机器学习
Machine learning

第四章 非线性分类
Nonlinear Classifier

授课人：周晓飞
zhouxiaofei@iie.ac.cn
2023-10-27

课件放映 –〉PDF 视图-〉全屏模式

第四章 非线性分类

4.1 概述

4.2 决策树

4.3 最近邻方法

4.4 集成学习

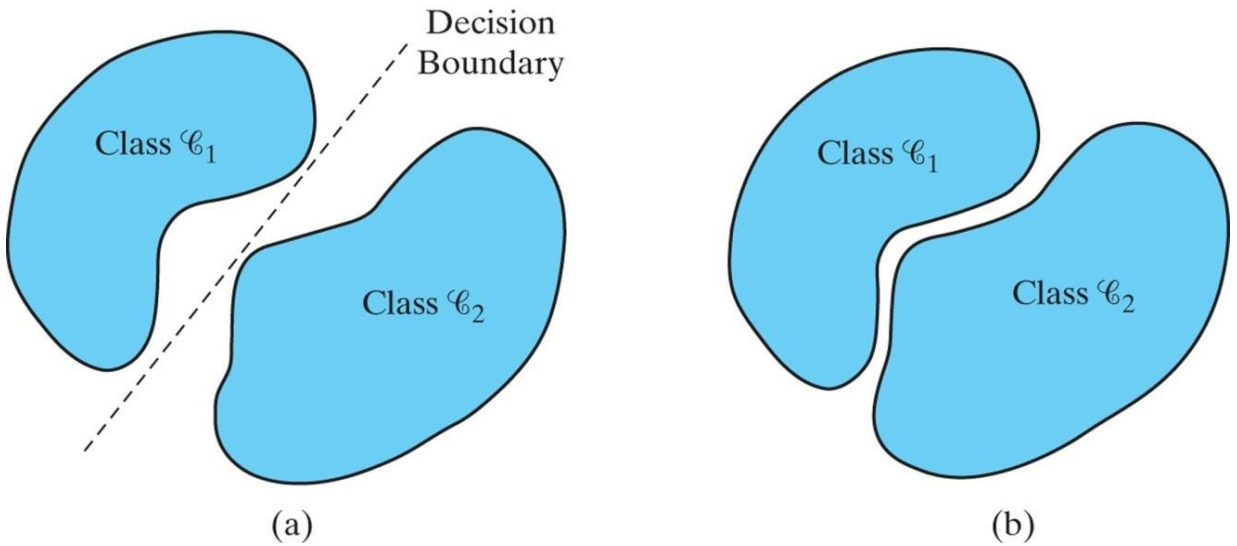
4.5 非线性 SVM

第四章 非线性分类

概述

非线性问题

对于线性不可分数据，采用非线性决策的方法。



(a) Linearly separable patterns. (b) Non-linearly separable.

概述

非线性方法概述

线性扩展的思想

--线性扩展模型 $g(x) = w^{\mathrm{T}}\varphi(x) + \mathrm{b}$

--核函数方法 $g(x) = w^{\mathrm{T}}\mathbf{k}(x, x_i) + \mathrm{b}$

非线性的思想

--最近邻

--决策树

--神经网络

--集成学习

概述

本章内容

掌握决策树、集成学习；

了解最近邻、核函数方法；

神经网络方法在课程最后一章讲授。

第四章 非线性分类

4.1 概述

4.2 决策树

4.3 最近邻方法

4.4 集成学习

4.5 非线性 SVM

决策树

决策树方法概述

决策树的目标

在树结构上，根据节点的判断，搜索类别.

问题：如何构建这样的一个决策树?

理解与解释（二叉树为例）

- 决策问题是二判决，如：X<a?
- 节点：决策问题的划分，X_t into X_{tY} and X_{tN}：

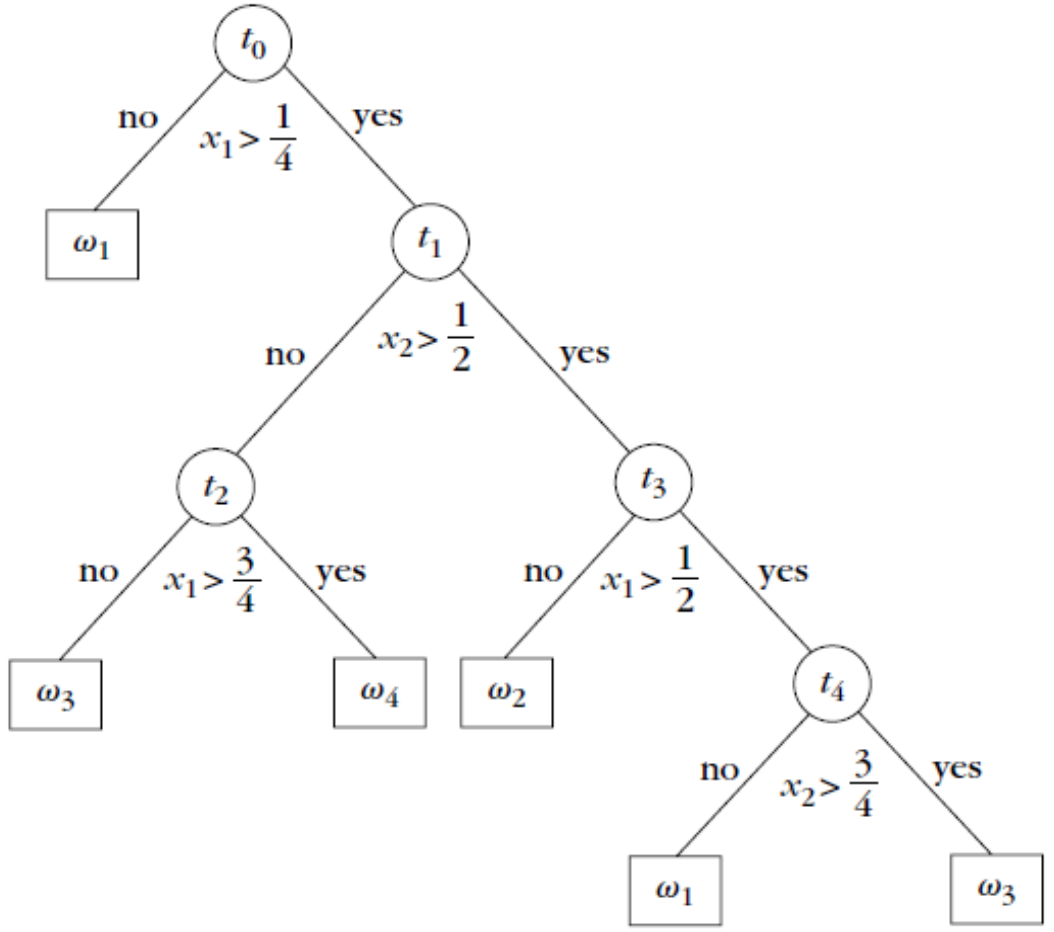
$$X_{tY} \cap X_{tN} = \Phi$$
$$X_{tY} \cup X_{tN} = X_t$$

- 叶子：A special class

决策树

决策树方法概述

二叉树直观解释



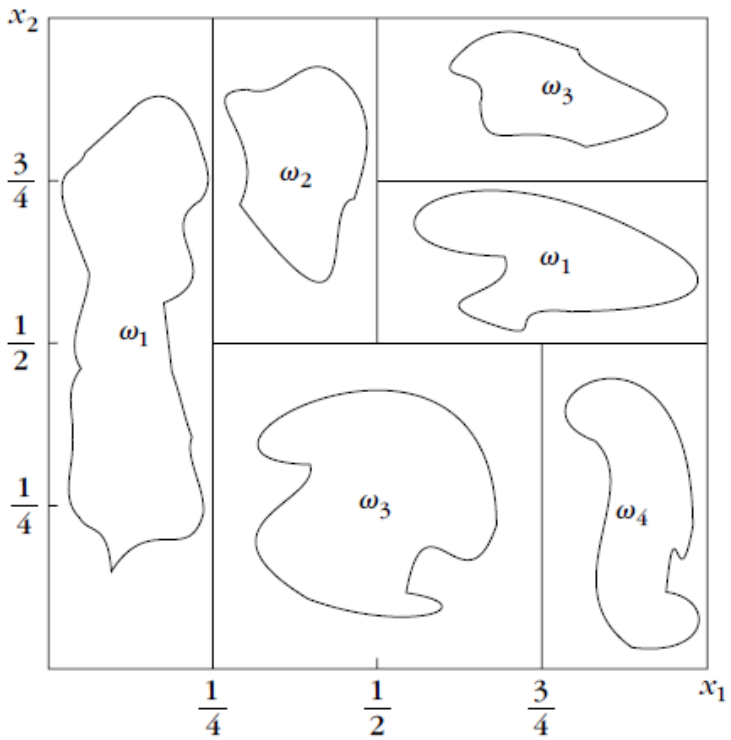
决策树

决策树方法概述

二叉树几何解释：

向量空间中特征不相关的矩形划分。

经过一系列的决策划分，得到相应类别区域。



决策树方法概述

树结构的优点：

可以不必测试所有特征和区域。

问题：

1. 高维特征空间不可见，如何确定每个节点问题?
2. 我们怎么能知道该从那些特征开始?

