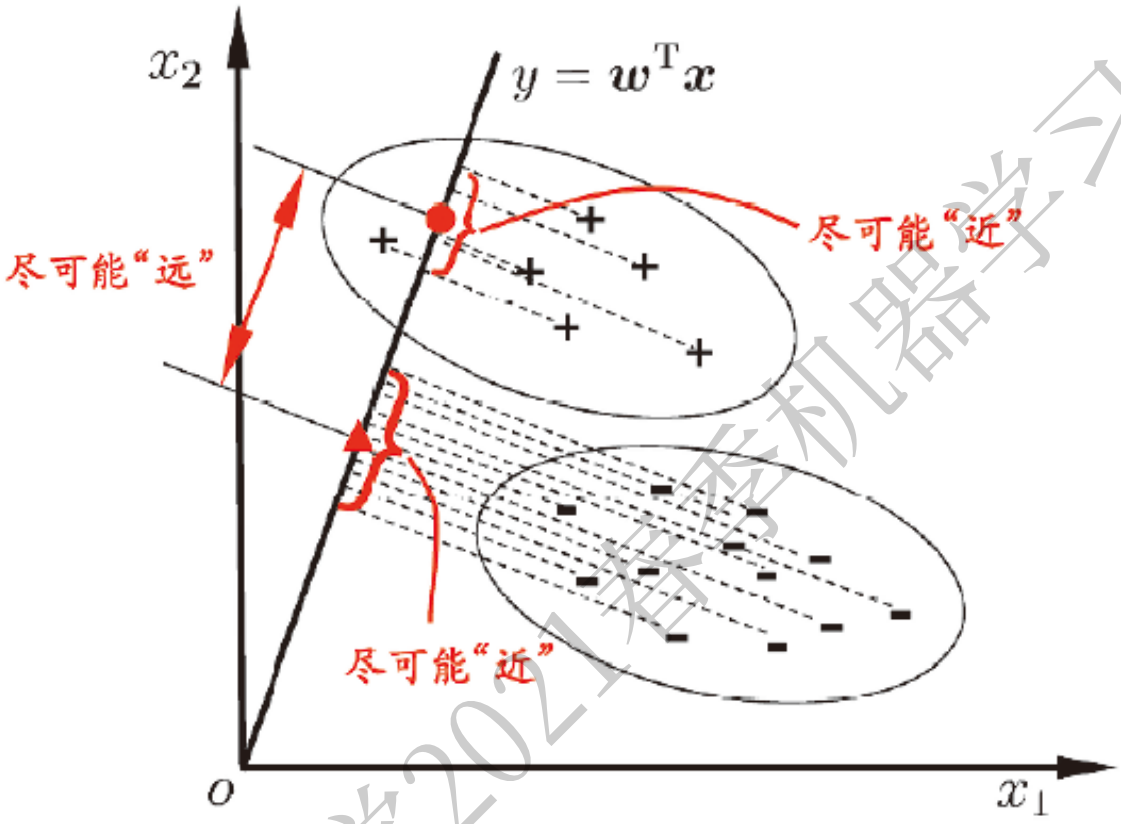


如何“直接”做分类？

广义线性模型；
通过“联系函数”

例如，对率回归

线性判别分析 (Linear Discriminant Analysis)



由于将样例投影到一条直线（低维空间），因此也被视为一种“监督降维”技术

降维 ⟶ 第10章

LDA的目标

给定数据集 $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$

第 i 类示例的集合 X_i

第 i 类示例的均值向量 $\boldsymbol{\mu}_i$

第 i 类示例的协方差矩阵 $\boldsymbol{\Sigma}_i$

两类样本的中心在直线上的投影：$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0$ 和 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1$

两类样本的协方差：$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_0\boldsymbol{w}$ 和 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_1\boldsymbol{w}$

同类样例的投影点尽可能接近 → $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_0\boldsymbol{w} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_1\boldsymbol{w}$ 尽可能小

