> 每个模型都是一个视角

每个结点：一个属性测试

每个分支：测试的一种可能性

每个叶结点：一个预测结果

学习过程：对训练样本的分析来划分属性

预测过程：从根下行到叶

一、算法

输入： 训练集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
属性集 $A = \{a_1, a_2, \ldots, a_d\}$.
过程： 函数 TreeGenerate(D, A)

```
生成结点 node;
if D 中样本全属于同一类别 C then
    将 node 标记为 C 类叶结点; return
end if
if A = ∅ OR D 中样本在 A 上取值相同 then
    将 node 标记为叶结点, 其类别标记为 D 中样本数最多的类; return
end if
从 A 中选择最优划分属性 a*;
for a* 的每一个值 a*^v do
    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;
    if D_v 为空 then
        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
    else
        以 TreeGenerate(D_v, A \ {a*})为分支结点
    end if
end for
```

输出： 以 node 为根结点的一棵决策树

图 4.2 决策树学习基本算法

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

$$a_* = \underset{a \in A}{\arg\min}\ \mathrm{Gini\_index}(D, a)$$

停止条件

- 结点包含的样本都同类
- 当前属性集为空 or 所有样本在所有属性上取值相同

> 标记为叶结点，类别为样本最多的类别
>
> 利用当前结点后验分布

- 结点样本空

> 标记为叶结点，类别为父结点样本最多的类别
>
> 父结点样本分布当作当前的先验分布

标记方式

- 都一样：C
- 当前后验：标记为最多那个
- 空：以父节点后验作为当前先验

> 先验和后验可通过贝叶互换

二、划分选择

1、信息增益

信息熵：度量样本集合纯度的指标，越小，D纯度越高

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

- 集合D中第k类样本占比pk，共|y|类
- 最小值为0(至少2分类)

信息增益：越大，属性a划分出的纯度提升越大

前减后

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

- 属性a有V个可能取值，划分产生V个分支结点
- 理解：原来的一个点信息熵minus新的V个点信息熵加权和
- 缺点：对可能取值多的属性有偏好（肯定会更纯）

2、增益率

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

$$\mathrm{IV}(a)=-\sum_{v=1}^{V}\frac{|D^v|}{|D|}\log_2\frac{|D^v|}{|D|}$$

固有值IV（a）：V越大，IV越大

- 缺点：对可能取值少的属性有偏好
- 启发式：用于筛选高于平均水平的属性

3、基尼指数

基尼值：D中任意两样本不一致概率，越小纯度越高

$$\begin{aligned}\mathrm{Gini}(D)&=\sum_{k=1}^{|\mathcal{Y}|}\sum_{k'\neq k}p_k p_{k'}\\&=1-\sum_{k=1}^{|\mathcal{Y}|}p_k^2\,.\end{aligned}$$

基尼指数

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

三、剪枝处理：防过拟合

1、预剪枝

若结点划分不能带来泛化性能提升则停止划分（标记为叶）

- 优点：减少训练、测试时间开销
- 缺点：贪心策略通病

2、后剪枝

完整树自底向上检查，若子树替换为叶提升泛化性能则替换

- 缺点：训练时间开销大
- 优点：泛化性能往往优于预剪枝

四、连续与缺失值

> 从离散扩展到连续

1、连续值

- 连续属性离散化：对连续属性a设置划分点集合Ta

$$T_a = \left\{ \frac{a^i + a^{i+1}}{2} \mid 1 \leqslant i \leqslant n-1 \right\}$$

- 注意：连续属性划分后还可继续使用

2、缺失值

（1）划分属性选择

> 对每个属性分别找无缺失样本

$$\rho = \frac{\sum_{\boldsymbol{x} \in \tilde{D}} w_{\boldsymbol{x}}}{\sum_{\boldsymbol{x} \in D} w_{\boldsymbol{x}}},$$

$$\tilde{p}_k = \frac{\sum_{\boldsymbol{x} \in \tilde{D}_k} w_{\boldsymbol{x}}}{\sum_{\boldsymbol{x} \in \tilde{D}} w_{\boldsymbol{x}}} \quad (1 \leqslant k \leqslant |\mathcal{Y}|)$$

$$\tilde{r}_v = \frac{\sum_{\boldsymbol{x} \in \tilde{D}^v} w_{\boldsymbol{x}}}{\sum_{\boldsymbol{x} \in \tilde{D}} w_{\boldsymbol{x}}} \quad (1 \leqslant v \leqslant V).$$