决策树

决策树方法概述

决策树的关键问题：

- 问题数
- 划分（问题）选择
- 决策树生成
- 剪枝处理

决策树

问题数

1. 离散值情况：以特征或特征的可能离散值作为问题；

属性 A*i* 的可能离散取值个数为 n_i；

- 方法 1

  每个特征可以作为候选问题，例如 ID3、C4.5；

  属性 A*i* 产生的候选问题数为 $N_i=1$；

- 方法 2

  每个特征的每个离散值作为候选问题，例如 CART；

  属性 A*i* 产生的候选问题数为 $N_i=n_i$；

决策树

问题数

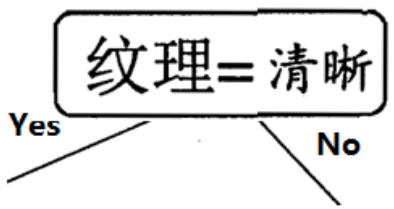
编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	0.774	0.376	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	0.634	0.264	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	0.608	0.318	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	0.556	0.215	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	0.403	0.237	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	0.481	0.149	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	0.437	0.211	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	0.666	0.091	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	0.243	0.267	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	0.245	0.057	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	0.343	0.099	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	0.639	0.161	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	0.657	0.198	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	0.360	0.370	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	0.593	0.042	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	0.719	0.103	否

决策树

问题数

方法 1：纹理这一个属性作为一个问题；

方法 2：纹理的 3 个属性，可以分别作为一个问题；



决策树

问题数

2. 连续值情况：以每个维度的样本特征值作为问题

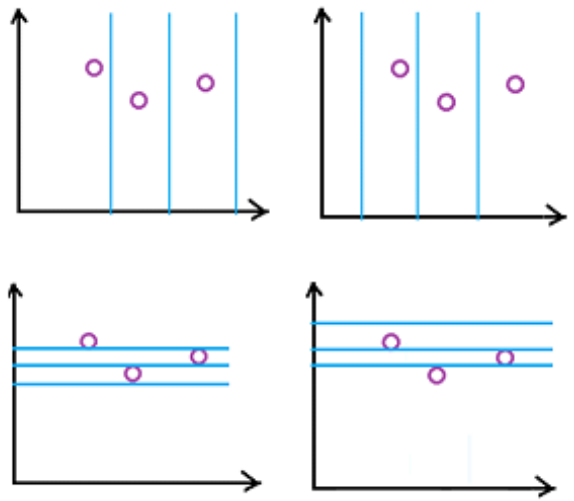
属性 A*i* 上出现的样本特征值个数为 ns_i；

- 方法：每个特征上的样本特征值作为候选问题；

  属性 A*i* 产生的候选问题数为 $N_i=ns_i$；

每个维度上，对于 N 个样本，最多有 N 种划分。

$$
\begin{aligned}
x_1 &= (a_{11}, a_{12}, \ldots a_{1j}, \ldots a_{1d}) \\
x_2 &= (a_{21}, a_{22}, \ldots a_{2j}, \ldots a_{2d}) \\
&\ldots \\
x_i &= (a_{i1}, a_{i2}, \ldots a_{ij}, \ldots a_{id}) \\
&\ldots \\
x_N &= (a_{N1}, a_{N2}, \ldots a_{Nj}, \ldots a_{Nd})
\end{aligned}
$$



决策树

问题数

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	0.774	0.376	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	0.634	0.264	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	0.608	0.318	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	0.556	0.215	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	0.403	0.237	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	0.481	0.149	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	0.437	0.211	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	0.666	0.091	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	0.243	0.267	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	0.245	0.057	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	0.343	0.099	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	0.639	0.161	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	0.657	0.198	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	0.360	0.370	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	0.593	0.042	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	0.719	0.103	否

N_i=17

决策树

问题数

候选问题数:

无论特征值是连续还是离散，确定每个属性所产生的候选问题，候选的问题总数为 $N = \sum N_i$

决策树

问题数--例子

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

- 以属性作为问题，一共问题数？
- 以属性值作为问题，一共多少问题？

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	0.774	0.376
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	0.634	0.264
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	0.608	0.318
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	0.556	0.215
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	0.403	0.237
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	0.481	0.149
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	0.437	0.211
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	0.666	0.091
10	青绿	硬挺	清脆	清晰	平坦	软粘	0.243	0.267
11	浅白	硬挺	清脆	模糊	平坦	硬滑	0.245	0.057
12	浅白	蜷缩	浊响	模糊	平坦	软粘	0.343	0.099
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	0.639	0.161
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	0.657	0.198
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	0.360	0.370
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	0.593	0.042
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	0.719	0.103

- 包含了连续值属性后，一共问题数？

决策树

划分选择

1. 非纯度

Impurity Measure 是度量类别划分的如何?

Impurity Measure（IM）定义应满足两点:

- **IM 最大值时，各类别概率相等**

$$p_i = \frac{1}{number\ \ of\ \ classes}$$

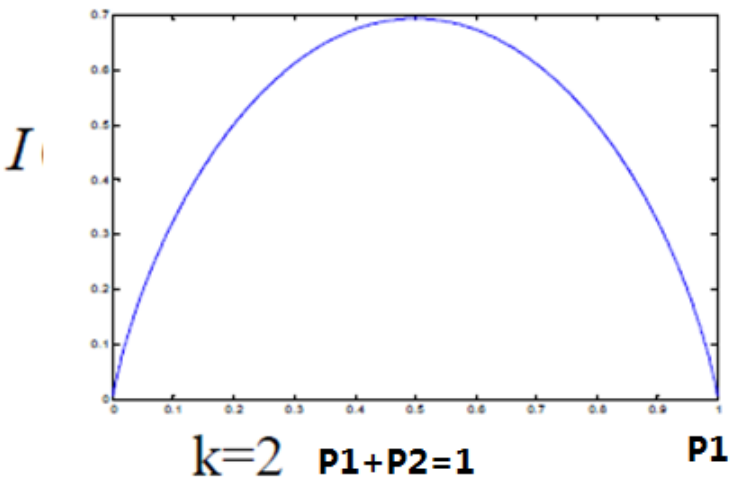
- **IM 最小时为 0，只有一类（期望的目标）**

决策树

划分选择

非纯度的**熵度量(Quinlan, C4.5)**

$$I(D) = Entropy\ (D) = -\sum_{i=1}^{k} p_i \log\ p_i$$



决策树

划分选择

非纯度的**基尼度量（Breiman CART）**

$$I(D) = Gini(D) = 1 - \sum_{i=1}^{k} p_i^2$$

决策树

划分选择

2. 划分选择目标

划分目标：选择最大减少类别非纯度的问题作为划分节点。非纯度的减少量为：

$$\Delta I(t) = I(t) - \frac{N_{tY}}{N_t} I(tY) - \frac{N_{tN}}{N_t} I(tN)$$

3. 基于非纯度变化量的三个指标：

信息增益（ID3）：越大越好

增益率（C4.5）：越大越好

基尼指数：越小越好

决策树

划分选择

- **信息增益（熵度量）**

$$Gain(D,a) = Entropy(D) - \sum_{v \in Values(A)} \frac{|D^v|}{|D|} Entropy(D^v)$$

v 是问题 a 导致的决策划分数目；

存在的问题：

$Gain(D,a)$ 倾向于选择划分集合个数 v 多的节点，

例如：区间划分的越细，区间内纯度越高，

极端情况每个区间只有一个样本，则熵为 0.