异类样例的投影点尽可能远离 → $\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0 - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1\|_2^2$ 尽可能大

于是，最大化

$$J = \frac{\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0 - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1\|_2^2}{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_0\boldsymbol{w} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_1\boldsymbol{w}} = \frac{\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1)\boldsymbol{w}}$$

LDA的目标

类内散度矩阵 (within-class scatter matrix)

$$\begin{aligned}\mathbf{S}_w &= \boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1 \\ &= \sum_{\boldsymbol{x}\in X_0} (\boldsymbol{x}-\boldsymbol{\mu}_0)(\boldsymbol{x}-\boldsymbol{\mu}_0)^{\mathrm{T}} + \sum_{\boldsymbol{x}\in X_1} (\boldsymbol{x}-\boldsymbol{\mu}_1)(\boldsymbol{x}-\boldsymbol{\mu}_1)^{\mathrm{T}}\end{aligned}$$

类间散度矩阵 (between-class scatter matrix)

$$\mathbf{S}_b = (\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^{\mathrm{T}}$$

LDA的目标：最大化广义瑞利商 (generalized Rayleigh quotient)

$$J = \frac{\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_b\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w}}$$

$\boldsymbol{w}$ 成倍缩放不影响 J 值
仅考虑方向

求解思路

令 $\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w}=1$，最大化广义瑞利商等价形式为

$$\min_{\boldsymbol{w}} \ -\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_b\boldsymbol{w}$$

$$\text{s.t. } \boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w}=1$$

运用拉格朗日乘子法，有 $\mathbf{S}_b\boldsymbol{w}=\lambda\mathbf{S}_w\boldsymbol{w}$

由 $\mathbf{S}_b$ 定义，有 $\mathbf{S}_b\boldsymbol{w}=(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{w}$

注意到 $(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{w}$ 是标量，令其等于 λ

于是 $\boldsymbol{w}=\mathbf{S}_w^{-1}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)$

实践中通常是进行奇异值分解 $\mathbf{S}_w=\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{T}}$ → 附录 A

然后 $\mathbf{S}_w^{-1}=\mathbf{V}\boldsymbol{\Sigma}^{-1}\mathbf{U}^{\mathrm{T}}$

推广到多类

假定有 N 个类

- 全局散度矩阵 $\mathbf{S}_t = \mathbf{S}_b + \mathbf{S}_w = \sum_{i=1}^{m} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T$
- 类内散度矩阵 $\mathbf{S}_w = \sum_{i=1}^{N} \mathbf{S}_{w_i} \quad \mathbf{S}_{w_i} = \sum_{\boldsymbol{x} \in X_i} (\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^T$
- 类间散度矩阵 $\mathbf{S}_b = \mathbf{S}_t - \mathbf{S}_w = \sum_{i=1}^{N} m_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T$

多分类LDA有多种实现方法：采用 $\mathbf{S}_b$, $\mathbf{S}_w$, $\mathbf{S}_t$ 中的任何两个

例如， $\max_{\mathbf{W}} \frac{\mathrm{tr}\left(\mathbf{W}^{\mathrm{T}} \mathbf{S}_b \mathbf{W}\right)}{\mathrm{tr}\left(\mathbf{W}^{\mathrm{T}} \mathbf{S}_w \mathbf{W}\right)} \Rightarrow \mathbf{S}_b \mathbf{W} = \lambda \mathbf{S}_w \mathbf{W}$

