

十一、特征选择与稀疏学习

主讲教师：詹德川

特征

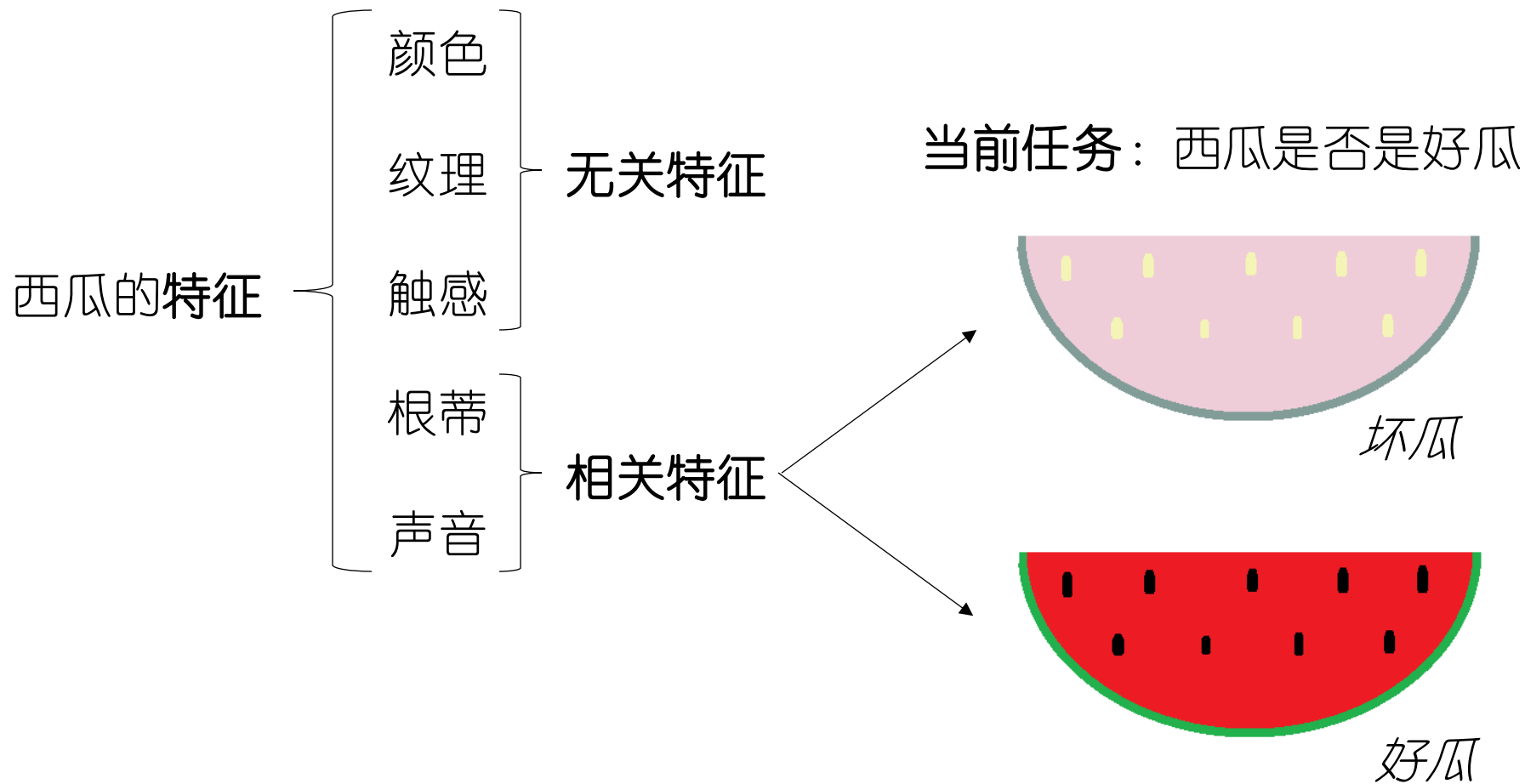
□ 特征

- 描述物体的属性

□ 特征的分类

- 相关特征：对**当前学习任务**有用的属性
- 无关特征：与**当前学习任务**无关的属性
- 冗余特征*：其所包含信息能由其他特征推演出来

例子：西瓜的特征



特征选择

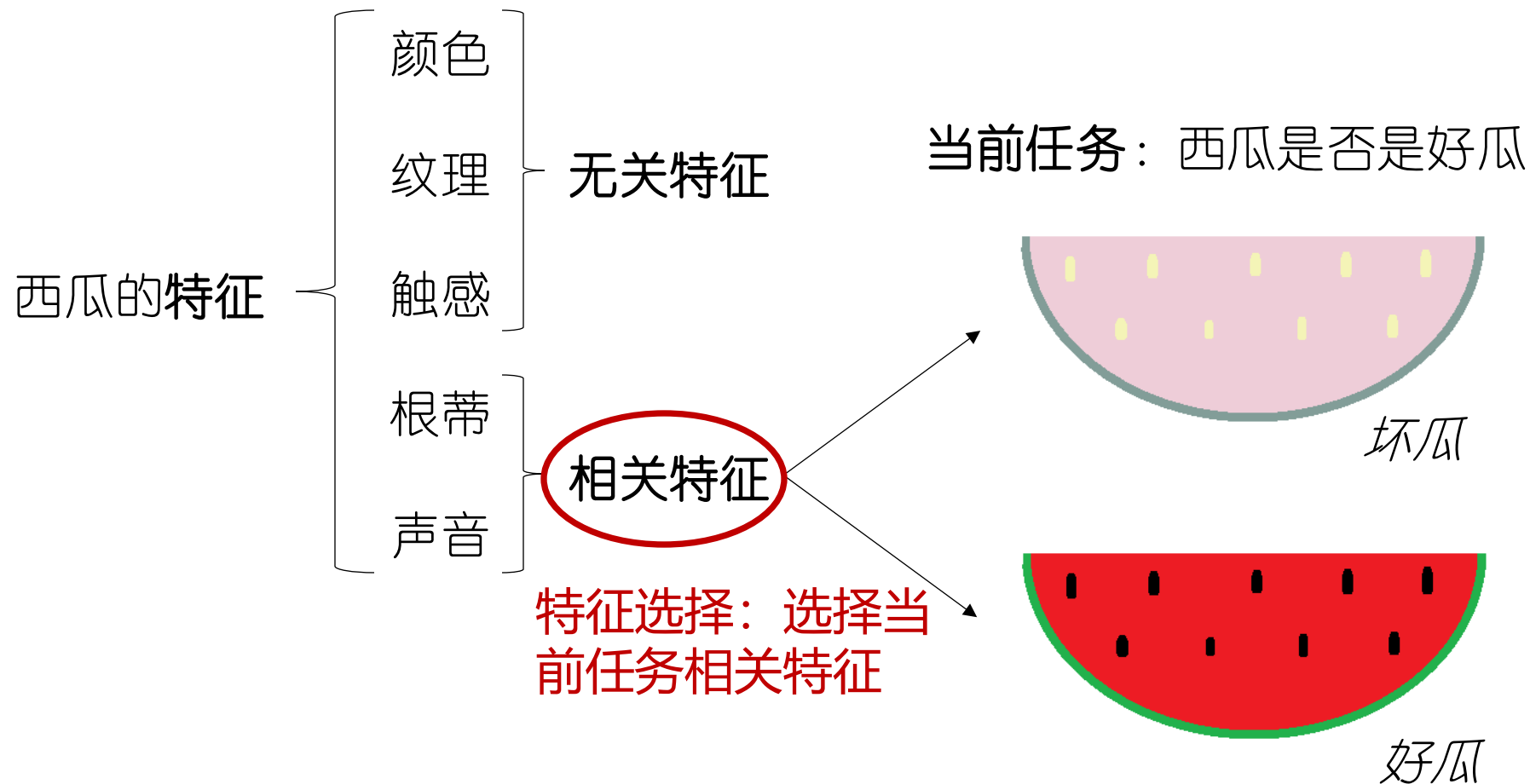
□ 特征选择

- 从给定的特征集合中选出**任务相关**特征子集
- 必须确保不丢失重要特征

□ 原因

- 减轻维度灾难：在少量属性上构建模型
- 降低学习难度：留下关键信息

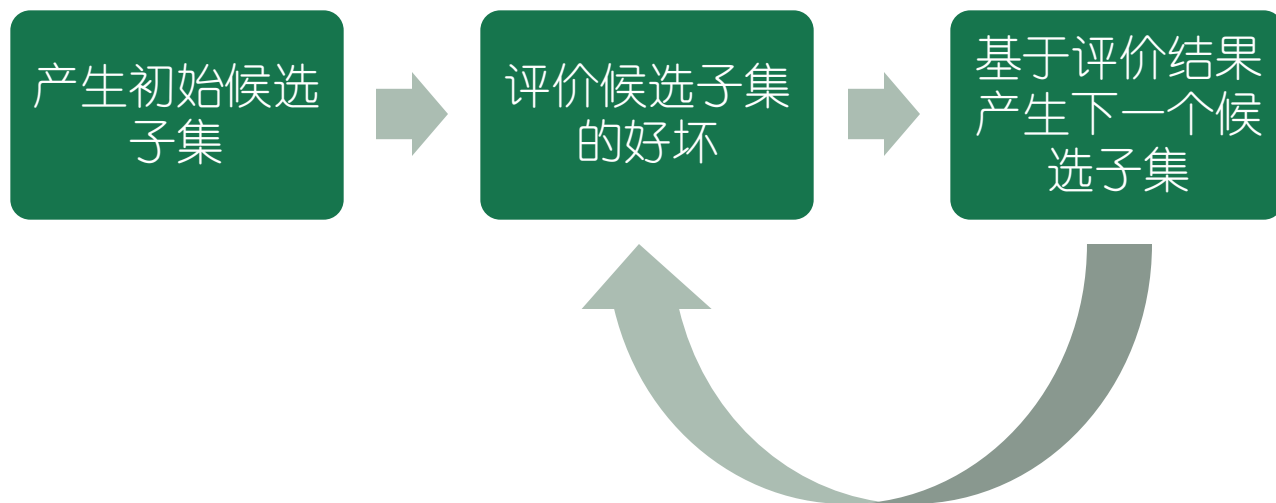
例子：判断是否好瓜时的特征选择



特征选择的一般方法