决策树

划分选择

- 增益率（信息增益与数据集 D 关于问题 a 的熵值之比）

$$Gini\_ratio(D,a)=\frac{Gain(D,a)}{IV(a)}$$

$$IV(a)=-\sum_{v=1}^{V}\frac{|D^v|}{|D|}\log\frac{|D^v|}{|D|}$$

增益率改善 $Gain(D,a)$： $IV(a)$对划分集合个数 v 少的属性有所偏好，

$IV(a)$的最大值随 v 的变化是：v 越小则 IV(a)越小。例如：

$$-\sum_{v=1}^{3}\frac{1}{3}\log\frac{1}{3} > -\sum_{v=1}^{2}\frac{1}{2}\log\frac{1}{2}$$

决策树

划分选择

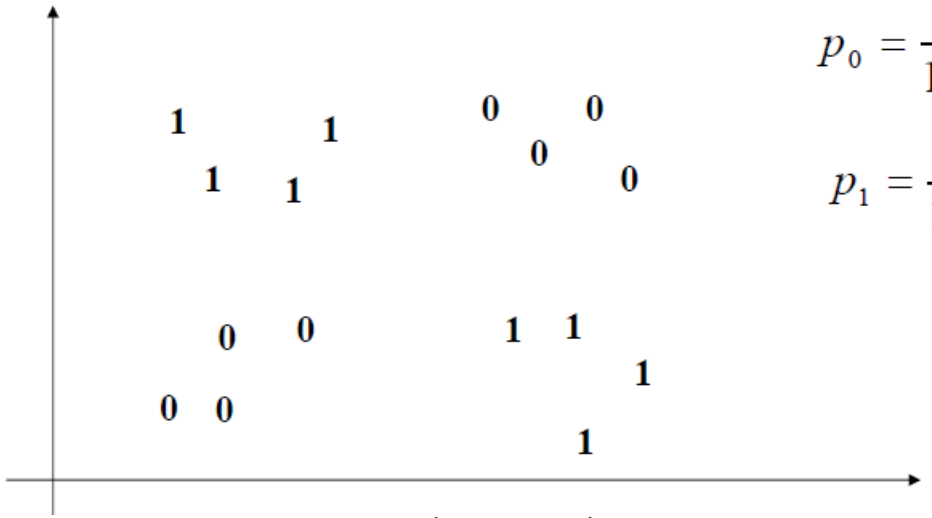
- 基尼指数（基尼度量）

$$Gini\_index(D,a)=\sum_{v=1}^{V}\frac{|D^v|}{|D|}Gini(D^v)$$

划分选择

4. 划分选择的示例：

$$a^* = \underset{a \in A}{\arg\max}\ Gain(D,a)$$



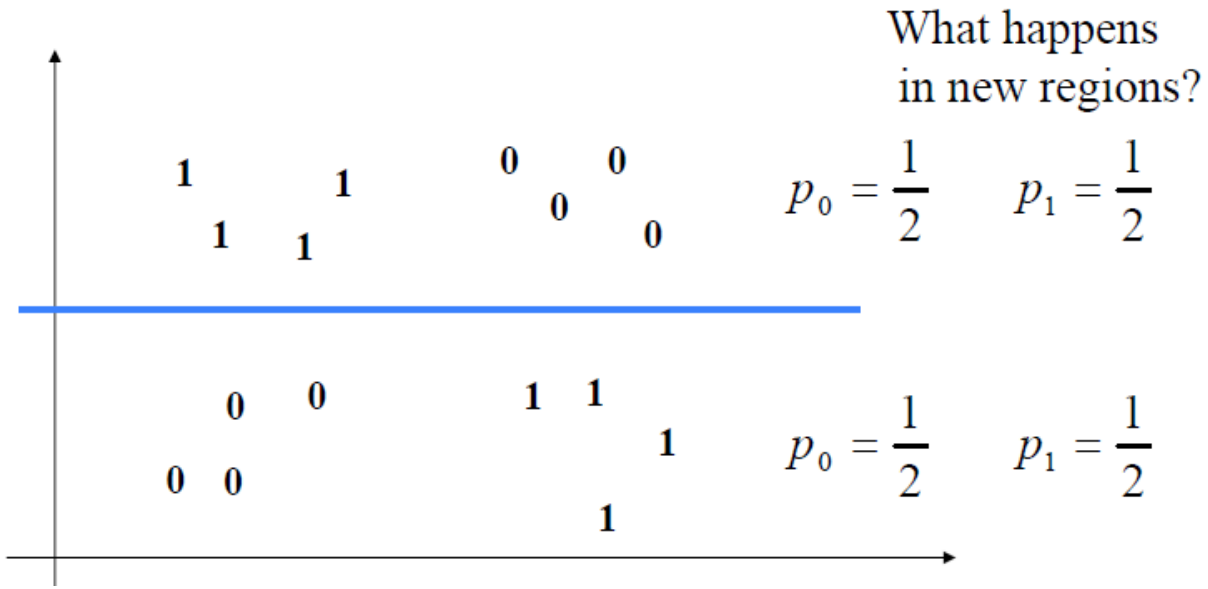
$$p_0 = \frac{8}{16} = \frac{1}{2}$$

$$p_1 = \frac{8}{16} = \frac{1}{2}$$

划分前 $I = 2 \times \left(-\frac{1}{2}\log\frac{1}{2} \right)$

决策树

划分选择



No improvement in the impurity measure !!!

划分前 I=I(t)，划分后 $I = \frac{N_{tY}}{N_t} I(t_Y) + \frac{N_{tN}}{N_t} I(t_N) = I(t)$ **，** $\Delta I = 0$

决策树

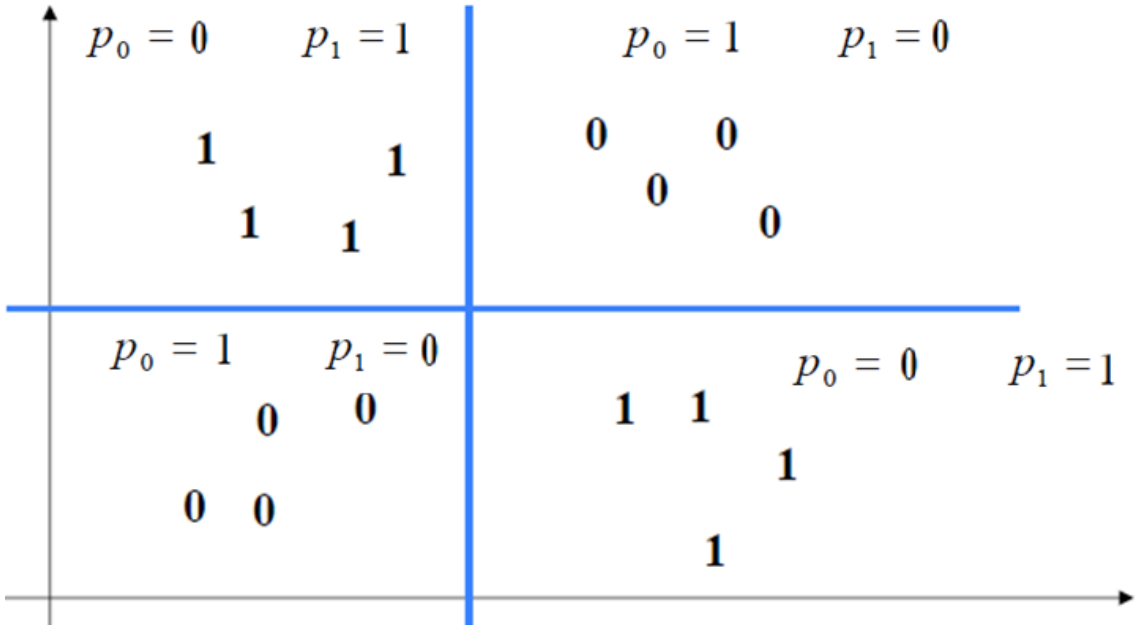
划分选择

No improvement in the impurity measure !!!