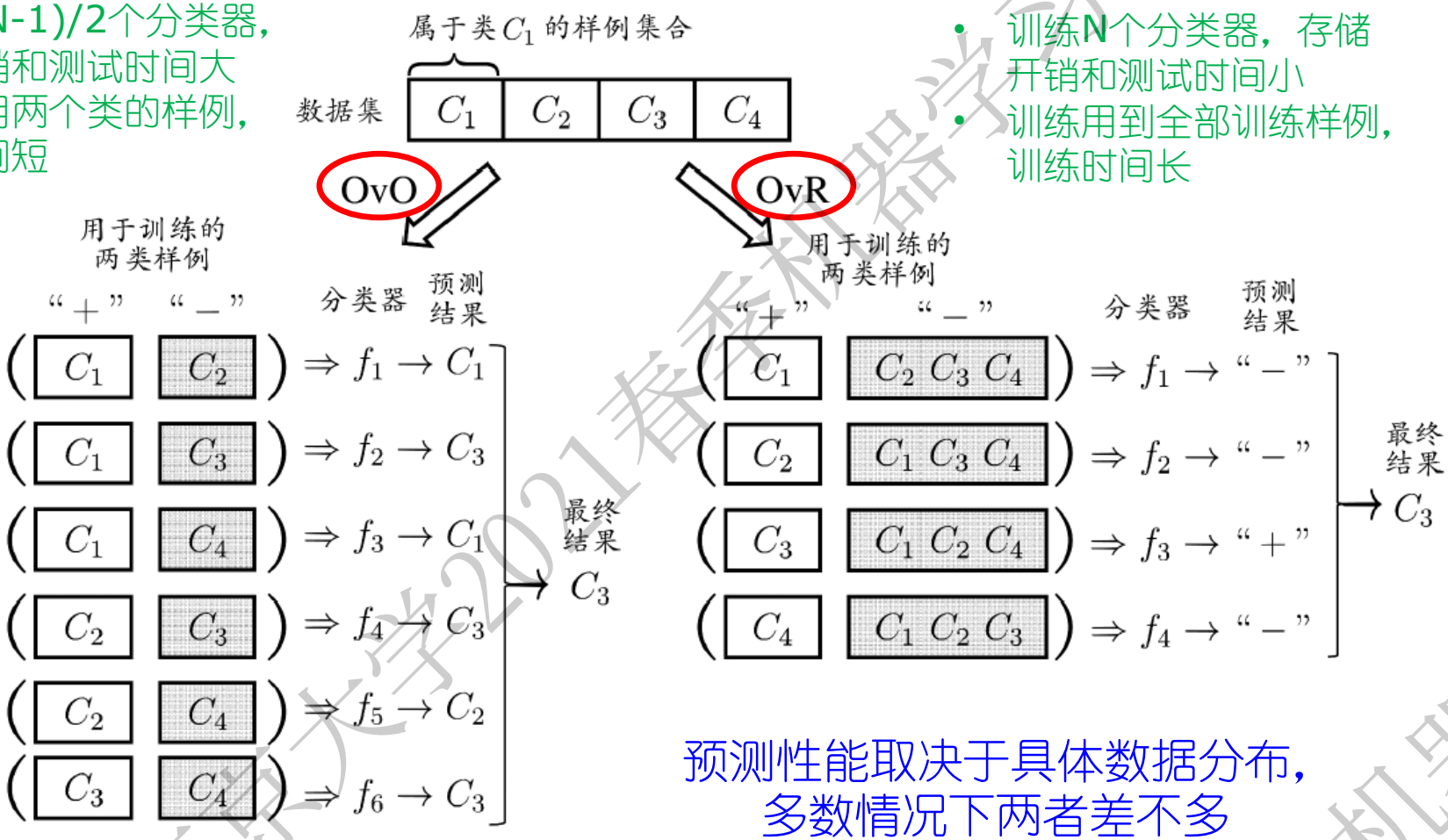
$\mathbf{W} \in \mathbb{R}^{d \times (N-1)}$

$\mathbf{W}$的闭式解是 $\mathbf{S}_w^{-1}\mathbf{S}_b$的 d' ($\leq N$-1) 个最大非零广义特征值对应的特征向量组成的矩阵