- 遍历所有可能的子集
 - 计算上遭遇组合爆炸，不可行

- 可行方法



两个关键环节：子集搜索和子集评价

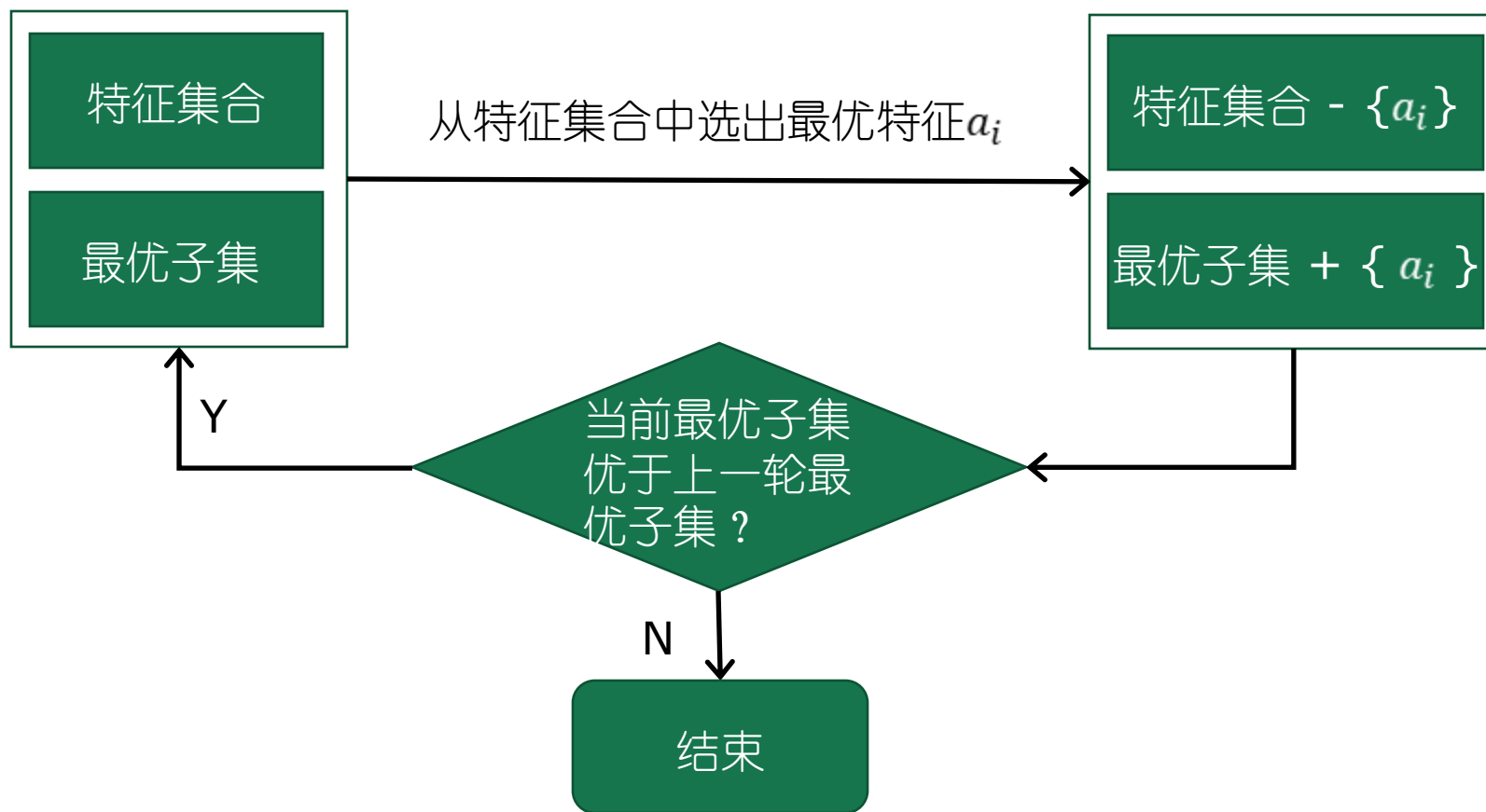
子集搜索

用贪心策略选择包含重要信息的特征子集

- 前向搜索：逐渐增加相关特征
- 后向搜索：从完整的特征集合开始，逐渐减少特征
- 双向搜索：每一轮逐渐增加相关特征，同时减少无关特征

前向搜索

- 最优子集初始为空集，特征集合初始时包括所有给定特征



子集搜索

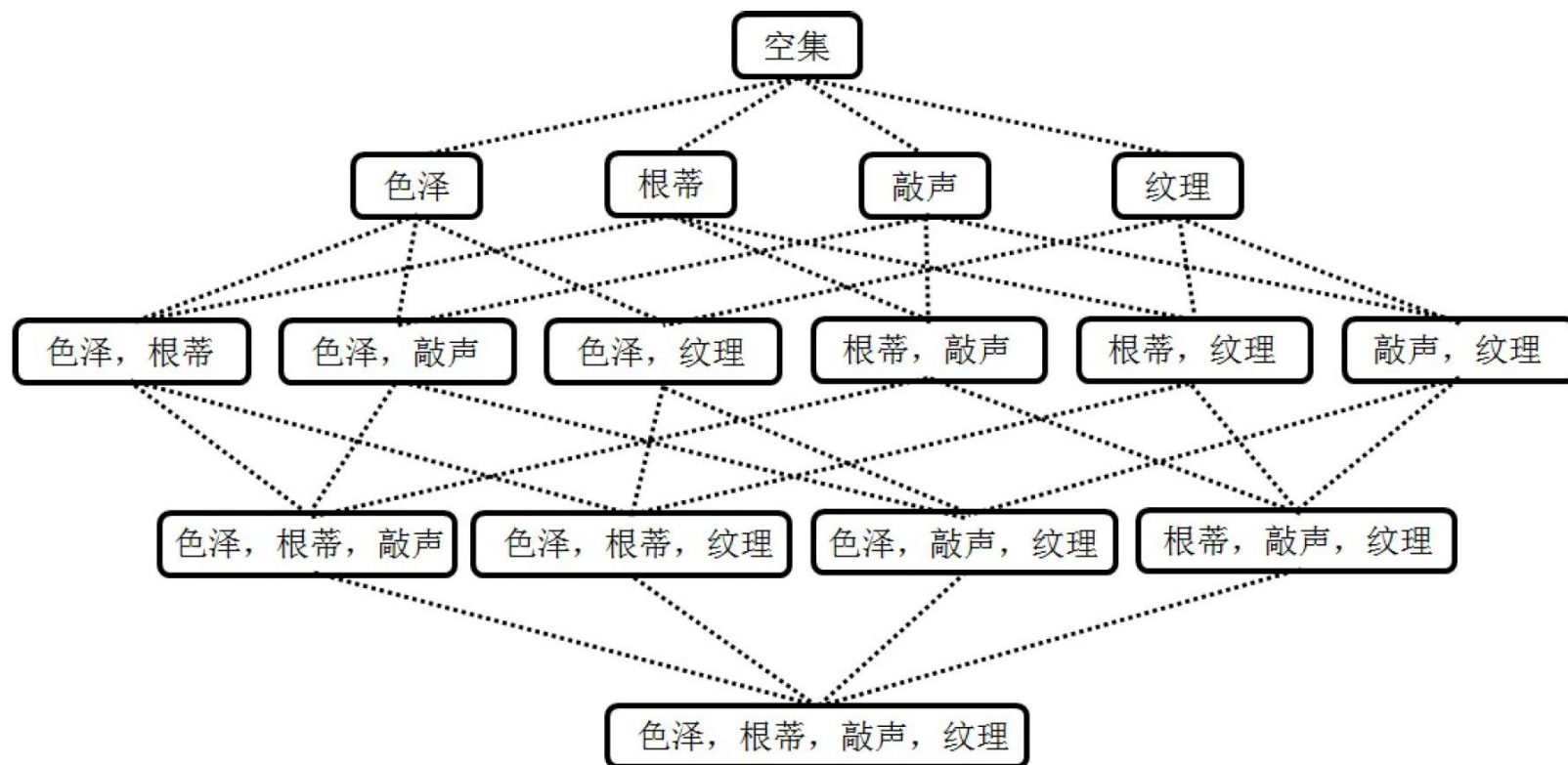


图 1.1 由4个属性及在属性子集上的交并运算定义的格

子集评价

- 特征子集确定了对数据集的一个划分
 - 每个划分区域对应着特征子集的某种取值
- 样本标记对应着对数据集的真实划分

通过估算这两个划分的差异，就能对特征子集进行评价；与样本标记对应的划分的差异越小，则说明当前特征子集越好

用信息熵进行子集评价

- 特征子集 A 确定了对数据集 D 的一个划分
 - A 上的取值将数据集 D 分为 V 份，每一份用 D^v 表示
 - $\text{Ent}(D^v)$ 表示 D^v 上的信息熵
- 样本标记 Y 对应着对数据集 D 的真实划分
 - $\text{Ent}(D)$ 表示 D 上的信息熵

D 上的信息熵定义为

$$\text{Ent}(D) = - \sum_{i=1}^{|y|} p_k \log_2 p_k$$

第 i 类样本所占比例为 p_i

特征子集 A 的信息增益为：

$$\text{Gain}(A) = \text{Ent}(D) - \sum_{v=1}^V \frac{|D^v|}{|D|} \text{Ent}(D^v)$$