划分前 I=I(t)，划分后 $I = \frac{N_{tY}}{N_t} I(t_Y) + \frac{N_{tN}}{N_t} I(t_N) = I(t)$ **，** $\Delta I = 0$

决策树

划分选择



划分前 I=I(t)，划分后 I=0，ΔI =I(t)

划分选择

5. 节点类别设置

叶子节点纯度达到预设阈值后，停止划分，并对叶子节点进行类别设置。

按概率最大的类别设定：

$$j = \arg\max_{i} p_i$$

决策树生成

1. 决策树生成过程

从顶向下（不断增加一个节点）

- 准则：所有划分中选择一个使 ΔI（非纯度减少量）最大的划分为节点，加入决策树。

ΔI 的计算：ΔI = 划分前 - 划分后；

- 贪婪学习： 根据划分准则，在问题集上进行划分，直到 Impurity 不能再改善，或达到较小的改善。
- 停止规则：设定阈值

决策树

决策树生成

2. ID3 决策树

- 属性特征作为结点问题，
  划分选择实际是特征选择过程；
- 划分选择依据：最大化信息增益。

ID3 算法流程

输入：训练数据集 D，特征集 A，阈值 ε；
输出：决策树 T.

（1）若 D 中所有实例属于同一类 C_k，则 T 为单结点树，并将类 C_k 作为该结点的类标记，返回 T；

（2）若 $A=\varnothing$，则 T 为单结点树，并将 D 中实例数最大的类 C_k 作为该结点的类标记，返回 T；

（3）否则，按算法 计算 A 中各特征对 D 的信息增益，选择信息增益最大的特征 A_g；

（4）如果 A_g 的信息增益小于阈值 ε，则置 T 为单结点树，并将 D 中实例数最大的类 C_k 作为该结点的类标记，返回 T；

（5）否则，对 A_g 的每一可能值 a_i，依 $A_g=a_i$ 将 D 分割为若干非空子集 D_i，将 D_i 中实例数最大的类作为标记，构建子结点，由结点及其子结点构成树 T，返回 T；

（6）对第 i 个子结点，以 D_i 为训练集，以 $A-\{A_g\}$ 为特征集，递归地调用步 (1) ～步 (5)，得到子树 T_i，返回 T_i. ■

李航《统计学习方法》

决策树

决策树生成

3. C4.5 决策树

- 属性特征作为结点问题，
  划分选择实际是特征选择过程；
- 划分选择依据：最大化信息增益率

C4.5 算法流程

输入：训练数据集 D，特征集 A，阈值 ε；

输出：决策树 T.

（1）如果 D 中所有实例属于同一类 C_k，则置 T 为单结点树，并将 C_k 作为该结点的类，返回 T；

（2）如果 $A=\varnothing$，则置 T 为单结点树，并将 D 中实例数最大的类 C_k 作为该结点的类，返回 T；

（3）否则，计算 A 中各特征对 D 的信息增益比，选择信息增益比最大的特征 A_g；

（4）如果 A_g 的信息增益比小于阈值 ε，则置 T 为单结点树，并将 D 中实例数最大的类 C_k 作为该结点的类，返回 T；

（5）否则，对 A_g 的每一可能值 a_i，依 $A_g=a_i$ 将 D 分割为子集若干非空 D_i，将 D_i 中实例数最大的类作为标记，构建子结点，由结点及其子结点构成树 T，返回 T；

（6）对结点 i，以 D_i 为训练集，以 $A-\{A_g\}$ 为特征集，递归地调用步(1)～步(5)，得到子树 T_i，返回 T_i. ■

李航《统计学习方法》

决策树

决策树生成

4. CART 决策树

- 属性特征离散值作为结点问题，本质是二叉树；
- 划分选择依据：最小化基尼指数。

（李航《统计学习方法》）

CART 算法流程

输入：训练数据集 D，停止计算的条件；

输出：CART 决策树.

根据训练数据集，从根结点开始，递归地对每个结点进行以下操作，构建二叉决策树：

（1）设结点的训练数据集为 D，计算现有特征对该数据集的基尼指数. 此时，对每一个特征 A，对其可能取的每个值 a，根据样本点对 $A=a$ 的测试为“是”或“否”将 D 分割成 D_1 和 D_2 两部分，计算 $A=a$ 时的基尼指数.

（2）在所有可能的特征 A 以及它们所有可能的切分点 a 中，选择基尼指数最小的特征及其对应的切分点作为最优特征与最优切分点. 依最优特征与最优切分点，从现结点生成两个子结点，将训练数据集依特征分配到两个子结点中去.

（3）对两个子结点递归地调用（1），（2），直至满足停止条件.

（4）生成 CART 决策树. ■

算法停止计算的条件是结点中的样本个数小于预定阈值，或样本集的基尼指数小于预定阈值（样本基本属于同一类），或者没有更多特征.

李航《统计学习方法》

决策树生成

5. 连续值二叉决策树

- Begin with the root node, that is, $X_t = X$
- For each new node t
  - For every feature $x_k, k = 1, 2, \ldots, l$
    - For every value $\alpha_{kn}, n = 1, 2, \ldots, N_{tk}$
      - — Generate X_{tY} and X_{tN} according to the answer in the question: is $x_k(i) \leq \alpha_{kn},\ i = 1, 2, \ldots, N_t$
      - — Compute the impurity decrease
    - End
    - Choose α_{kn_0} leading to the maximum decrease w.r. to x_k
  - End
  - Choose x_{k_0} and associated $\alpha_{k_0 n_0}$ leading to the overall maximum decrease of impurity
  - If the stop-splitting rule is met, declare node t as a leaf and designate it with a class label
  - If not, generate two descendant nodes t_Y and t_N with associated subsets X_{tY} and X_{tN}, depending on the answer to the question: is $x_{k_0} \leq \alpha_{k_0 n_0}$
- End

决策树

决策树生成--例子

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^1 (色泽=青绿)

D^2 (色泽= 乌黑)

D^3 (色泽=浅白)

$$\mathrm{Ent}(D^1) = -\left(\frac{3}{6}\log_2\frac{3}{6} + \frac{3}{6}\log_2\frac{3}{6}\right) = 1.000\,,$$

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

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

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

$$\mathrm{Gain}(D, \text{色泽}) = \mathrm{Ent}(D) - \sum_{v=1}^{3}\frac{|D^v|}{|D|}\mathrm{Ent}(D^v)$$

$$= 0.998 - \left(\frac{6}{17}\times 1.000 + \frac{6}{17}\times 0.918 + \frac{5}{17}\times 0.722\right)$$

决策树生成--例子

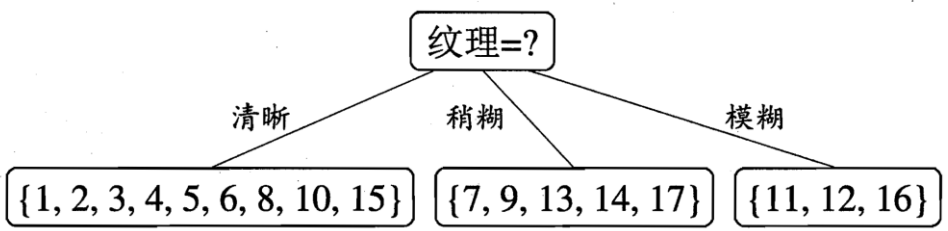
类似的, 我们可计算出其他属性的信息增益:

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

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

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

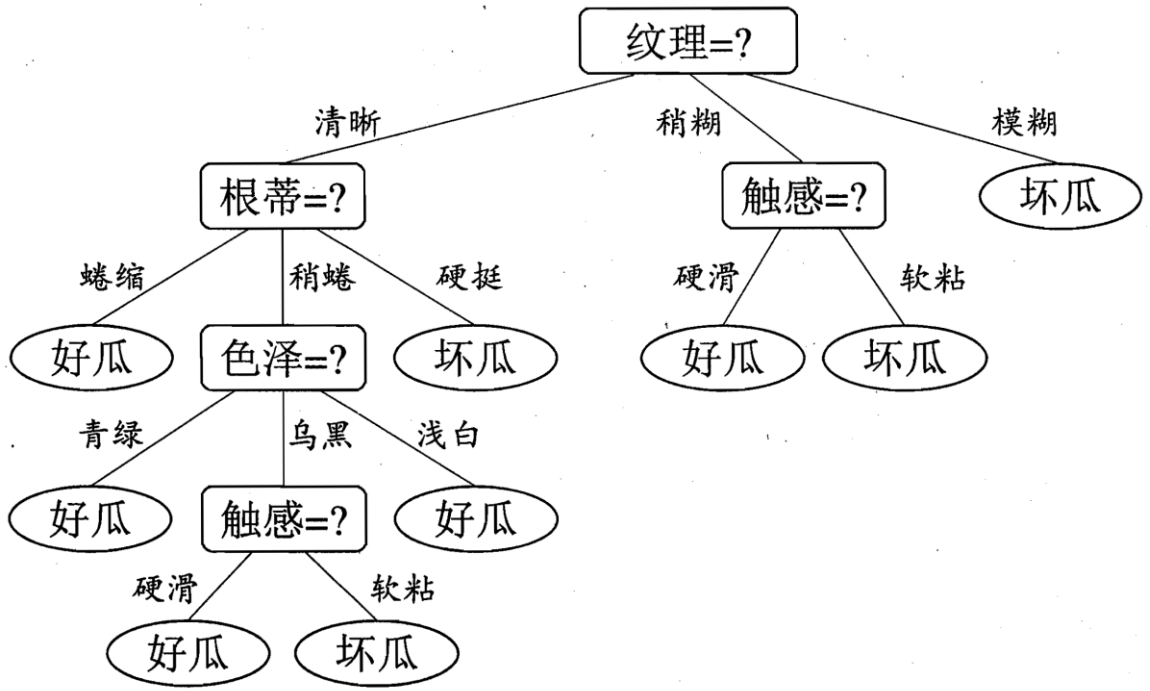
显然, 属性“纹理”的信息增益最大, 于是它被选为划分属性. 图 4.3 给出了基于“纹理”对根结点进行划分的结果, 各分支结点所包含的样例子集显示在结点中.



决策树

决策树生成--例子

最后的决策树



决策树

剪枝处理

1. ID3、C4.5 决策树剪枝

- 代价函数

设树 T 的叶结点个数为 $|T|$，t 是树 T 的叶结点，

叶结点有 N_t 个样本点，其中 k 类的样本点有 N_{tk} 个，$k=1,2,\cdots,K$，$H_t(T)$ 为叶结点 t 上的经验熵，$\alpha \geqslant 0$ 为参数，则决策树学习的损失函数可以定义为

$$C_\alpha(T)=C(T)+\alpha|T|$$

决策树的类别不纯度平均 + 树结构的复杂度（正则项）

经验熵为 $H_t(T)=-\sum_k \frac{N_{tk}}{N_t}\log\frac{N_{tk}}{N_t}$

$$C(T)=\sum_{t=1}^{|T|}N_tH_t(T)=-\sum_{t=1}^{|T|}\sum_{k=1}^{K}N_{tk}\log\frac{N_{tk}}{N_t}$$

决策树

剪枝处理

- **剪枝算法：**

输入：生成算法产生的整个树 T，参数 α；

输出：修剪后的子树 T_α.

（1）计算每个结点的经验熵.

（2）递归地从树的叶结点向上回缩.

设一组叶结点回缩到其父结点之前与之后的整体树分别为 T_B 与 T_A，其对应的损失函数值分别是 $C_\alpha(T_B)$ 与 $C_\alpha(T_A)$，如果

$$C_\alpha(T_A) \leqslant C_\alpha(T_B) \tag{5.15}$$

则进行剪枝，即将父结点变为新的叶结点.