多分类学习

拆解法：将一个多分类任务拆分为若干个二分类任务求解

- 训练N(N-1)/2个分类器，存储开销和测试时间大
- 训练只用两个类的样例，训练时间短



- 训练N个分类器，存储开销和测试时间小
- 训练用到全部训练样例，训练时间长

预测性能取决于具体数据分布，多数情况下两者差不多

纠错输出码 (ECOC)

多对多(Many vs Many, MvM)：将若干类作为正类，若干类作为反类

一种常见方法：纠错输出码 (Error Correcting Output Code)

编码：对 N 个类别做 M 次划分，每次将一部分类别划为正类，一部分划为反类

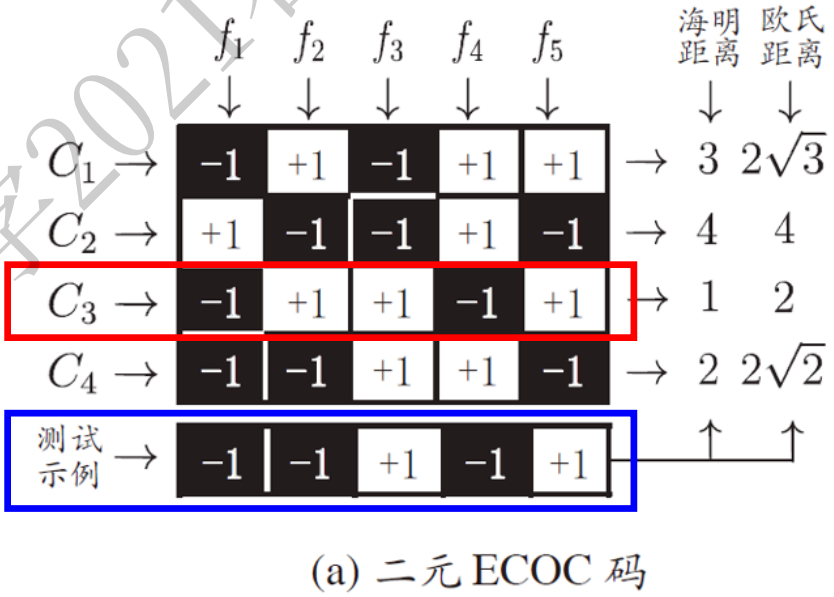
M 个二类任务；
(原)每类对应一个长为 M 的编码

距离最小的类为最终结果

解码：测试样本交给 M 个分类器预测

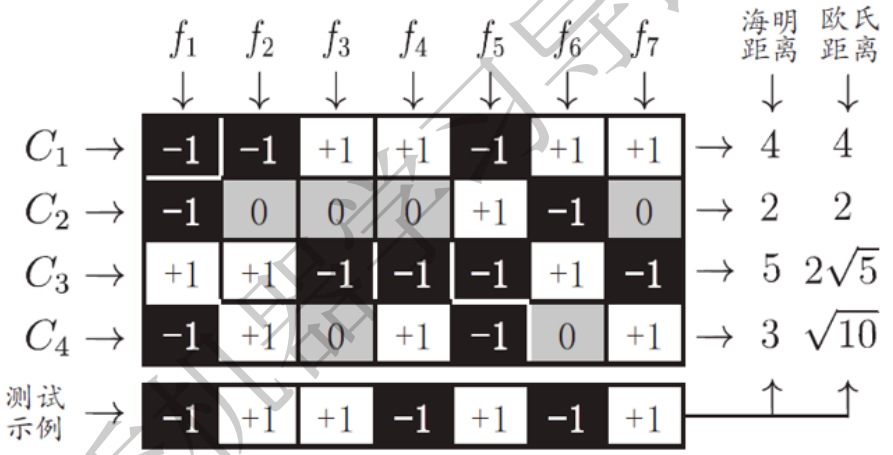
长为 M 的预测结果编码

纠错输出码



(a) 二元 ECOC 码

[Dietterich and Bakiri,1995]



(b) 三元 ECOC 码

[Allwein et al. 2000]

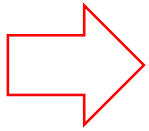
- ECOC编码对分类器错误有一定容忍和修正能力，编码越长、纠错能力越强
- 对同等长度的编码，理论上来说，任意两个类别之间的编码距离越远，则纠错能力越强

类别不平衡 (class-imbalance)

不同类别的样本比例相差很大；“小类”往往更重要

基本思路：

若 $\frac{y}{1-y} > 1$ 则 预测为正例.



