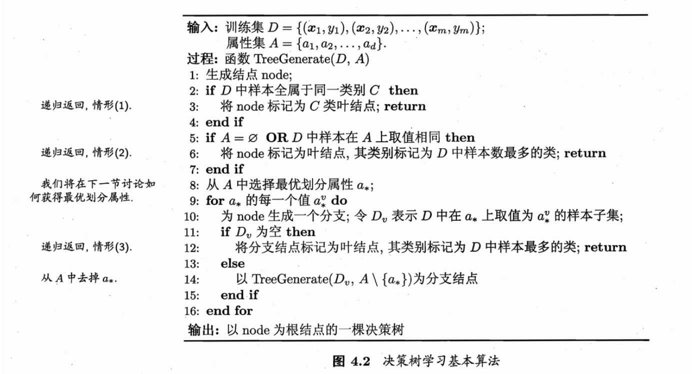
决策树划分选择

引言

我们在上一篇文章中，我们介绍了什么是决策树以及决策树的决策过程。现在我们要考虑另一个重要的东西，即图 4.2 中第 8 行中的最优化分。可以预见的是，决策树中最重要的就是选择，你的先前选择决定了你后来的选择域，所以我们需要做的就是保证你的选择是最优解，它使得你的选择域不断精准，直至接近甚至到达目标值。这就是我们今天要说的“决策树划分选择”。



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)

```
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              递归返回, 情形(1).
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     递归返回, 情形(2).
7: end if
8: 从 A 中选择最优划分属性 a*;                     我们将在下一节讨论如何获得最优划分属性.
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     递归返回, 情形(3).
13:    else
14:        以 TreeGenerate(D_v, A \ {a*})为分支结点          从 A 中去掉 a*.
15:    end if
16: 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

为了实现这个操作，我们有多种算法比如：ID3、C4.5和CART，这些在前置导学也都提到过。

ID3：信息论与决策树划分选择

在前置导学中，我们提到了有关于信息论的一些基本概念，比如：

1. 信息量，一个事件越罕见，一句话中某个单词出现次数越少（出现频率），出现概率 $p(x)$ 越小，信息量 $I(x)$ 越大；
2. 信息熵，信息量的期望值，度量一个系统平均的不确定性 $H(X)=-\sum_{i=1}^{n}p(x_i)\log p(x_i)$，一句话的信息熵越大，信息量越丰富，可预测性越低，不确定性越高；
3. 条件熵，已知随机变量 X，另一个随机变量 Y 的剩余不确定性，即 $H(Y|X)=\sum_{x}p(x)H(Y|X=x)$；
4. 互信息，知道 X 后，Y 的不确定性减少了多少，在这里我们用来处理确定决策树一侧后，另一侧的不确定性减少了多少，即信息熵减少了多少。

1. 信息熵：期望(Information Entropy)

于是，我们就引入了信息论当中的概念，来解决我们的决策树划分选择（祖师爷nb）。

设当前样本集合 D 中第 k 类样本所占的比例为 $p_k\,(k=1,2,\ldots,|\mathcal{Y}|)$，其中 $\mathcal{Y}$ 是类别集合。

定义： 样本集合 D 的信息熵定义为：

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

其中：

1. 符号 $\log_2$ 意味着熵的单位是**比特（二进制么，当然也存在以e为底的）**。
2. 公式前边的符号是为了标准化，将熵值始终保持正值。
3. 当 D 中所有样本都属于同一类别时，不确定性最小，熵为 0。当 D 中样本类别均匀分布时，不确定性最大，熵取得最大值 $\log_2 |\mathcal{Y}|$ （证明可以参考南瓜书P54-57，同时这个观点在上文也有写：信息量越丰富，可预测性越低，不确定性越高）。

如果你不理解，那么我这里还有一个例子：假设一个袋子里有 5 个红球和 5 个蓝球，其熵为

$$\mathrm{Ent}(D) = -(\frac{5}{10}\log_2\frac{5}{10} + \frac{5}{10}\log_2\frac{5}{10}) = -(-0.5 - 0.5) = 1 \text{ (bit)}。$$

如果全是红球，则熵为 0。

2. 互信息：信息增益 (Information Gain)

在信息熵的基础上，我们希望的是通过不断划分，使得划分后子集的**纯度**越来越高，即**熵越来越小**。也就是说，最小化：$\mathrm{Ent}(D) = -\sum_{k=1}^{|\mathcal{Y}|} p_k \log_2 p_k$ 。于是互信息(信息增益)就被引了出来......