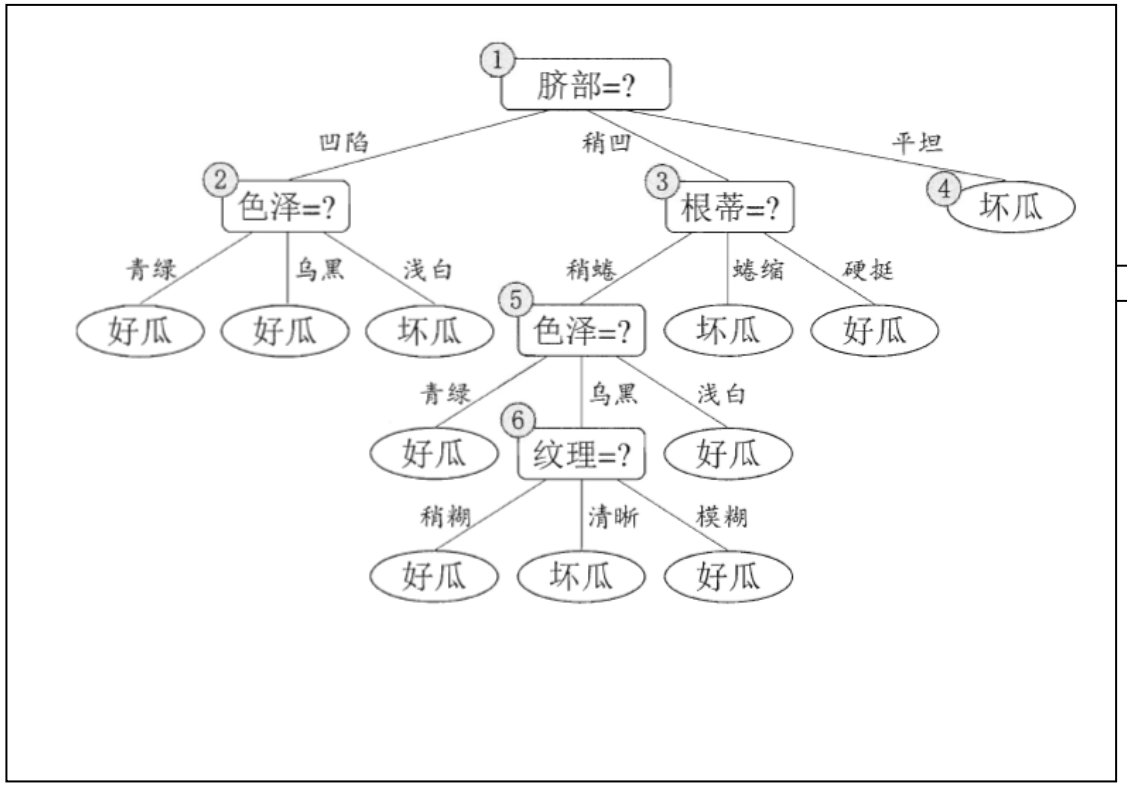
（3）返回 (2)，直至不能继续为止，得到损失函数最小的子树 T_α. ■

决策树

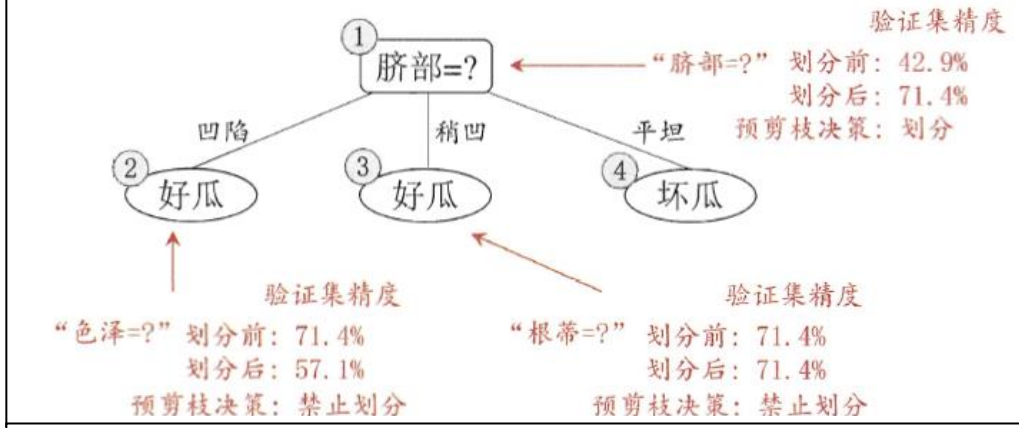
剪枝处理

2. 泛化性能评估法

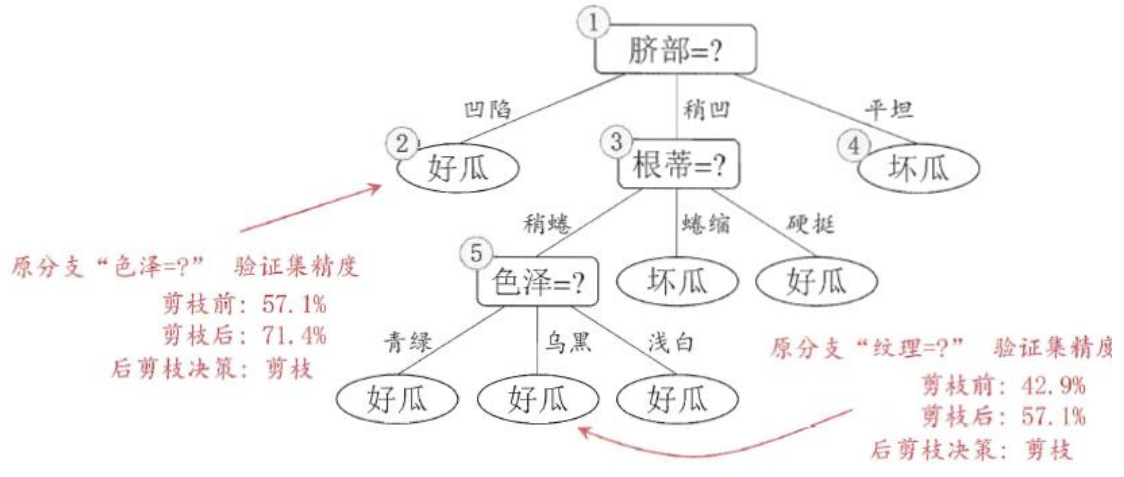
周志华《机器学习》



预剪枝



后剪枝



第四章 非线性分类

最近邻

最近近邻算法，最早是由 Cover & Hart 于 1968 年提出的，由于对该方法在理论上进行了深入分析，直至现在仍是模式识别非参数法中最重要的方法之一。

以下介绍一般的近邻法，然后讨论几种改进的近邻法

- 最近邻法
- k-近邻法
- 近邻法的快速算法
- 可做拒绝决策的近邻法

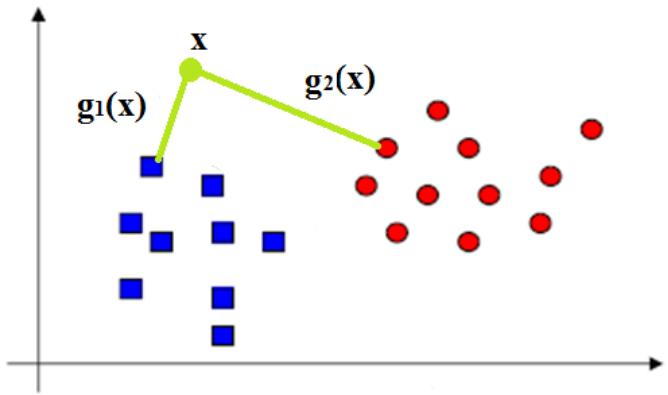
最近邻

原理：将样本分类为离之最近的样本类别

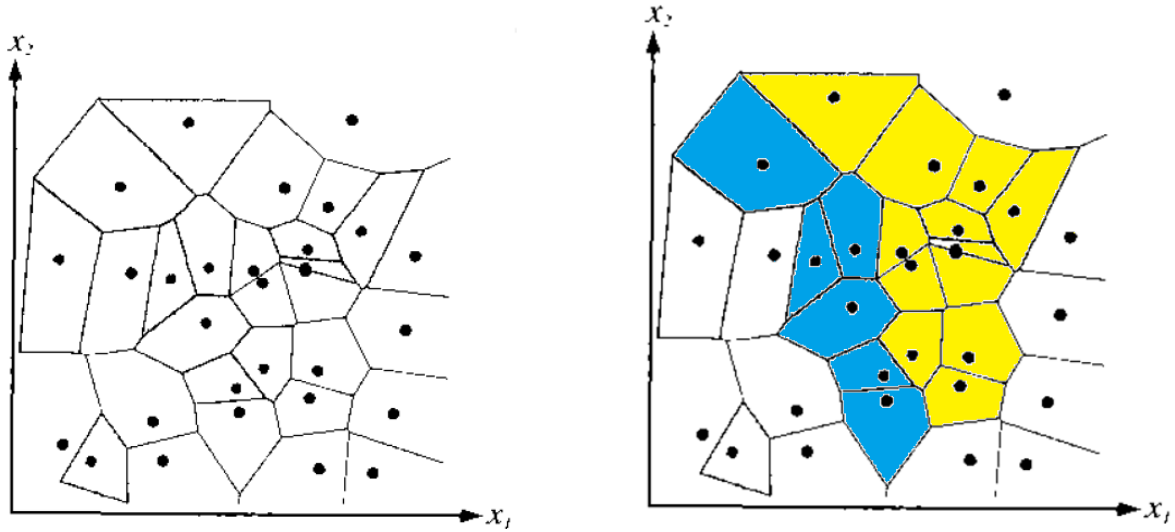
分类准则：

$$\omega_i\text{类判别函数} \quad g_i(x) = \min_k \left\| x - x_i^k \right\|, \ x_i^k \in \omega_i, \ k = 1, 2..., N_i$$

$$\text{决策规则：} \quad if \ \ g_j(x) = \min_{i=1,...c} g_i(x), \ then \ \ x \in \omega_j$$