若 $\frac{y}{1-y} > \frac{m^+}{m^-}$ 则 预测为正例.

基本策略

——“再缩放” (rescaling)：

$$\frac{y'}{1-y'} = \frac{y}{1-y} \times \frac{m^-}{m^+}$$

然而，精确估计 m^-/m^+ 通常很困难！

常见类别不平衡学习方法：

- 过采样 (oversampling)
  - 例如：SMOTE
- 欠采样 (undersampling)
  - 例如：EasyEnsemble
- 阈值移动 (threshold-moving)

前往第四站……

四、决策树

主讲教师：周志华

决策树模型

决策树基于“树”结构进行决策

- 每个“内部结点”对应于某个属性上的“测试”（test）
- 每个分支对应于该测试的一种可能结果（即该属性的某个取值）
- 每个“叶结点”对应于一个“预测结果”

学习过程：通过对训练样本的分析来确定“划分属性”（即内部结点所对应的属性）

预测过程：将测试示例从根结点开始，沿着划分属性所构成的“判定测试序列”下行，直到叶结点

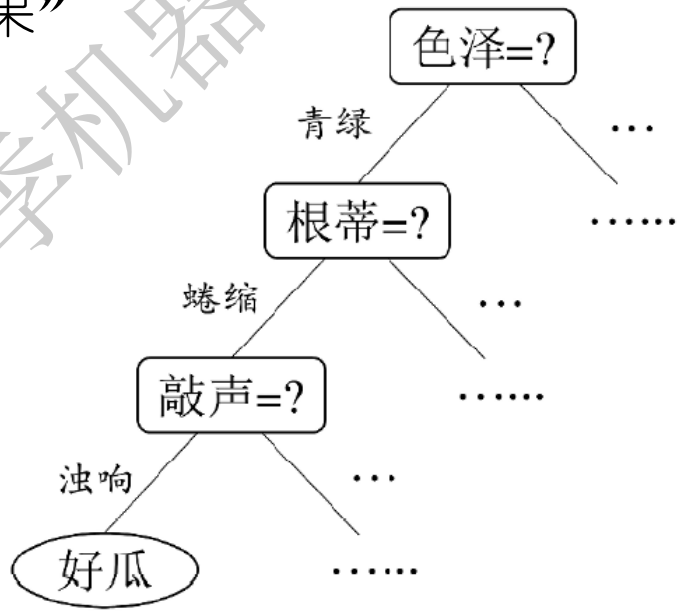


图 4.1 西瓜问题的一棵决策树

基本流程

策略：“分而治之”(divide-and-conquer)

自根至叶的递归过程

在每个中间结点寻找一个“划分”(split or test)属性

三种停止条件：

(1) 当前结点包含的样本全属于同一类别，无需划分；

(2) 当前属性集为空，或是所有样本在所有属性上取值相同，无法划分；

(3) 当前结点包含的样本集合为空，不能划分.

基本算法

输入: 训练集 $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$;
属性集 $A = \{a_1, a_2, \ldots, a_d\}$.