D~：D中在属性a上无缺失样本子集

p：无缺失样本比例（参与权重计算，默认权重1）

p~k：无缺失样本中第k类比例（参与权重计算，默认权重1）

r~v：无缺失样本中取值为a^v的样本比例（参与权重计算，默认权重1）

$$\begin{aligned}\operatorname{Gain}(D,a) &= \rho \times \operatorname{Gain}(\tilde{D},a) \\ &= \rho \times \left(\operatorname{Ent}\left(\tilde{D}\right) - \sum_{v=1}^{V} \tilde{r}_v \operatorname{Ent}\left(\tilde{D}^v\right)\right)\end{aligned}$$

), 有

$$\operatorname{Ent}(\tilde{D}) = -\sum_{k=1}^{|\mathcal{Y}|} \tilde{p}_k \log_2 \tilde{p}_k \ .$$

（2）划分样本

- 若样本x在划分属性a上取值缺失，同时划入所有子结点
- 样本权值在属性a^v的子结点中调整为r~v*w_x

五、多变量决策树

引子：

把样本对应到多维空间上，决策树相当于找到分类边界，且有明显特点：边界一定与坐标轴垂直。那么对于以下情况，斜划分能大大简化

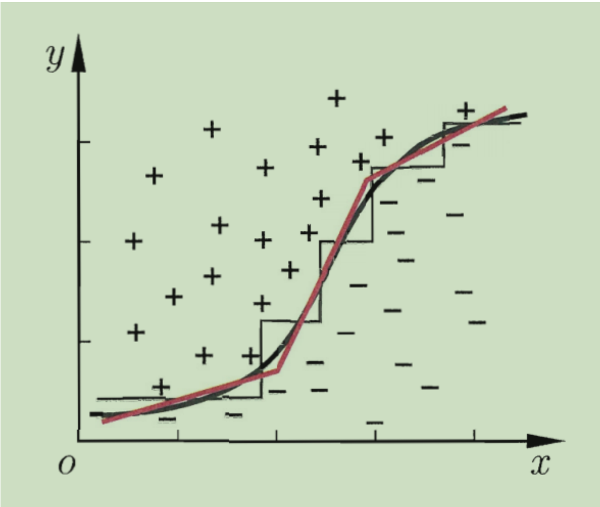

图 4.12　决策树对复杂分类边界的分段近似

举例

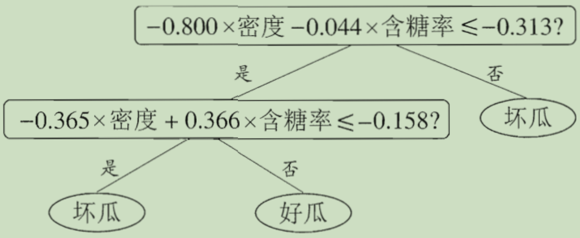

图 4.13 在西瓜数据集 3.0α 上生成的多变量决策树

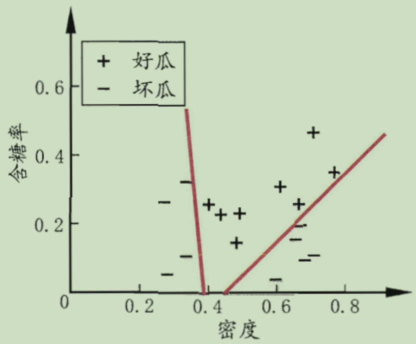

图 4.14 图 4.13 多变量决策树对应的分类边界

方法评价

1、训练含缺失值的处理方法

- 在确定某一属性的分支权重时不考虑有缺失值的样本
- 有缺失值的样本以不同权重进入不同分支
- 计算熵要按权重计算
- 增益算完之后还要乘上p(按缺失比例恢复)
- 构建出的决策树会有一些改变

局限性：

- 样本非独立同分布采样或缺失过多，效果不好
- 需要特殊处理缺失，开销大
- 过拟合，产生原来没有的分支

2、测试含缺失值的处理方法

- 训练集每个分支的分布作为先验，用于预测缺失进入不同分支概率
- 选择概率最大的类别作为预测标签

局限性:

- 过分依赖训练集分布，训练集不能太少
- 开销大

- 标签概率接近时，分类错误概率极大