最近邻



最近邻分类隐含的决策边界是非线性的

最近邻

K 近邻

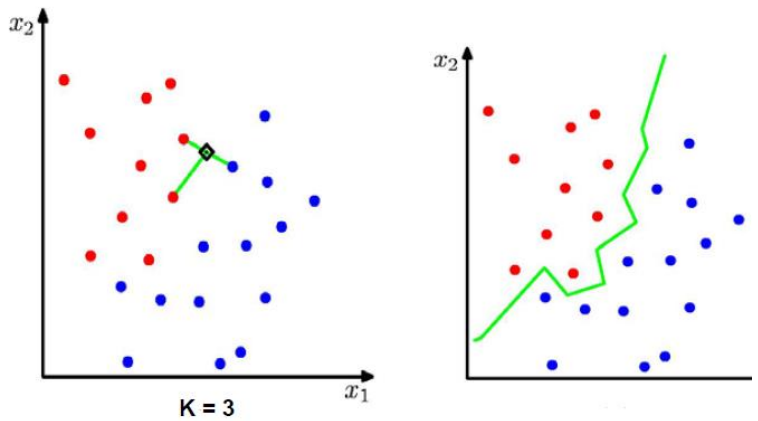
原理：将样本分给 k 个近邻中类别样本个数最多的类

分类准则：

k_i， $i=1,\cdots,c$ 为 x 的 k 个近邻中属于 ω_i 的样本数

判别函数： $g_i(x)=k_i$， $i=1,\cdots,c$

决策规则： if $g_j(x)=\max\limits_{i=1,\cdots,c} k_i$， then $x\in\omega_j$



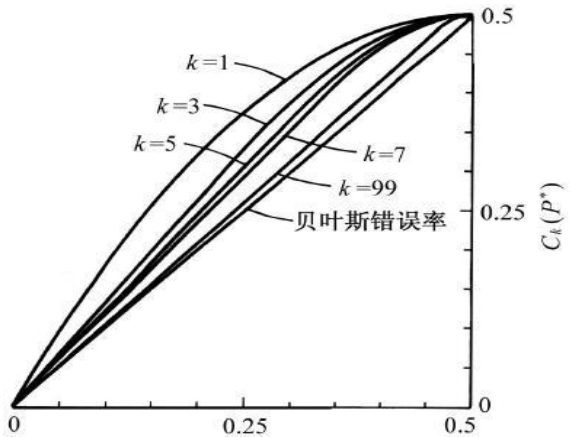
K 近邻

误差讨论

$$P^* \leqslant P \leqslant P^*\left(2-\frac{c}{c-1}P^*\right)$$

由于 P^* 一般较小，若将上式右边括号中第二项忽略，则可粗略表示为

$$P^* \leqslant P \leqslant 2P^*$$



其中 P^* 为贝叶斯错误率，c 为类数

K 近邻

近邻法的缺点

存储量大：训练样本需要存到内存

计算量大：每次决策都要计算所有样本的相似性

最近邻

快速算法一：快速搜索近邻法

原理：

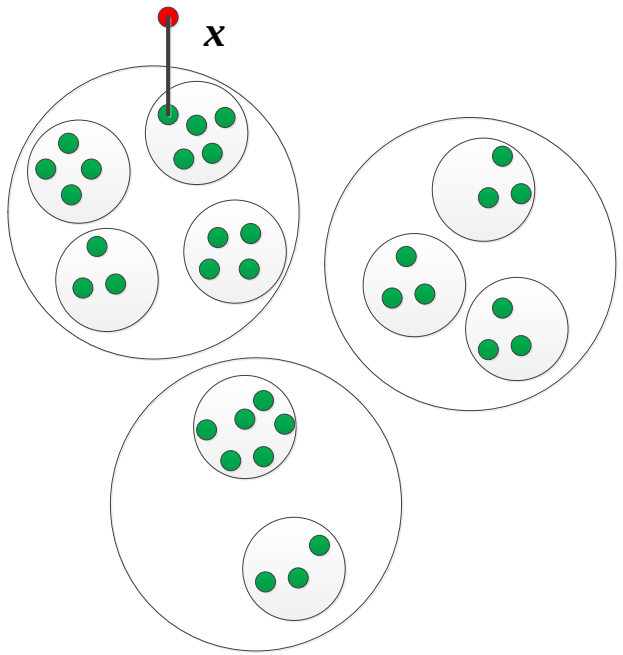
将样本分成不相交的子集，基于子集的搜索

1) 样本分级分层为多个子集
2) 逐层搜出一个最优子集
3) 在最后的子集中局部找最近样本点

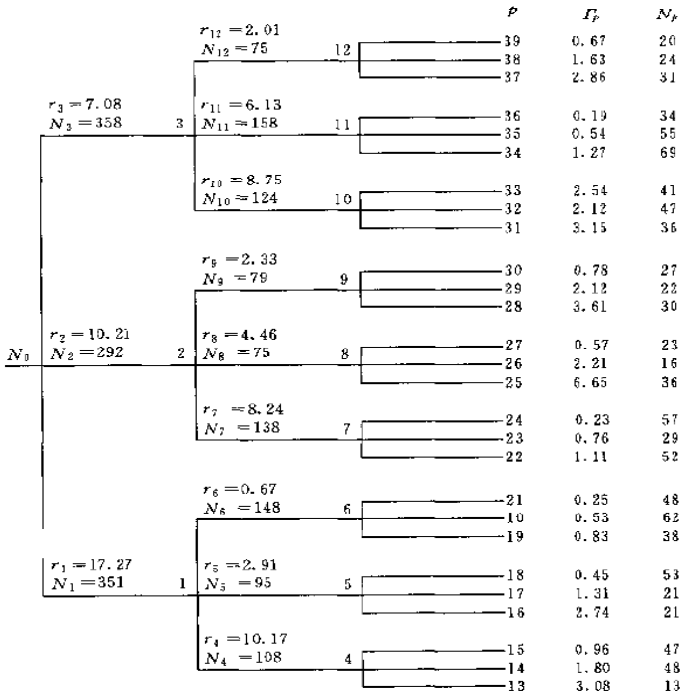
最近邻

快速算法一：快速搜索近邻法

样本集分级



几何空间中，不同层的半径圆



树结构

最近邻

快速算法一：快速搜索近邻法

符号：

M_p：子集 S_p 中的样本均值（中心点）

$r_p = \max_{x_i \in X_p} D(x_i, M_p)$：$S_p$ 中离中心点最远的距离

B：当前搜索到的最近邻距离

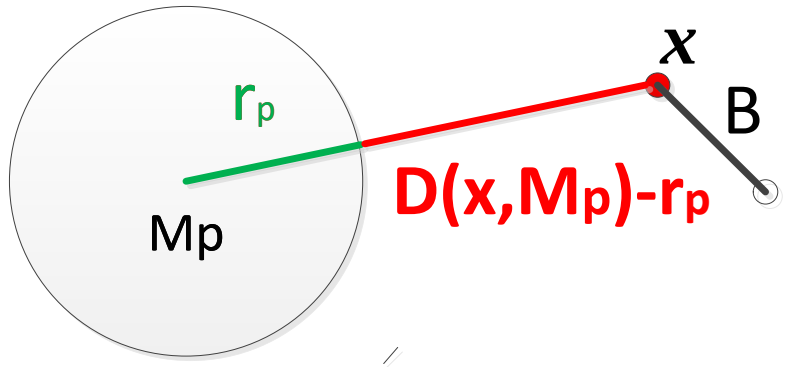
近似为子集半径

算法规则：

规则 1：找最近子集

如果 x 到 S_p 的距离 > 当前最近子集距离 B，S_p 被忽略。

$$D(x, M_p) - r_p > B$$



最近邻

快速算法一：快速搜索近邻法

符号：

M_p：子集 S_p 中的样本均值（中心点）

$r_p = \max_{x_i \in X_p} D(x_i, M_p)$：$S_p$ 中离中心点最远的距离

B：当前搜索到的最近邻距离

近似为子集半径

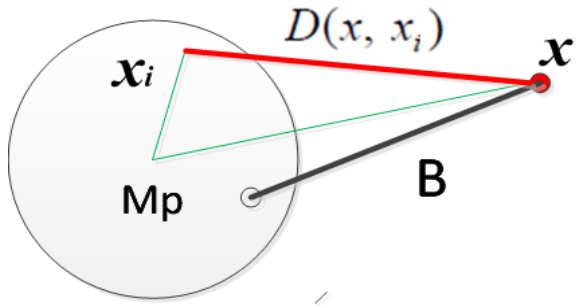
算法规则：

规则 2：找最近样本

如果 x 到 x_i 的距离>已存在的最近点，样本 x_i 被忽略。

$$D(x, x_i) > B$$

利用三角定理 $D(x, x_i) > D(x, M_p) - D(x_i, M_p) > B$



最近邻

快速算法一：快速搜索近邻法

算法流程：

步骤 1-5 搜子集；

步骤 6 搜样本。

k 近邻快速搜索推广：

子集搜索过程与最近邻一致，

样本搜索时，B 存有 k 个最近距离值.

