决策树

耿远昊

Datawhale

2021 年 8 月 12 日

信息论基础

树具有天然的分支结构。对于分类问题而言，决策树的思想是用节点代表样本集合，通过某些判定条件来对节点内的样本进行分配，将它们划分到该节点下的子节点，并且要求各个子节点中类别的纯度之和应高于该节点中的类别纯度，从而起到分类效果。

节点纯度反映的是节点样本标签的不确定性。当一个节点的纯度较低时，说明每种类别都倾向于以比较均匀的频率出现，从而我们较难在这个节点上得到关于样本标签的具体信息，其不确定性较高。当一个节点的纯度很高时，说明有些类别倾向于以比较高的频率出现，从而我们能够更有信心地把握这个节点样本标签的具体信息，即确定性较高。

信息论基础

为了定义纯度的概念，我们首先需要思考如何度量不确定性。在生活中，高概率事件代表的不确定性比低概率事件代表的不确定性低，例如：明天太阳从东边升起是必然的，故这个事件的不确定性为 0；而明天下雨并不是必然事件，它相比前一个事件具有更高的不确定性。因此，若定义一个可微函数 $I(p), p \in [0,1]$ 来表示事件 A 发生的概率 $p(A)$ 所代表的不确定性，那么从直觉上应当满足下面四个必要条件，我们将它们称为信息量公理。其中，$A_1, ..., A_n$ 为独立事件。

1. $I(0) = +\infty$
2. $I(1) = 0$
3. $I(p)$ 关于 p 单调递减
4. $I(\prod_{i=1}^{n} p(A_i)) = \sum_{i=1}^{n} I(p(A_i))$

信息论基础

我们已经对信息量公理的前三个条件做出了说明，其第四个条件的含义是：独立事件同时发生的不确定性应当等于这些事件对应发生的不确定性之和，这是非常合理的假设。

根据这些条件，我们容易想到函数

$$I(p) = a\log_b(p)(a(b-1) < 0)$$

符合信息量公理的要求。事实上从充分性的角度而言，它也是能够满足信息量公理的唯一函数。

信息论基础

定理

设 $I(p)$ 在 $[0,1]$ 上可微，则 $I(x)$ 满足信息量公理的充要条件是 $I(p) = a\log_b(p)(a(b-1) < 0)$。

证明

当 $x \in (0,1]$，此时由导数定义有

$$
\begin{aligned}
I^{'}(x) &= \lim_{\Delta x \to 0^-} \frac{I(x+\Delta x) - I(x)}{\Delta x} \\
&= \lim_{\Delta x \to 0^-} \frac{I(\frac{x+\Delta x}{x} \cdot x) - I(x)}{\Delta x}
\end{aligned}
$$

信息论基础

证明 (续)

$$I^{'}(x)=\lim_{\Delta x\to 0^{-}}\frac{I(\frac{x+\Delta x}{x})}{\Delta x}=\frac{1}{x}\lim_{\Delta x\to 0^{-}}\frac{I(1+\frac{\Delta x}{x})}{\frac{\Delta x}{x}}=\frac{1}{x}I^{'}(x)$$

两边积分 $I(x)=I^{'}(1)\ln x+C, x\in(0,1]$，代入 $I(1)=0$ 得 $C=0$，从而

$$I(x)=I^{'}(1)\ln(x)=\frac{I^{'}(1)}{\log_b e}\log_b x$$

记 $a=\frac{I^{'}(1)}{\log_b e}$，由单调性可知 $I^{'}(1)<0$。当 $b>1$ 时有 $\log_b e>0$，即 $a<0$；当 $b<1$ 时有 $\log_b e<0$，即 $a>0$。因此，符合信息量公理的函数只能是 $I(p)=a\log_b(p)(a(b-1)<0)$。

信息论基础

我们已经知道了信息量对应函数的形式，那么究竟应该如何选取合适的 a 和 b 呢？对于一个以概率为 p 发生的事件 A 而言，我们可以选择一种二进制编码的方式来记录它的信息：当 $p = \frac{1}{4}$ 时，我们可以认为事件 A 的发生本质上是某个随机变量的一种状态，且该随机变量会以等概率出现 4 种状态，那么我们就可以用 00、01、10 和 11 来进行状态信息的记录；当 $p = \frac{1}{8}$ 时，我们需要用 000、001、010、011、100、101、110、111 的编码来记录。因此，p 越小则不确定性越高，需要消耗的编码长度越大。此时，编码种类的数量即为 $\frac{1}{p}$，事件 A 的二进制编码长度代表的不确定性大小就是 $\log_2 \frac{1}{p}$。因此，我们可以取 $I(p)$ 中的 a 为 -1，且取 b 为 2，用 $I(p) = -\log_2(p)$ 来代表度量不确定性的指标。