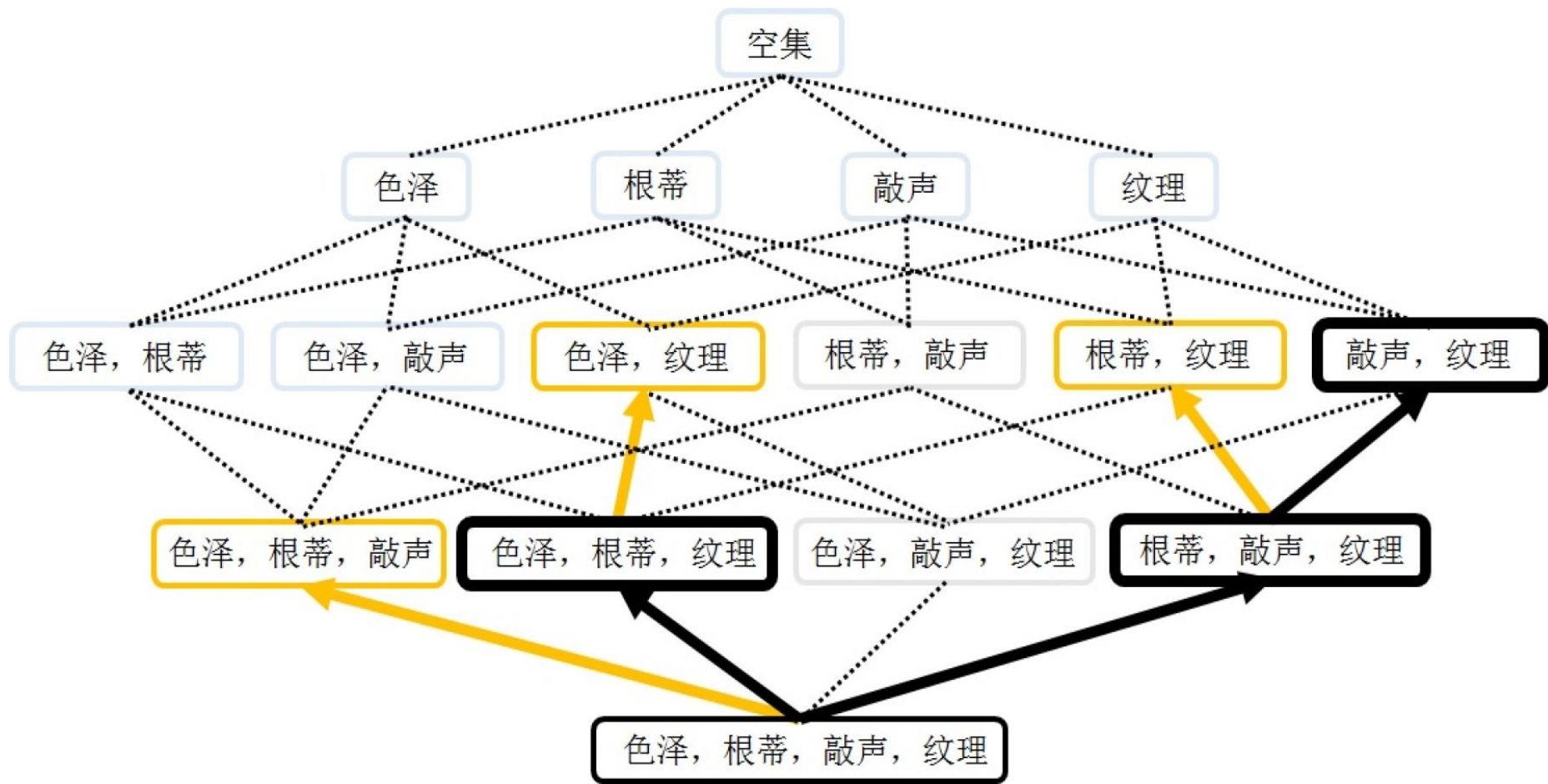


图 1.3 分支限界策略在格上的搜索路径：黑色箭头指向该策略每一步选取的属性子集（粗黑线框标注出每层去掉一个属性后可行的选择结果）；黄色箭头表示每一步尝试但不成功的搜索方向（黄色方框框出了在每次尝试后决定剪枝的属性子集）；蓝色框中的属性子集无需评价（已经被剪枝或一定会被其他分支搜索）

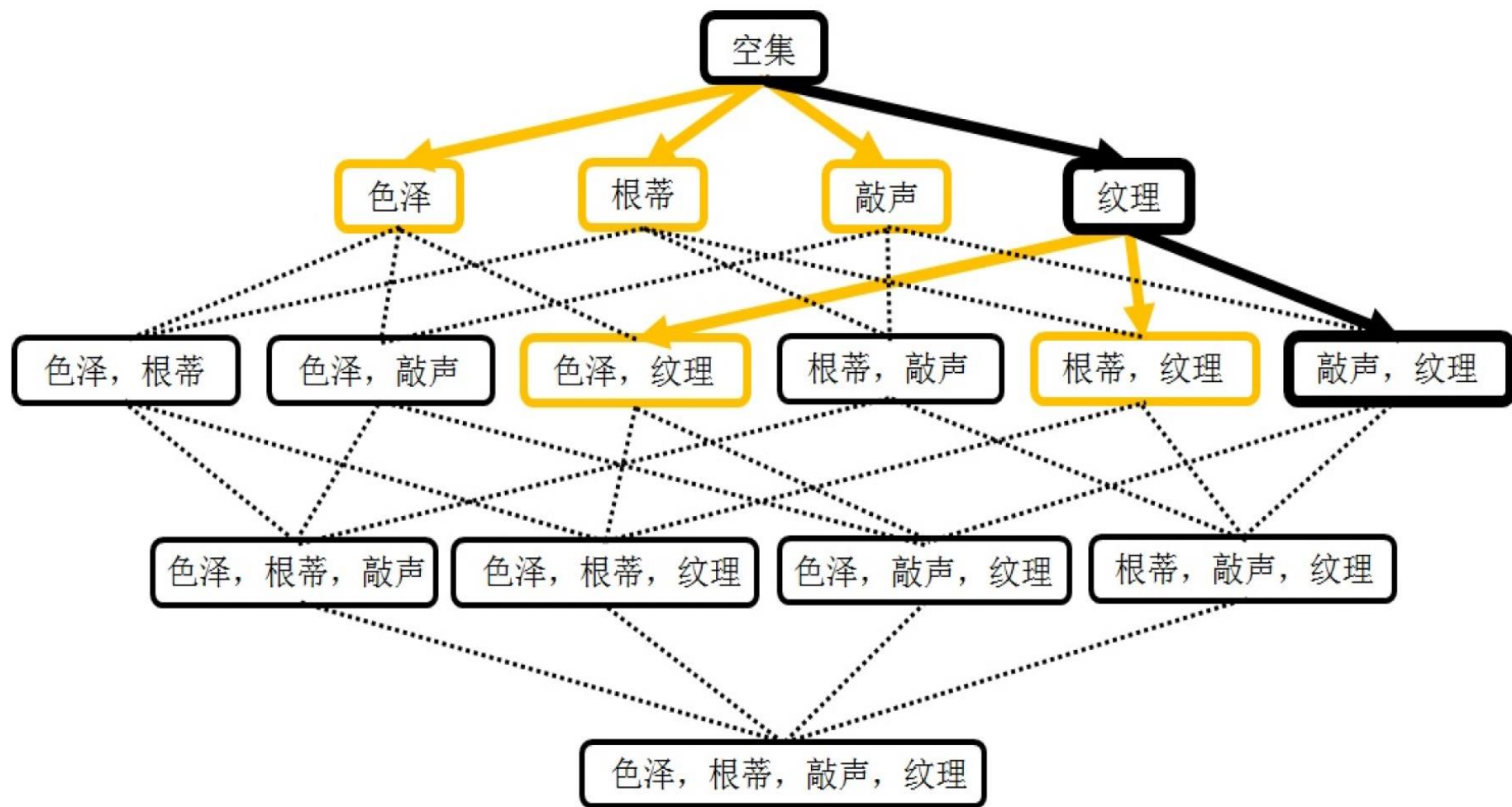


图 1.4 贪心搜索策略在格上的搜索路径：黑色箭头标注出该策略每一步选取的属性子集；黄色箭头表示每一步尝试的搜索方向；粗黑线框标注出最终的选择结果；黄色线框出了策略未选中的属性子集；其他框中的属性子集无需进行评估。