信息增益，实质上是特征 A 与数据集 D 的类别标记之间的**互信息**。它表示在已知特征 A 的信息后，**类别 Y 的不确定性减少的程度**。信息增益越大，意味着使用特征 A 进行划分所获得的“纯度提升”越大。这正是 ID3 决策树算法使用的准则。

假设离散特征 A 有 V 个可能的取值 $\{a^1, a^2, ..., a^V\}$。根据特征 A 的取值，可将 D 划分为 V 个子集 $\{D^1, D^2, ..., D^V\}$。每个子集包含 D 中在特征 A 上取值为 a^v 的样本。

定义： 用特征 A 对数据集 D 进行划分所获得的**信息增益**为：

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

在这个公式中：

1. $\mathrm{Ent}(D)$：划分前数据集 D 的不确定性（父节点）。
2. $\sum_{v=1}^{V} \frac{|D^v|}{|D|}\mathrm{Ent}(D^v)$：这是划分后的**条件熵** $\mathrm{Ent}(D|A)$ （子节点）。我们计算了每个分支子集 D^v 的熵，然后按其样本数占总数的比例（即权重）进行加权平均。它代表了在已知特征 A 的条件下，数据集 D 的不确定性。

所以：信息增益 = 父亲节点的熵 - 所有子节点的熵的加权和。

决策树算法会选择信息增益**最大**的特征作为当前划分特征：

$$A_* = \arg\max_A \text{Gain}(D, A)$$

例子：ID3例子

增益率 (Gain Ratio)

但是 ID3 有个缺点：其信息增益准则对**可取值数目较多**的特征有偏好。例如“编号”特征，当我们打算把编号也算入属性用来训练模型时，每个编号就是一个单独的节点，划分后每个子集熵都为 0，信息增益最大，这会影响模型训练，导致过拟合。为了克服这一问题，我们引入了增益率。

来自西瓜书：

实际上，信息增益准则对可取值数目较多的属性有所偏好，为减少这种偏好可能带来的不利影响，著名的 C4.5 决策树算法 [Quinlan, 1993] 不直接使用信息增益，而是使用“增益率”(gain ratio) 来选择最优划分属性．采用与式(4.2)相同的符号表示，增益率定义为

增益率是 C4.5 决策树算法为了减少信息增益对多值特征的偏好而提出的。它将信息增益除以一个称为**“固有值” (Intrinsic Value)** 的标准化因子。

定义： 用特征 A 对数据集 D 进行划分所获得的**信息增益**为：

$$\text{IV}(A) = -\sum_{v=1}^{V} \frac{|D^v|}{|D|} \log_2 \frac{|D^v|}{|D|} \quad \text{（公式1）}$$

为了更好的理解，我们可以把信息熵的公式搬过来做一下改造：

$$\text{Ent}(D) = -\sum_{k=1}^{|\mathcal{Y}|} p_k \log_2 p_k = -\sum_{v=1}^{|\mathcal{Y}|} \frac{|D^k|}{|D|} \text{Ent}(D^k) \quad \text{（公式2）}$$

你再观察这两个公式，是不是很像？事实上，公式 2 的 $\frac{|D^k|}{|D|} = p_k$ ，它代表 k 类样本所占的比例，而公式 1 的 $\frac{|D^v|}{|D|} = p_v$ ，它代表属性 A 取值为 A^v 的样本所占比例。公式 2 是按样本的类别进行标记的信息熵，而公式 1 则是按样本属性的取值计算信息熵。

这段话怎么理解呢？看图：

假设有一个数据集 D，包含 10 个样本，类别标签为"是"或"否"。我们关注特征"天气"，它有3个取值：晴天、雨天、阴天。

- **计算类别标签的熵（公式 2）**：
  - 假设有6个"是"和4个"否"，则：