树搜索算法

步骤 1　置 $B=\infty, L=0, p=0$。(L 是当前水平，p 是当前结点)

步骤 2　将当前结点的所有直接后继结点放入一个目录表中，并对这些结点计算 $D(x, M_p)$。

步骤 3　对步骤 2 中的每个结点 p，根据规则 1，如果有 $D(x,M_p)>B+r_p$，则从目录表中去掉 p。

步骤 4　如果步骤 3 的目录表中已没有结点，则后退到前一个水平，即置 $L=L-1$。如果 $L=0$ 则停止，否则转步骤 3。如果目录表中有一个以上的结点存在，则转步骤 5。

步骤 5　在目录表中选择最近结点 p'，它使 $D(x,M_p)$ 最小化，并称该 p' 为当前执行结点，从目录表中去掉 p'。如果当前的水平 L 是最终水平，则转步骤 6。否则置 $L=L+1$，转步骤 2。

步骤 6　对现在执行结点 p' 中的每个 x_i，利用规则 2 作如下检验。如果

$$D(x,M_p) > D(x_i,M_p) + B$$

则 x_i 不是 x 的最近邻，从而不计算 $D(x,x_i)$，否则计算 $D(x,x_i)$。若

$$D(x,x_i) < B$$

置 $NN=i$ 和 $B=D(x,x_i)$。在当前执行结点中所有 x_i 被检验之后，转步骤 3。当算法结束时，输出 x 的最近邻 x_{NN} 和 x 与 x_{NN} 的距离 $D(x,x_{NN})=B$。

最近邻

快速算法二：剪辑近邻法

原理：

通过剪掉边界样本（错误分类样本），缩减样本规模

剪辑规则：

两分剪辑近邻法

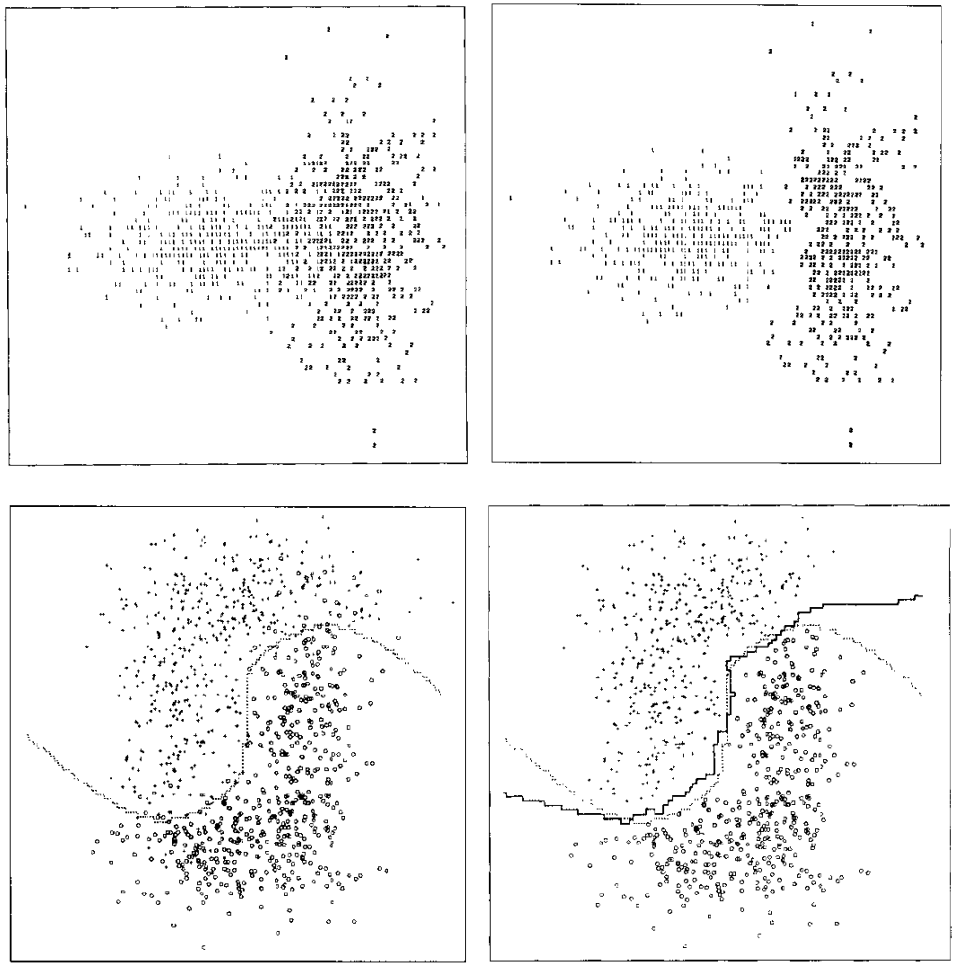
--将训练样本集，分成两个子集 A，B，

--A 做分类参考，对 B 进行剪辑（错分样本被剪掉）

--剪辑后的 B- 作为最终的训练集，训练近邻分类器

最近邻

快速算法二：剪辑近邻法



最近邻

快速算法三：压缩近邻法

原理：去掉中心附近样本，保留错误样本

基本思想：分类中通常被正确分类的样本，较少支持决策，

将常分误的样本保留.

压缩规则：

(1) 初始化，训练集分为 S，G，

S 中仅 1 个样本；G 中 N-1 个；

(2) S 作为训练，分类 G 中第 *i* 个样本；如果错误，将该样本放入 S 中；

(3) 对每一个样本重复（2）

(4) 直到 G 无错分样本，或 G 为空

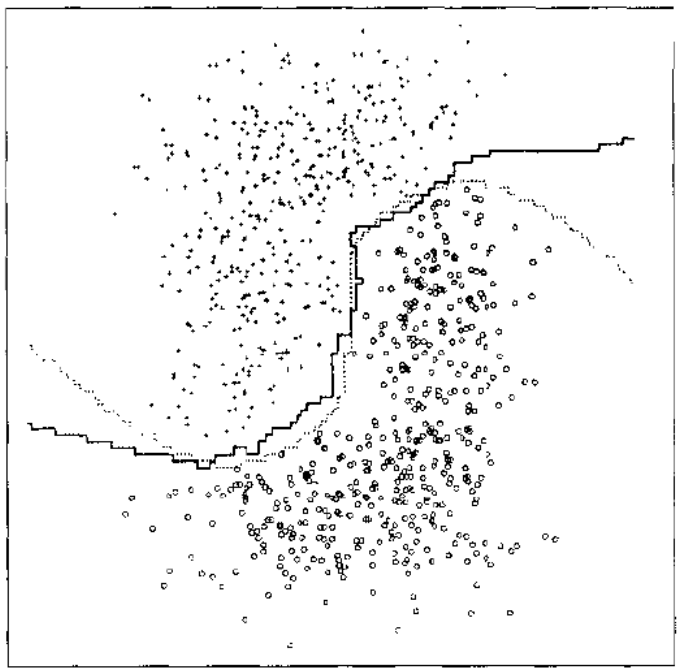
(5) 将 G 中样本放弃，S 是最终压缩样本集

最近邻

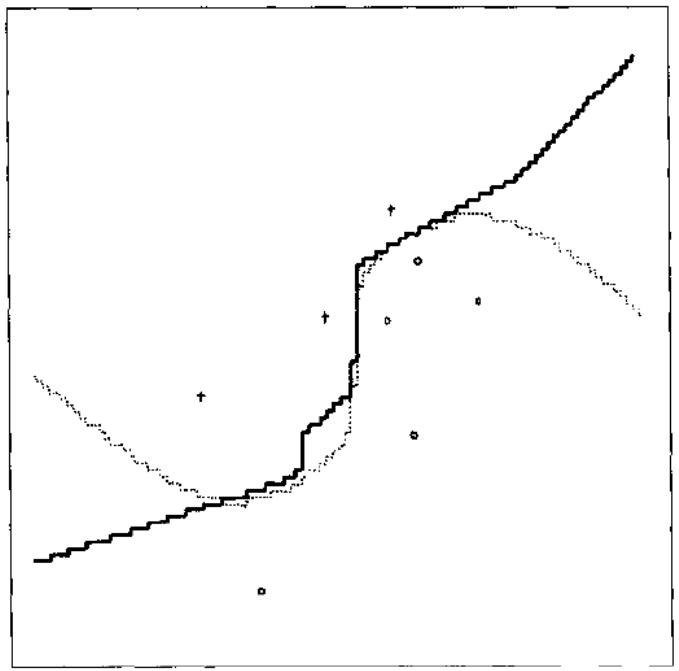
快速算法三：压缩近邻法

剪辑基础上，进行压缩，分类结果：

剪辑



压缩



最近邻

拒绝决策近邻法

原理：对于与各类别相似度较低的样本，不做判断;

优点：在样本压缩时，给 可是可非的样本 机会.

算法 1：可拒绝的 k 近邻法（分类决策）

k 近邻中，各类样本的个数小于 k_i，拒绝分类

算法 2：可拒绝的编辑近邻法（样本压缩）

与编辑近邻法比较，不同之处：除保留正确分类样本外，还保留了拒绝样本。

参考文献

1. 机器学习，周志华，清华大学，2016.

2. 统计学习方法，李航，清华大学，2012.

3. 模式识别（第二版)，边肇祺，张学工等,清华大学出版社，2000.1