信息论基础

先前我们讨论了随机变量取定某个值情况下不确定性的度量，那么如果要定义一个随机变量 X 的平均不确定性，只需要对这个随机变量按照对应的概率密度分布 $p(x)$ 取期望即可，我们将其称为分布的信息熵（Information Entropy）$H(X)$（熵是一种反应系统不确定性的指标，由于此处指随机变量信息的不确定性，故称为信息熵），即

$$H(X) = \mathbb{E}_X I(p) = \mathbb{E}_{X\sim p(x)}[-\log_2 p(X)]$$

对于定义在有限状态集合 $\{x_1, ..., x_K\}$ 上的离散变量而言，对应信息熵的最大值在离散均匀分布时取到，最小值在单点分布时取到。此时，离散信息熵为

$$H(X) = -\sum_{k=1}^{K} p(x_k)\log_2 p(x_k)$$

信息论基础

首先，我们需要定义当 p 时 $p\log_2 p \triangleq 0$，原因在于

$$\lim_{p\to 0^+} p\log p = \lim_{p\to 0^+} \frac{\log p}{1/p} = \lim_{p\to 0^+} \frac{1/p}{-1/p^2} = \lim_{p\to 0^+} -p = 0$$

离散熵的极值问题是带有约束的极值问题，记 $p_k = P(X = x_k)$ 和 $\mathbf{p} = [p_1, ..., p_K]^T$，则约束条件为 $\mathbb{1}^T\mathbf{p} = 1$，拉格朗日函数为

$$L(\mathbf{p}) = -\mathbf{p}^T \log \mathbf{p} + \lambda(\mathbb{1}^T\mathbf{p} - 1)$$

求偏导数后可解得 $\mathbf{p}^* = [\frac{1}{K}, ..., \frac{1}{K}]$，此时 $\mathbb{E}_X I(p) = \log K$。

信息论基础

对于离散随机变量 X，由于 $p(X) \in [0,1]$，故 $-\log_2 p(X) \geq 0$，从而 $\mathbb{E}_X I(p) \geq 0$。注意到对于 $\forall k \in \{1,...,K\}$，当 $p_k = 1$，即 $p_{k'} = 0(k' \in \{1,...,K\}/k)$ 时，$H(X) = 0$。因此，离散信息熵的最小值为 0 且在单点分布时取到。由于 $\mathbf{p}^*$ 是极值问题的唯一解，因此离散熵的最大值为 $\log K$ 且在离散均匀分布时取到。

这些结论都是与直觉高度吻合的。单点分布的取值被唯一确定，因此随机变量的不确定性为 0；在给定状态集合数量下，分布越是均匀，则随机变量的不确定性越是大；当 $K \to +\infty$ 时，离散均匀分布的熵为无穷大，一个合理的解释是：随着取值集合元素数量的增加，我们对每一个元素平均而言的信息把握程度就越少，不确定性就越大。

信息论基础

由于在决策树的分裂过程中，我们不但需要考察本节点的不确定性或纯度，而且还要考察子节点的平均不确定性或平均纯度来决定是否进行分裂。子节点的产生来源于决策树分支的条件，因此我们不但要研究随机变量的信息熵，还要研究在给定条件下随机变量的平均信息熵或条件熵（Conditional Entropy）。从名字上看，条件熵就是条件分布的不确定性，那么自然可以如下定义条件熵 $H(X|Y)$ 为

$$\mathbb{E}_Y[\mathbb{E}_{X|Y}[-\log_2 p(X|Y)]]$$

对于离散条件熵，设 Y 所有可能的取值为 $\{y_1, ..., y_M\}$，上式可展开为

$$-\sum_{m=1}^{M} p(y_m) \sum_{k=1}^{K} p(x_k|Y = y_m) \log_2 p(x_k|Y = y_m)$$

信息论基础

有了信息熵和条件熵的基础，我们就能很自然地定义信息增益（Information Gain），即节点分裂之后带来了多少不确定性的降低或纯度的提高。当给定随机变量 Y 的取值 y 时，随机变量 X 的不确定性减少量为

$$G(X,Y) = H(X) - H(X|Y)$$

从直觉上说，随机变量 X 的信息增益一定是非负的，因为我们额外地知道了随机变量 Y 的取值，这个条件降低了 X 的不确定性。下面我们就从数学角度来证明其正确性。

信息论基础

$$
\begin{aligned}
G(X,Y) &= \mathbb{E}_X[-\log_2 p(X)] - \mathbb{E}_Y[\mathbb{E}_{X|Y}[-\log_2 p(X|Y)]] \\
&= -\sum_{k=1}^{K} p(x_k)\log_2 p(x_k) \\
&\quad + \sum_{m=1}^{M} p(y_m) \sum_{k=1}^{K} p(x_k|Y=y_m)\log_2 p(x_k|Y=y_m) \\
&= -\sum_{k=1}^{K}[\sum_{m=1}^{M} p(x_k,y_m)]\log_2 p(x_k) \\
&\quad + \sum_{k=1}^{K}\sum_{m=1}^{M} p(y_m)\frac{p(x_k,y_m)}{p(y_m)}\log_2 \frac{p(x_k,y_m)}{p(y_m)}
\end{aligned}
$$