常见的特征选择方法

将特征子集搜索机制与子集评价机制相结合，即可得到特征选择方法

常见的特征选择方法大致分为如下三类：

- 过滤式

- 包裹式

- 嵌入式

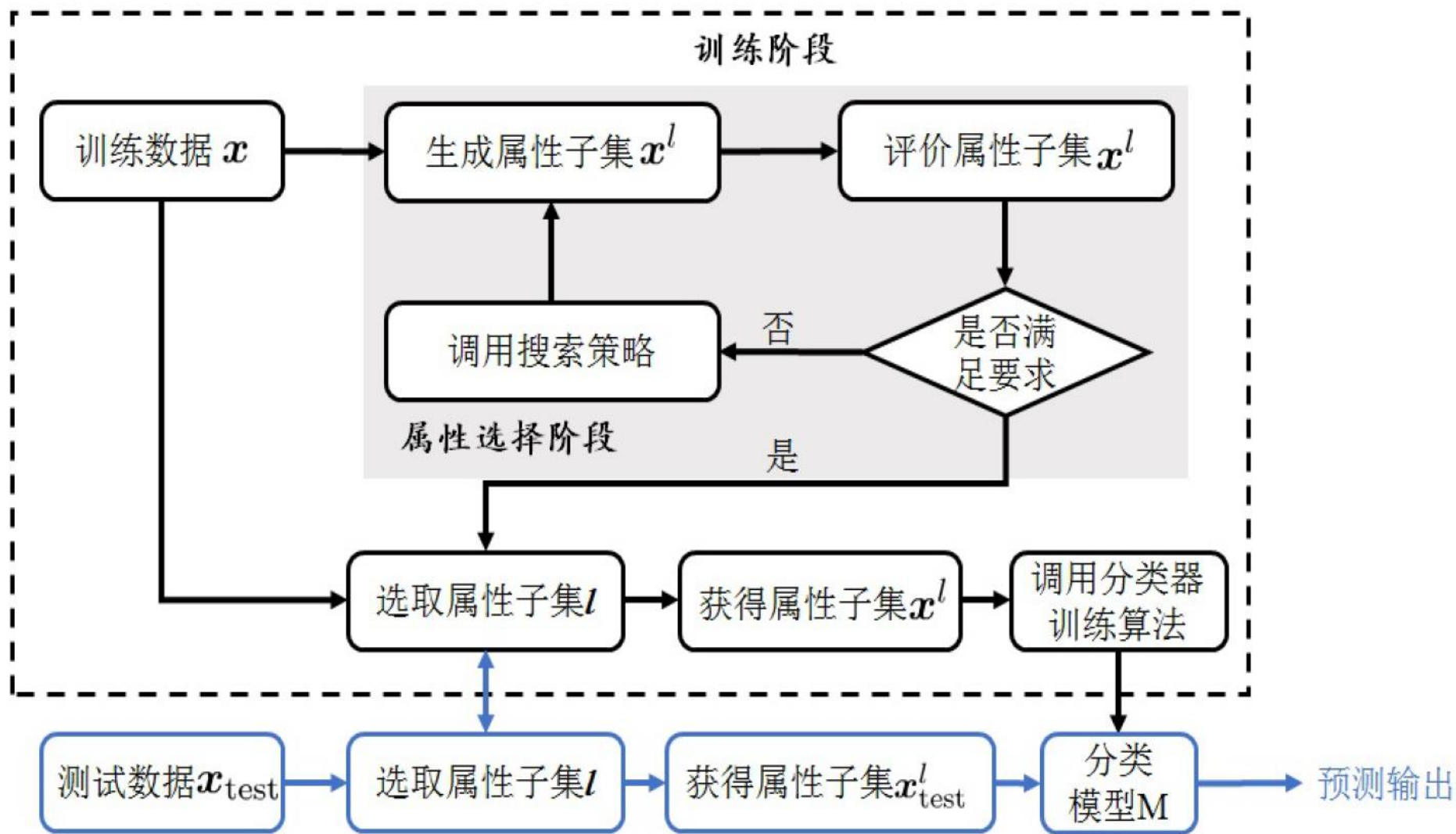


图 1.5 Filter式属性选择的一般流程

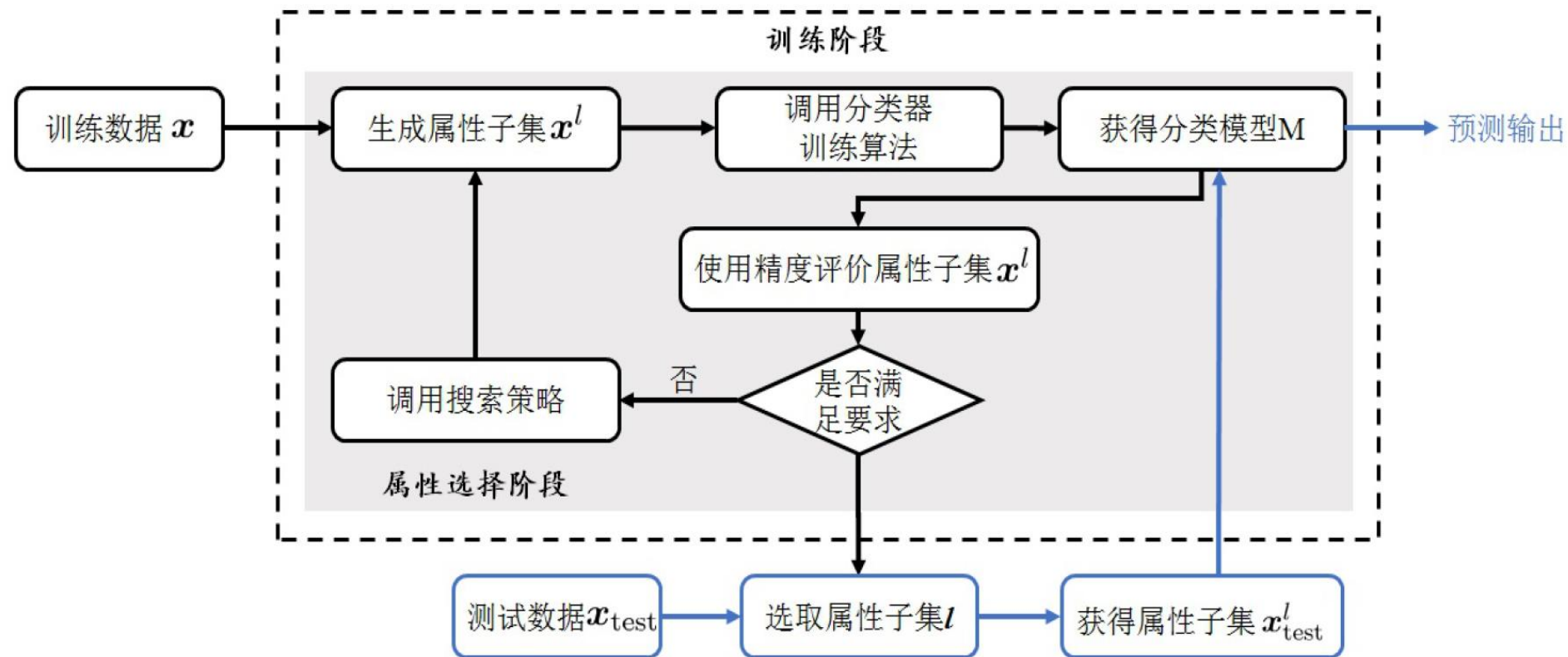


图 1.6 Wrapper式属性选择的一般流程

过滤式选择 - Relief

先用特征选择过程过滤原始数据，再用过滤后的特征来训练模型；特征选择过程与后续学习器无关

□ Relief (Relevant Features) 方法 [Kira and Rendell, 1992]

- 为每个初始特征赋予一个“相关统计量”，度量特征的重要性
- 特征子集的重要性由子集中每个特征所对应的相关统计量之和决定
- 设计一个阈值，然后选择比阈值大的相关统计量分量所对应的特征
- 或者指定欲选取的特征个数，然后选择相关统计量分量最大的指定个数特征