过程: 函数 TreeGenerate(D, A)

```
1: 生成结点 node;
2: if D 中样本全属于同一类别 C then
3:     将 node 标记为 C 类叶结点; return
4: end if
5: if A = ∅ OR D 中样本在 A 上取值相同 then
6:     将 node 标记为叶结点, 其类别标记为 D 中样本数最多的类; return
7: end if
8: 从 A 中选择最优划分属性 a_*;
9: for a_* 的每一个值 a_*^v do
10:    为 node 生成一个分支; 令 D_v 表示 D 中在 a_* 上取值为 a_*^v 的样本子集;
11:    if D_v 为空 then
12:        将分支结点标记为叶结点, 其类别标记为 D 中样本最多的类; return
13:    else
14:        以 TreeGenerate(D_v, A\{a_*}) 为分支结点
15:    end if
16: end for
```

输出: 以 node 为根结点的一棵决策树

递归返回，情形(1)

递归返回，情形(2)

利用当前结点的后验分布

递归返回，情形(3)

将父结点的样本分布作为当前结点的先验分布

决策树算法的核心

信息增益 (information gain)

信息熵 (entropy) 是度量样本集合“纯度”最常用的一种指标

假定当前样本集合 D 中第 k 类样本所占的比例为 p_k，则 D 的信息熵定义为

$$\mathrm{Ent}(D) = -\sum_{k=1}^{|\mathcal{Y}|} p_k \log_2 p_k$$

计算信息熵时约定：若 $p = 0$，则 $p \log_2 p = 0$.

$\mathrm{Ent}(D)$ 的最小值为 0，最大值为 $\log_2 |\mathcal{Y}|$.

$\mathrm{Ent}(D)$ 的值越小，则 D 的纯度越高

信息增益直接以信息熵为基础，计算当前划分对信息熵所造成的变化

信息增益

离散属性 a 的取值：$\{a^1, a^2, \ldots, a^V\}$

D^v：D 中在 a 上取值 $= a^v$ 的样本集合

以属性 a 对 数据集 D 进行划分所获得的信息增益为：

$$\mathrm{Gain}(D, a) = \mathrm{Ent}(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} \mathrm{Ent}(D^v)$$

划分前的信息熵

划分后的信息熵

第 v 个分支的权重，
样本越多越重要

ID3算法中使用

一个例子

表 4.1 西瓜数据集 2.0

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

该数据集包含17个训练样例，$|\mathcal{Y}| = 2$，其中正例占 $p_1 = \frac{8}{17}$ 反例占 $p_2 = \frac{9}{17}$

根结点的信息熵为

$$\mathrm{Ent}(D) = -\sum_{k=1}^{2} p_k \log_2 p_k = -\left(\frac{8}{17}\log_2\frac{8}{17} + \frac{9}{17}\log_2\frac{9}{17}\right) = 0.998$$

一个例子 (续)

以属性“色泽”为例，其对应的3个子集分别为：

D^1(色泽=青绿)

D^2(色泽=乌黑)

D^3(色泽=浅白)

表 4.1 西瓜数据集 2.0

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

对D^1(色泽=青绿)，正例3/6，反例3/6

于是：

$$\mathrm{Ent}(D^1) = -(\tfrac{3}{6}\log_2\tfrac{3}{6} + \tfrac{3}{6}\log_2\tfrac{3}{6}) = 1.000$$

一个例子 (续)

表 4.1 西瓜数据集 2.0

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

D^2(色泽=乌黑)，
正例4/6，反例2/6

$$\mathrm{Ent}(D^2) = -\left(\frac{4}{6}\log_2\frac{4}{6} + \frac{2}{6}\log_2\frac{2}{6}\right) = 0.918$$

D^3(色泽=浅白)，
正例1/5，反例4/5

$$\mathrm{Ent}(D^3) = -\left(\frac{1}{5}\log_2\frac{1}{5} + \frac{4}{5}\log_2\frac{4}{5}\right) = 0.722$$

于是，属性“色泽”的信息增益为

$$\begin{aligned}\mathrm{Gain}(D, \text{色泽}) &= \mathrm{Ent}(D) - \sum_{v=1}^{3}\frac{|D^v|}{|D|}\mathrm{Ent}(D^v) \\ &= 0.998 - \left(\frac{6}{17}\times 1.000 + \frac{6}{17}\times 0.918 + \frac{5}{17}\times 0.722\right) = 0.109\end{aligned}$$

