# 微痛学习 决策树


**MLNLP**
深度学习打工人


## 目录

**一、ID3** / 熵、条件熵、信息增益、贪婪搜索

**二、C4.5** / 信息增益比

**三、CART** / 基尼指数、分类树、回归树

**四、ID3, C4.5, CART 比较**

决策树是一类机器学习算法，它是一个能够做**决策**的**树**模型，由若干个节点组成树状结构（下文假设你了解数据结构中的树，并且知道监督学习是什么）。决策树比较简单，通常还会将多棵决策树放到一起，做个集成学习，提高模型拟合能力和预测效果。

决策树的学习过程包括三个步骤：

a）特征选择。不同的特征和预测目标具有不同强度的相关性，选择相关性最强的特征能够有效提高预测效果。

b）节点分裂。训练集会在决策树中按照节点规则分流，如果 节点A 没办法给出一个满意的分类结果，那它就会选择**分裂**，分成 2 个或者多个节点。那么根据什么分裂呢？节点A 会用熵来判断用哪个特征分裂是最优的。

c）剪枝。决策树不加限制地分裂容易产生过拟合现象，**剪枝**可以一定程度地缓解过拟合，提高泛化能力。

决策树学习算法包含特征选择、决策树的生成与决策树的剪枝过程。由于决策树表示一个条件概率分布，所以深浅不同的决策树对应着不同复杂度的概率模型。决策树的生成对应于模型的局部选择，决策树的剪枝对应于模型的全局选择。决策树的生成只考虑局部最优，相对地，决策树的剪枝则考虑全局最优。

决策树的学习算法有多种，常用的有：ID3，C4.5，CART。下面逐个介绍

## 一、Iterative Dichotomiser 3 (ID3)

- **熵：**

在信息论与概率统计中，熵是这样定义的。有事件集合 $Y=\{y_1,y_2,\cdots,y_n\}$ ，事件 $y_i$ 发生的概率为 $p(y_i)$ ，事件集合 $Y$ 的熵为：

$$H(Y) = -\sum_{i=1}^{n} p(y_i) \log(p(y_i))$$

对应到决策树来，假设树 节点A 的训练样本集合为 $D$ ，类别标签为 $k$ 的样本集合用 $C_k$ 表示，用 $|\cdot|$ 表示集合元素数量，那么 节点A 的训练样本集合 $D$ 的熵为：

$$H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} \log \frac{|C_k|}{|D|}$$

因为 $H(D)$ 是由数据估计的，所以此时的熵叫做经验熵。

- **条件熵**

在信息论与概率统计中，条件熵是这样定义的。 $Y$ 和 $p(\cdot)$ 定义如上， $X$ 是另一事件集合， $Y$ 的条件熵是：

$$H(Y|X) = \sum_{i=1}^{n} p(x_i) H(Y|x_i)$$

对应到决策树来，假设已知条件是特征 $A$ 的值， $A$ 包含多个值 $A = \{a_1, a_2, \cdots, a_n\}$，每个 $a_i$ 对应一个子数据集 $D_i$， $D_i$ 按照类别标签 $k$ 可以细分为多个 $D_{ik}$，那么训练样本集合 $D$ 的条件熵为：

$$\begin{aligned} H(D|A) &= \sum_{i=1}^{n} \frac{|D_i|}{|D|} H(D_i) \\ &= -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \sum_{k=1}^{K} \frac{|D_{ik}|}{|D_i|} \log \frac{|D_{ik}|}{|D_i|} \end{aligned}$$

因为 $H(D|A)$ 是由数据估计的，所以此时的条件熵叫做经验条件熵。

- **信息增益**

在信息论与概率统计中，有一个和信息增益等价的概念，叫互信息，它是这样定义的。

$$I(X;Y) = H(Y) - H(Y|X)$$

对应到决策树来，特征 $A$ 对训练样本集合 $D$ 的信息增益为：

$$g(D,A) = H(D) - H(D|A)$$

从直觉上来讲特征 $A$ 能让数据集 $D$ 的不确定度降低，也就是经验熵降低，具体降低多少可以用信息增益衡量，信息增益越大，说明 $A$ 起到的效果越大，信息增益越小，说明起到的效果越小。

- **贪婪搜索**

从根节点开始，对每个特征计算信息增益，贪婪地选择信息增益 $g(D|A)$ 最大的特征 $A_g$，按照 $A_g$ 中的每个值 $a_i$ 生成一个孩子节点，如此递归至信息增益过小。每个叶子节点上都会有最终类别，这个类别为叶子节点上样本数量最多的那个。

## 二、C4.5

- **信息增益比**

当特征取值较多时，信息增益会比较大，但这样的特征并不一定合适，比如 样本ID 这种全不相同的特征。因此设计了信息增益比，如下：

$$
\begin{aligned}
g_R(D, A) &= \frac{g(D, A)}{H_A(D)} \\
&= \frac{g(D, A)}{-\sum_{i=1}^{n} \frac{|D_i|}{|D|} \log \frac{|D_i|}{|D|}}
\end{aligned}
$$

分母 $H_A(D) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \log \frac{|D_i|}{|D|}$ 是 $D$ 关于特征 $A$ 的熵，它能够有效控制信息增益 $g(D, A)$ 对多值特征的倾向。