如何确定相关统计量？

Relief方法中相关统计量的确定

□ 猜中近邻 (near-hit) : x_i 的同类样本中的最近邻 $x_{i,nh}$

□ 猜错近邻 (near-miss) : x_i 的异类样本中的最近邻 $x_{i,nm}$

□ 相关统计量对应于属性 j 的分量为

$$\delta^j = \sum_i -\text{diff}(x_i^j, x_{i,nh}^j)^2 + \text{diff}(x_i^j, x_{i,nm}^j)^2$$

若 j 为离散型, 则 $x_a^j = x_b^j$ 时 $\text{diff}(x_a^j, x_b^j) = 0$, 否则为 1 ;
若 j 为连续型, 则 $\text{diff}(x_a^j, x_b^j) = |x_a^j - x_b^j|$, 注意 x_a^j, x_b^j 已规范化到 $[0,1]$ 区间

□ 相关统计量越大, 属性 j 上, 猜对近邻比猜错近邻越近, 即属性 j 对区分对错越有用

□ Relief方法的时间开销随采样次数以及原始特征数线性增长, 运行效率很高

Relief方法的多类拓展

Relief方法是为二分类问题设计的，其扩展变体
Relief-F_[Kononenko, 1994]能处理多分类问题

- 数据集中的样本来自 $|\mathcal{Y}|$ 个类别，其中 \mathbf{x}_i 属于第 k 类
- 猜中近邻：第 k 类中 \mathbf{x}_i 的最近邻 $\mathbf{x}_{i,nh}$
- 猜错近邻：第 k 类之外的每个类中找到一个 \mathbf{x}_i 的最近邻作为猜错近邻，记为 $\mathbf{x}_{i,l,nm}$ ($l = 1, 2, \dots, |\mathcal{Y}|; l \neq k$)
- 相关统计量对应于属性的分量为

$$\delta^j = \sum_i -\text{diff}(x_i^j, x_{i,nh}^j)^2 + \sum_{l \neq k} \left(p_l \times \text{diff}(x_i^j, x_{i,l,nm}^j)^2 \right)$$

p_l 为第 l 类样本
在数据集 D 中
所占的比例

包裹式选择

包裹式选择直接把最终将要使用的学习器的性能作为特征子集的评价准则

- ❑ 包裹式特征选择的目的是为给定学习器选择最有利于其性能、“量身定做”的特征子集
- ❑ 包裹式选择方法直接针对给定学习器进行优化，因此从最终学习器性能来看，包裹式特征选择比过滤式特征选择更好
- ❑ 包裹式特征选择过程中需多次训练学习器，计算开销通常比过滤式特征选择大得多

LVW包裹式特征选择方法

LVW (Las Vegas Wrapper) [Liu and Setiono, 1996] 在拉斯维加斯方法框架下使用随机策略来进行子集搜索，并以最终分类器的误差作为特征子集评价准则

基本步骤

- 在循环的每一轮随机产生一个特征子集
- 在随机产生的特征子集上通过交叉验证推断当前特征子集的误差
- 进行多次循环，在多个随机产生的特征子集中选择误差最小的特征子集作为最终解*

*若有运行时间限制，则该算法有可能给不出解

输入：属性全体上的数据 \mathbf{X}^L ， L 为全体属性组成的集合

输入：最大迭代次数 K

输入：数据标记 \mathbf{Y}

过程：

```
1: 初始化  $\text{err} \leftarrow 0$ ,  $k \leftarrow 0$ ,  $C \leftarrow \text{card}(L)$ ,  $T \leftarrow L$ 
2: repeat
3:   随机获取属性子集  $L^*$ ,  $C^* \leftarrow \text{card}(L^*)$ 
4:    $\text{err}^* \leftarrow \text{Learner}(\mathbf{X}^{L^*}, \mathbf{Y}, \text{'CrossValidation'})$ 
5:   if  $\text{err}^* < \text{err}$  or  $(\text{err}^* = \text{err} \text{ and } C^* < C)$  then
6:      $k \leftarrow 0$ ,  $\text{err} \leftarrow \text{err}^*$ ,  $C \leftarrow C^*$ ,  $T = L^*$ 
7:   else
8:      $k = k + 1$ 
9:   end if
10: until  $\text{err}$ 连续 $K$ 次不更新, 即 $k = K$ 
11: return  $T$ 
```

计算机领域以两个著名的赌城命名了两个著名的随机方法：拉斯维加斯（Las Vegas）方法和蒙特卡洛（Monte Carlo）方法，这两者在效用上区别在于，前者在规定时间内或者获得需要的解或者不输出，后者在规定的时间内可能会得到达不到要求的解。但如果无执行时间限制，两者都能输出满足需求的解。蒙特卡洛方法在图模型一章有介绍。

图 1.7 LVW算法伪代码

嵌入式选择

嵌入式特征选择是将特征选择过程与学习器训练过程融为一体，两者在同一个优化过程中完成，在学习器训练过程中自动地进行特征选择

- 考虑最简单的线性回归模型，以平方误差为损失函数，并引入 L_2 范数正则化项防止过拟合，则有

$$\min_{\mathbf{w}} \sum_{i=1}^m (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda \|\mathbf{w}\|_2^2$$

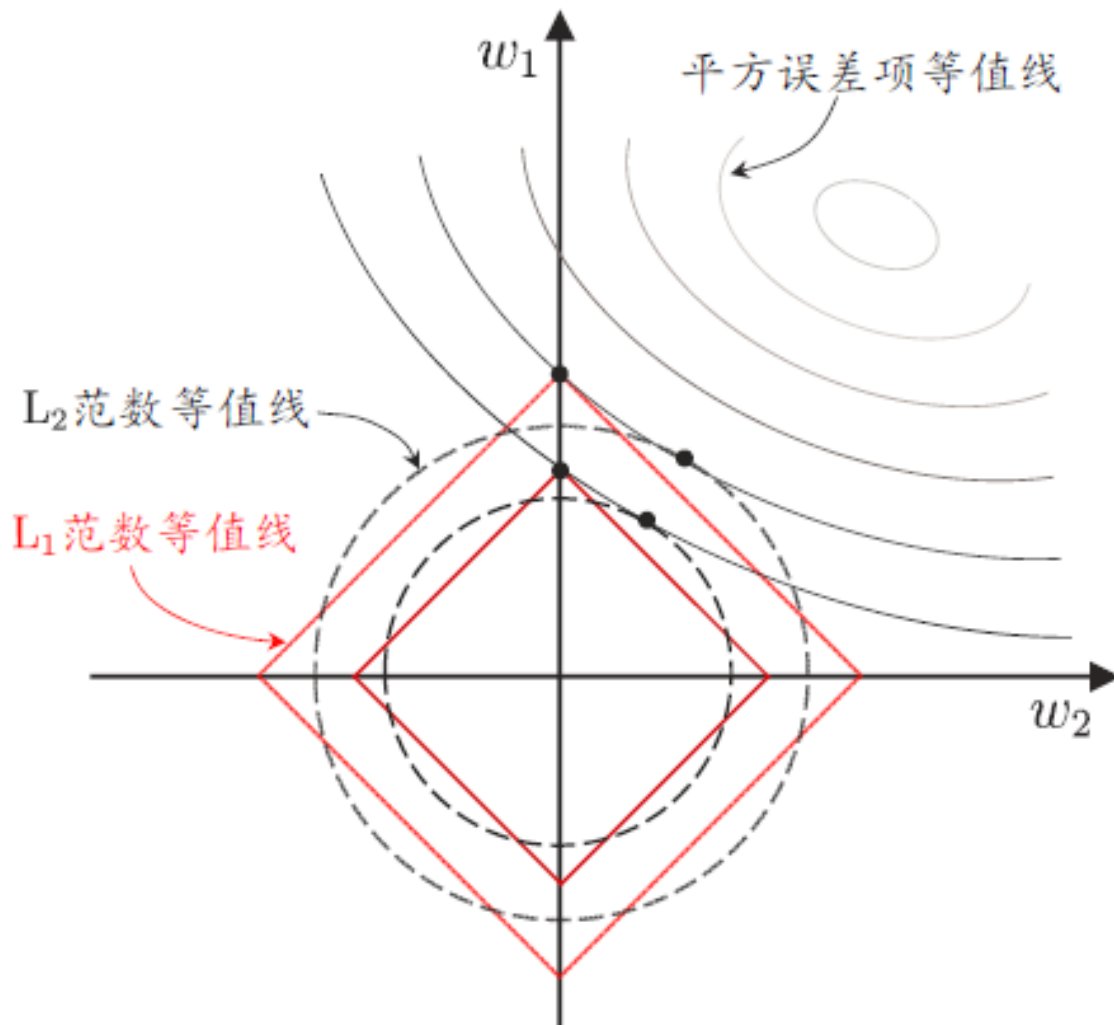
岭回归 (ridge regression)
[Tikhonov and Arsenin, 1977]