一个例子 (续)

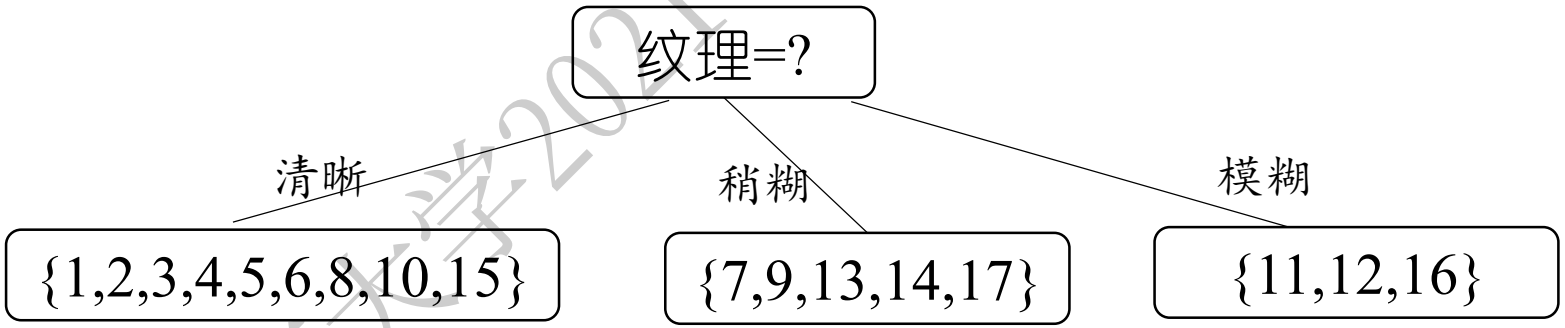
类似的，其他属性的信息增益为

$$\text{Gain}(D, \text{根蒂}) = 0.143 \qquad \text{Gain}(D, \text{敲声}) = 0.141$$

$$\text{Gain}(D, \text{纹理}) = 0.381 \qquad \text{Gain}(D, \text{脐部}) = 0.289$$

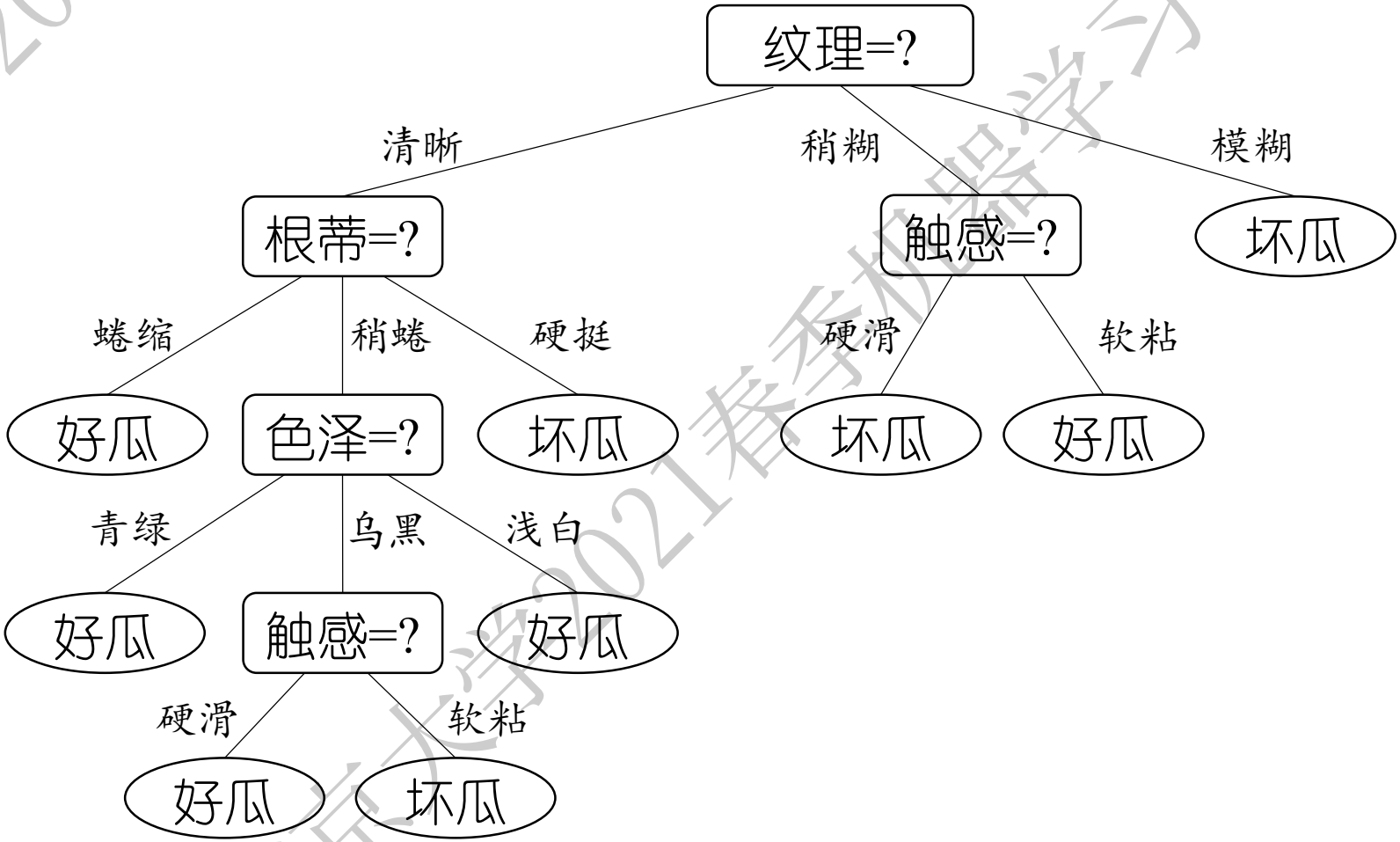
$$\text{Gain}(D, \text{触感}) = 0.006$$

属性“纹理”的信息增益最大，被选为划分属性



一个例子 (续)

对每个分支结点做进一步划分，最终得到决策树



决策树简史

- 第一个决策树算法：CLS (Concept Learning System)

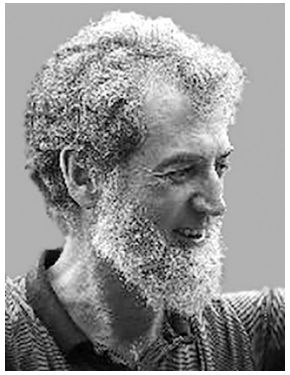
  [E. B. Hunt, J. Marin, and P. T. Stone's book "*Experiments in Induction*" published by Academic Press in 1966]

- 使决策树受到关注、成为机器学习主流技术的算法：ID3

  [J. R. Quinlan's paper in a book "*Expert Systems in the Micro Electronic Age*" edited by D. Michie, published by Edinburgh University Press in 1979]

- 最常用的决策树算法：C4.5

  [J. R. Quinlan's book "*C4.5: Programs for Machine Learning*" published by Morgan Kaufmann in 1993]



J. Ross Quinlan
(1943 -)

决策树简史(con't)

- 可以用于回归任务的决策树算法：CART (Classification and Regression Tree)

  [L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone's book "Classification and Regression Trees" published by Wadsworth in 1984]

- 基于决策树的最强大算法之一：RF (Random Forest)

  [L. Breiman's MLJ'01 paper "Random Forest"]

  这是一种"集成学习"方法→第 8 章

Leo Breiman
(1928-2005)