$$\mathrm{Ent}(D) = -\left(\frac{6}{10}\log_2\frac{6}{10} + \frac{4}{10}\log_2\frac{4}{10}\right) \approx 0.971$$

    这表示数据集的不纯度较高。

- **计算特征"天气"的熵（公式 1）**：
  - 假设天气取值分布：晴天4个、雨天3个、阴天3个，则：

$$\mathrm{IV}(\text{天气}) = -\left(\frac{4}{10}\log_2\frac{4}{10} + \frac{3}{10}\log_2\frac{3}{10} + \frac{3}{10}\log_2\frac{3}{10}\right) \approx 1.571$$

    这表示特征"天气"的取值分布较均匀，不确定性高。

在公式 1 中：特征 A 的取值越多（V 越大），分布越均匀，$\mathrm{IV}(A)$ 的值通常就越大。

C4.5 算法的实际应用： 并非直接选择增益率最大的特征，而是使用一个启发式方法：先从候选特征中找出**信息增益高于平均水平**的那些特征，然后**再从这些特征中选择增益率最高**的。

但是，增益率准则反而对**可取值数目较少**的特征有所偏好。这就有点扯淡了...我们左也不是右也不是......因此，C4.5算法并不是直接选择增益率最大的候选划分属性，而是使用了一个启发式（C4.5决策树算法中，选择划分属性时采用了一种启发式方法，以平衡信息增益（Information Gain）和信息增益率（Gain Ratio）的优缺点） [Quinlan, 1993]：先从候选划分属性中找出信息增益高于平均水平的属性，再从中选择增益率最高的（上图就是例子）。

基尼指数

所以我们需要怎么做呢？那就不得不聊聊我们的 CART 了。CART 是 Classification and Regression Tree 的简称，这是一种著名的决策树学习算法，分类和回归任务都可用。它使用"基尼指数"(Gini index)来选择划分属性。它同样用于度量数据集的纯度。

假设我们从样本数据集是 D，其类别有三个，它们的占比分别是 p_1、p_2、p_3，现在我们从 D 中随机抽取两个样本，这两个样本的类别标记正好一致的概率为：

$$p_1p_1 + p_2p_2 + p_3p_3 = \sum_{k=1}^{|\mathcal{Y}|=3} p_k^2$$

很好理解，而这两个样本的类别标记不一致的概率（这就是"基尼值"）为：

$$p_1p_2 + p_1p_3 + p_2p_1 + p_2p_3 + p_3p_1 + p_3p_2 = \sum_{k=1}^{|\mathcal{Y}|=3}\sum_{k'\neq 1} p_k p_k'$$

你可以很快地发现，实际上两式相加等于 1。所以我们就可以得到下面这个公式：

即，数据集 D 的纯度可用**基尼值**来度量，其定义为：

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

可以知道的是：$\operatorname{Gini}(D)$ 直观反映了从数据集 D 中随机抽取两个样本，其**类别标记不一致**的概率。因此，$\operatorname{Gini}(D)$ 越小（拿到不一致的概率小），数据集 D 的纯度越高（相同度高）。与熵类似，当 D 中所有样本属于同一类时，基尼值为 0。

之后，对于特征 A，其划分后的**基尼指数**定义为各子集的基尼值的加权平均：

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

于是，我们在候选属性集合 A 中，选择那个使得划分后基尼指数最小的属性作为最优划分属性，即：

$$
A_* = \arg\min_{A} \operatorname{Gini\_index}(D, A)
$$