- 将 L_2 范数替换为 L_1 范数，则有**LASSO** [Tibshirani, 1996]

$$\min_{\mathbf{w}} \sum_{i=1}^m (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda \|\mathbf{w}\|_1$$

易获得稀疏解，是一种嵌入式特征选择方法

使用 L_1 范数正则化易获得稀疏解



假设 \mathbf{x} 仅有两个属性，那么 \mathbf{w} 有两个分量 w_1 和 w_2 。那么目标优化的解要在平方误差项与正则化项之间折中，即出现在图中平方误差项等值线与正则化等值线相交处。

从图中看出，采用 L_1 范数时交点常出现在坐标轴上，即产生 w_1 或者 w_2 为0的稀疏解。

等值线即取值相同的点的连线

L_1 正则化问题的求解(1)

近端梯度下降 (Proximal Gradient Descend, 简称PGD) 解法
[Boyd and Vandenberghe, 2004]

□ 写出 $f(\mathbf{x})$ 的二阶泰勒展式

$$f(\mathbf{x}) = f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k \rangle + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^\top \frac{\delta^2 f(\mathbf{x}_k)}{\delta \mathbf{x}_k^2} (\mathbf{x} - \mathbf{x}_k)$$

□ 假设 $f(\mathbf{x})$ 满足L-Lipschitz条件, 即存在常数 $L > 0$, 使得

$$\|\nabla f(\mathbf{x}') - \nabla f(\mathbf{x})\| \leq L \|\mathbf{x}' - \mathbf{x}\|_2$$

L1正则化问题的求解(2)

□ L-Lipschitz条件代入泰勒展式，可得

$$\begin{aligned} f(\mathbf{x}) &\cong f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k \rangle + \frac{L}{2} \|\mathbf{x} - \mathbf{x}_k\|^2 \\ &= \frac{L}{2} \|\mathbf{x} - (\mathbf{x}_k - \frac{1}{L} \nabla f(\mathbf{x}_k))\|^2 + \text{const} \end{aligned}$$

□ 将上式关于 $f(\mathbf{x})$ 的近似代入到原优化问题中，得

$$\min_{\mathbf{x}} \sum_{i=1}^m \frac{L}{2} \|\mathbf{x} - (\mathbf{x}_k - \frac{1}{L} \nabla f(\mathbf{x}_k))\|^2 + \lambda \|\mathbf{x}\|_1$$

L1正则化问题的求解(3)

- 每次在 \mathbf{x}_k 的附近寻找最优点，不断迭代，即寻找

$$\mathbf{x}_{k+1} = \min_{\mathbf{x}} \sum_{i=1}^m \frac{L}{2} \|\mathbf{x} - (\mathbf{x}_k - \frac{1}{L} \nabla f(\mathbf{x}_k))\|^2 + \lambda \|\mathbf{x}\|_1$$

- 假设 $\mathbf{z} = \mathbf{x}_k - 1/L \nabla f(\mathbf{x}_k)$ ，上式有闭式解

$$x_{k+1}^i = \begin{cases} z^i - \lambda/L, & \lambda/L < z_i; \\ 0, & |z^i| \leq \lambda/L; \\ z^i + \lambda/L, & z^i < -\lambda/L, \end{cases}$$

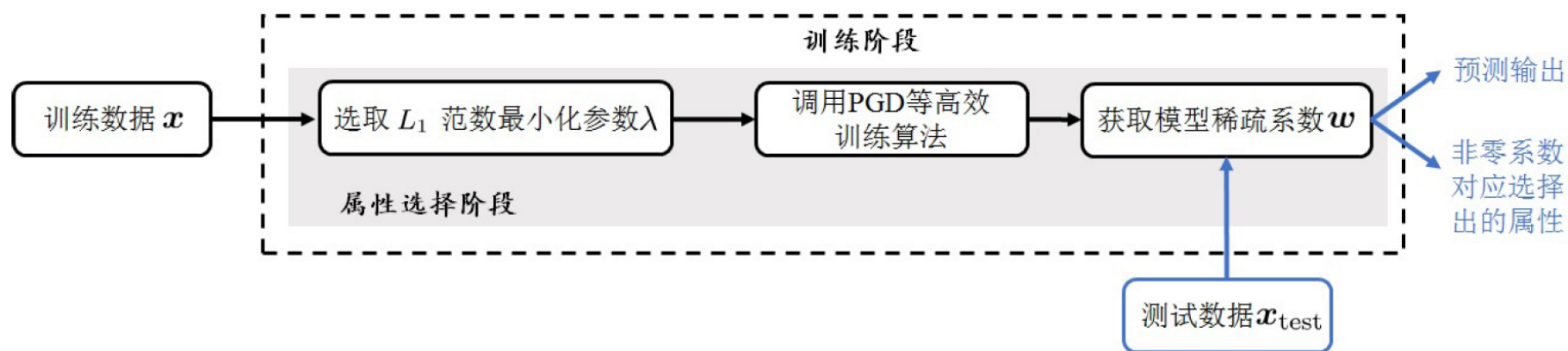


图 1.9 嵌入式属性选择的一般流程

稀疏表示

- 将数据集考虑成一个矩阵，每行对应一个样本，每列对应一个特征
- 矩阵中有很多零元素，且非整行整列出现
- 稀疏表达的优势：
 - 文本数据线性可分
 - 存储高效

能否将稠密表示的数据集转化为“稀疏表示”，使其享受稀疏表达的优势？

字典学习

为普通稠密表达的样本找到合适的字典，将样本转化为稀疏表示，这一过程称为字典学习

- 给定数据集 $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$, $\mathbf{x}_i \in \mathbb{R}^{n \times k}$
- 学习目标是字典矩阵 $\mathbf{B} \in \mathbb{R}^{d \times k}$ 以及样本的稀疏表示 $\boldsymbol{\alpha}_i \in \mathbb{R}^k$
- k 称为字典的词汇量，通常由用户指定
- 则最简单的字典学习的优化形式为

$$\min_{\mathbf{B}, \boldsymbol{\alpha}_i} \sum_{i=1}^m \|\mathbf{x}_i - \mathbf{B}\boldsymbol{\alpha}_i\|_2^2 + \lambda \sum_{i=1}^m \|\boldsymbol{\alpha}_i\|_1$$