C4.5算法就是把ID3中的信息增益改进成了信息增益比。

## 三、Classification And Regression Tree (CART)

ID3和C4.5都是多叉树，而CART是二叉树，内部节点的取值为“是”或“否”。并且CART既可用于分类，也可用于回归。

- **基尼指数**

ID3 和 C4.5 都是建立在熵的基础之上，然而熵里面有个很耗时的 $\log$ 计算，如何简化计算提高运算速度呢？可以用一阶泰勒展开式近似。

令 $f(x) = -\ln(x)$，对 $f(x)$ 在 $x = 1$ 处做一阶泰勒展开，得到：

$$
\begin{aligned}
f(x) &= -\ln x \\
&\approx f(1) + f'(1) \cdot (x - 1) \\
&= 1 - x
\end{aligned}
$$

让 $H(Y) = -\sum_{i=1}^{n} p(y_i) \log(p(y_i))$ 中对数的底取 $e$，将泰勒展开式代入得到基尼指数：

$$
\begin{aligned}
Gini(Y) &= \sum_{i=1}^{n} p(y_i)(1-p(y_i)) \\
&= 1-\sum_{i=1}^{n} p(y_i)^2
\end{aligned}
$$

对于二分类，假设属于第 1 类的概率为 $p$，则基尼指数为：

$$
\begin{aligned}
Gini(Y) &= 1-\sum_{i=1}^{n} p(y_i)^2 \\
&= 1-p^2-(1-p)^2 \\
&= 1-p^2-(1-2p+p^2) \\
&= 2p(1-p)
\end{aligned}
$$

CART 是二叉树，节点按照特征 $A$ 是否取某一可能值 $a$，将数据集分割成 $D_1, D_2$，此时的基尼指数为：

$$
Gini(D,A) = \frac{|D_1|}{|D|}Gini(|D_1|) + \frac{|D_2|}{|D|}Gini(|D_2|)
$$

- **分类树**

生成分类树的过程与 ID3 和 C4.5 类似。从根节点开始，对每个特征的每个切分点计算基尼指数，选择最小的那个作为最优特征和最优切分点。现在这个节点就可以分裂了( ･̀ ω ･́ )✧，直到节点样本数量太少或者基尼指数太小或者特征不够了才停止。叶子节点上的最终类别是这个叶子节点上样本数量最多的那个。

- **回归树**

分类树是针对离散的类别目标，回归树是针对连续的数值目标。两种树很像，区别有两点：

a）对切分点的使用方式不同。分类树是把样本按特征值 $=$ 或 $\neq$ 分到两个孩子节点；回归树是把样本按特征值 $\leq$ 或 $>$ 分到两个孩子节点。

b）评价指标不同。分类树是用基尼指数；回归树是用平方误差 $\sum_i(y_i - f(x_i))^2$ 。

为了表示回归树的原理，先定义下数学符号。

设训练数据集 $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$，将 $X = \{x_1, x_2, \cdots, x_n\}$ 划分为 $M$ 个单元 $R_1, R_2, \cdots, R_M$，用 $c_m$ 代表 $R_m$ 单元的输出值。回归树就可以用这些符号表示出来：

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$
$$I(x \in R_m) = \begin{cases} 1 & x \in R_m \\ 0 & x \notin R_m \end{cases}$$

其中 $I(\cdot)$ 是示性函数，事件发生为 1，不发生为 0。

如何划分得到 $R_1, R_2, \cdots, R_M$ 呢？答：遍历一遍所有特征 $A$，找出每个特征最优切分点 $s$，根据全部 $(A, s)$ 划分。还是很模糊，下面用数学来严谨定义下。

现在从根节点开始，对于特征 $A = \{a_1, \cdots, a_n\}$，设切分点为 $s$，切分点两侧的 $R$ 单元为：

$$R_1(A, s) = \{x_i | a_i \leq s\}$$
$$R_2(A, s) = \{x_i | a_i > s\}$$

那么我们的优化目标为最小化两个孩子节点的平方误差：

$$\min_{A,s}\left(\min_{c_1}\sum_{x_i\in R_1(A,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(A,s)}(y_i-c_2)^2\right)$$

内层的 $c$ 可以通过求导得到。当 $c$ 为 $\{y_i|x_i \in R(A,s)\}$ 均值时，内层优化目标取得最小值，即

$$c_1=\frac{1}{N_1}\sum_{x_i\in R_1(A,s)}y_i$$
$$c_2=\frac{1}{N_2}\sum_{x_i\in R_2(A,s)}y_i$$

特征 $A$ 和切分点 $s$ 可以通过遍历求得。在得到这一个节点的最优特征和最优切分点后，可以将训练样本分成两半，传递到两个孩子节点。然后对节点分裂过程疯狂递归，直到满足设定的条件为止，回归树就生成完了ヾ(≧▽≦*)o。

## 四、ID3, C4.5, CART 比较

| | ID3 | C4.5 | CART |
|---|---|---|---|
| 损失函数 | 信息增益 | 信息增益比 | 基尼指标 / 平方误差 |
| 特征重复使用 | ✗ | ✗ | ✓ |
| Normalization | ✗ | ✗ | ✗ |
| 树结构 | 多叉树 | 多叉树 | 二叉树 |
| 适用任务 | 分类 | 分类 | 分类 / 回归 |

## 参考


1. 《统计学习方法》李航


## TO DO

1. 剪枝
2. 缺失值处理
3. 树集成
4. 决策边界
5. 插图

编辑于 2021-05-11 18:16

决策树 cart 统计学习方法