字典学习的解法(1)

- 固定字典 \mathbf{B} , 参考LASSO的方法求解

$$\min_{\boldsymbol{\alpha}_i} \|\mathbf{x}_i - \mathbf{B}\boldsymbol{\alpha}_i\|_2^2 + \lambda \|\boldsymbol{\alpha}_i\|_1$$

- 以 \mathbf{B} 为初值求解字典 \mathbf{B} ,

$$\min_{\mathbf{B}} \|\mathbf{X} - \mathbf{B}\mathbf{A}\|_F^2$$

单击此处添加文本

$\|\cdot\|_F$ 是矩阵的Frobenius 范数

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m) \in \mathbb{R}^{d \times m}$$

$$\mathbf{A} = (\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_m) \in \mathbb{R}^{k \times m}$$

- 基于逐列更新策略的KSVD [Aharon et al., 2006]

$$\min_{\mathbf{b}_i; \boldsymbol{\alpha}_i} \left\| \mathbf{X} - \sum_{j=1}^k \mathbf{b}_j \boldsymbol{\alpha}_j^T \right\|_F^2$$

\mathbf{b}_i 表示矩阵 \mathbf{B} 的第 i 列, $\boldsymbol{\alpha}^i$ 表示矩阵 \mathbf{A} 的第 i 行

字典学习的解法(2)

- 上式可以变化为

$$\min_{\mathbf{b}_i; \boldsymbol{\alpha}^i} \left\| \left(\mathbf{X} - \sum_{j \neq i} \mathbf{b}_j \boldsymbol{\alpha}^j \right) - \mathbf{b}_i \boldsymbol{\alpha}^i \right\|_F^2 = \min_{\mathbf{b}_i; \boldsymbol{\alpha}^i} \|\mathbf{E}_i - \mathbf{b}_i \boldsymbol{\alpha}^i\|_F^2$$

- 对 \mathbf{E}_i 进行奇异值分解，取得最大奇异值对应的正交向量

为了不破坏 \mathbf{A} 的稀疏性，

$\boldsymbol{\alpha}^i$ 仅保留非零元素，

\mathbf{E}_i 仅保 $\boldsymbol{\alpha}^i$ \mathbf{b}_i 与 非零元素的乘积项

- 反复迭代以获得最优解

压缩感知

能否利用部分数据恢复全部数据？

- 数据传输中，能否利用接收到的压缩、丢包后的数字信号，精确重构出原信号？
- 压缩感知 (compressive sensing) [Cándes et al., 2006, Donoho, 2006] 为解决此类问题提供了新的思路。

- 长度为 m 的离散信号 x ，用远小于奈奎斯特采样定理的要求的采样率采样得到长度为 n 的采样后信号 y ， $n \ll m$ ，即

$$y = \Phi x$$

- 一般情况下， $n \ll m$ ，不能利用 y 还原 x ，但是

- 若存在某个线性变换 Ψ ，使得 $x = \Psi s$ ， s 是稀疏向量，即

$$y = \Phi x = \Phi \Psi s = A s$$

如傅里叶变换，余弦变换，小波变换等

A 具有“限定等距性”时，可以近乎完美地恢复 s

限定等距性

限定等距性 (Restricted Isometry Property, 即RIP)

[Cádes, 2008] : $\mathbf{A} \in \mathbb{R}^{n \times m}$, 若存在常数 $\delta_k \in (0,1)$ 使得对于任意向量 \mathbf{s} 和 \mathbf{A} 的所有子矩阵 $\mathbf{A}_k \in \mathbb{R}^{n \times k}$ 有

$$(1 - \delta_k) \|\mathbf{s}\|_2^2 \leq \|\mathbf{A}_k \mathbf{s}\|_2^2 \leq (1 + \delta_k) \|\mathbf{s}\|_2^2$$

则称 \mathbf{A} 满足 **k -限定等距性** (k -RIP)

压缩感知的优化目标和解法

若 \mathbf{A} 满足 k 限定等距性，则可通过下面的优化问题近乎完美地从 \mathbf{y} 中恢复出稀疏信号 \mathbf{s} ，进而恢复出 \mathbf{x} ：

$$\min_{\mathbf{s}} \quad \|\mathbf{s}\|_0$$

$$s.t. \quad \mathbf{y} = \mathbf{A}\mathbf{s}$$

□ L_0 范数最小化是NP难问题，将上式转化为共解的 L_1 范数最小化问题 [Cándes et al., 2006]：

$$\min_{\mathbf{s}} \quad \|\mathbf{s}\|_1$$

$$s.t. \quad \mathbf{y} = \mathbf{A}\mathbf{s}$$

□ 转化为LASSO的等价形式再通过近端梯度下降法求解，即使用“基追踪去噪”（Basis Pursuit De-Noising） [Chen et al., 1998]

矩阵补全

客户对书籍的喜好程度的评分

	《笑傲江湖》	《万历十五年》	《人间词话》	《云海玉弓缘》	《人类的故事》
赵大	5	?	?	3	2
钱二	?	5	3	?	5
孙三	5	3	?	?	?
李四	3	?	5	4	?

能否将表中已经通过读者评价得到的数据当作**部分信号**，基于压缩感知的思想**恢复**出**完整信号**从而进行书籍推荐呢？从题材、作者、装帧等角度看（相似题材的书籍有相似的读者），表中反映的信号是**稀疏**的，能通过类似压缩感知的思想加以处理。

“矩阵补全” 技术解决此类问题

矩阵补全的优化问题和解法

- 矩阵补全 (matrix completion) 技术的优化形式为

$$\min_{\mathbf{X}} \text{rank}(\mathbf{X})$$

$$\text{s.t. } \mathbf{X}_{ij} = \mathbf{A}_{ij}, (i, j) \in \Omega$$

约束表明，恢复出的矩阵中 \mathbf{X}_{ij} 应当与已观测到的对应元素相同

- \mathbf{X} : 需要恢复的稀疏信号
- $\text{rank}(\mathbf{X})$: \mathbf{X} 的秩
- \mathbf{A} : 已观测信号
- Ω : \mathbf{A} 中已观测到的元素的位置下标的集合

- NP难问题. 将 $\text{rank}(\mathbf{X})$ 转化为其凸包“核范数” (nuclear norm)

$$\min_{\mathbf{X}} \|\mathbf{X}\|_*$$

$$\text{s.t. } (\mathbf{X})_{ij} = (\mathbf{A})_{ij}, (i, j) \in \Omega$$

$\|\mathbf{X}\|_* = \sum_{j=1}^{\min\{m,n\}} \sigma_j(\mathbf{X})$ $\sigma_j(\mathbf{X})$ 表示 \mathbf{X} 的奇异值，即矩阵的核范数为矩阵的奇异值之和。

- 凸优化问题，通过半正定规划求解 (SDP, Semi-Definite Programming)

满足一定条件时，只需观察到 $O(mr \log^2 m)$ 个元素就能完美恢复 \mathbf{A}
[Recht, 2011]

前往.....