信息论基础

$$G(X,Y) = \sum_{k=1}^{K}\sum_{m=1}^{M} p(x_k, y_m)[\log_2 \frac{p(x_k, y_m)}{p(y_m)} - \log_2 p(x_k)]$$

$$= -\sum_{k=1}^{K}\sum_{m=1}^{M} p(x_k, y_m) \log \frac{p(x_k)p(y_m)}{p(x_k, y_m)}$$

上式说明信息增益 $G(X,Y)$ 就是 $p(x,y)$ 和 $p(x)p(y)$ 的 KL 散度，而 KL 散度的非负性由 Jensen 不等式可得：

$$G(X,Y) \geq -\log_2[\sum_{k=1}^{K}\sum_{m=1}^{M} p(x_k, y_m)\frac{p(x_k)p(y_m)}{p(x_k, y_m)}]$$

$$= -\log_2[\sum_{k=1}^{K}\sum_{m=1}^{M} p(x_k, y_m)] = 0$$

信息论基础

上式的取等条件为 $p(x,y)=p(x)p(y)$，其实际意义为随机变量 X 和 Y 独立。这个条件同样与直觉相符合，因为如果 X 和 Y 独立，那么意味着我们无论是否知道 Y 的信息，都不会对 X 的不确定性产生影响，此时信息增益为 0。

用信息增益的大小来进行决策树的节点分裂时，由于真实的分布函数未知，故用 $p(x)$ 和 $p(x|y)$ 的经验分布（即频率）来进行概率的估计。若节点 N 每个分支下的样本数量为 $D_1,...,D_M$，记 $\tilde{p}(y_m)=\frac{D_m}{\sum_{m'=1}^{M}D_{m'}}(m\in\{1,...,M\})$，$\tilde{p}(x_k)$ 和 $\tilde{p}(x_k|y_m)$ 分别为节点中第 k 个类别的样本占节点总样本的比例和第 m 个子节点中第 k 个类别的样本数量占该子节点总样本的比例，则节点 N 分裂的信息增益定义为

$$G_N(X,Y)=-\sum_{i=1}^{K}\tilde{p}(x_k)\log\tilde{p}(x_k)+\sum_{m=1}^{M}\tilde{p}(y_m)\sum_{k=1}^{K}\tilde{p}(x_k|y_m)\log_2\tilde{p}(x_k|y_m)$$

分类树的节点分裂

对于每个节点进行分裂决策时，我们会抽出 max_features 个特征进行遍历以比较信息增益的大小。特征的类别可以分为三种情况讨论：类别特征、数值特征和含缺失值的特征，它们各自的处理方法略有不同。

对于类别特征而言，给定一个阈值 ϵ，树的每一个节点会选择最大信息增益 $G_N^{max}(X,Y)$ 对应的特征进行分裂，直到所有节点的相对最大信息增益 $\frac{D_N}{D_{all}}G_N^{max}(X,Y)$ 小于 ϵ，D_N 和 D_{all} 分别指节点 N 的样本个数和整个数据集的样本个数，这种生成算法称为 ID3 算法。在 sklearn 中，ϵ 即为 min_impurity_decrease。

分类树的节点分裂

C4.5 算法在 ID3 算法的基础上做出了诸多改进，包括但不限于：处理数值特征、处理含缺失值的特征、使用信息增益比代替信息增益、提出处理带权样本的方法以及给出树的剪枝策略。其中，剪枝策略将在第 4 节进行讲解，下面先对前 4 个改进的细节来进行介绍。

在处理节点数值特征时，可以用两种方法来将数值特征通过分割转化为类别，它们分别是最佳分割法和随机分割法，分别对应了 sklearn 中 splitter 参数的 best 选项和 random 选项。

随机分割法下，取 $s \sim U[y_{min}, y_{max}]$，其中 $U[y_{min}, y_{max}]$ 代表特征最小值和最大值范围上的均匀分布，将节点样本按照特征 $\mathbf{y}$ 中的元素是否超过 s 把样本划分为两个集合，这等价于把数值变量转换为了类别变量。此时，根据这两个类别来计算树节点分裂的信息增益，并将它作为这个数值特征分裂的信息增益。

分类树的节点分裂

最佳分割法下，依次令 s 取遍所有的 $y_i(i = 1, ..., D_N)$，将其作为分割点，按照特征 $\mathbf{y}$ 中的元素是否超过 s 把样本划分为两个集合，计算所有 s 对应信息增益的最大值，并将其作为这个数值特征分裂的信息增益。

C4.5 算法处理缺失数据的思想非常简单，样本的缺失值占比越大，那么对信息增益的惩罚就越大，这是因为缺失值本身就是一种不确定性成分。设节点 N 的样本缺失值比例为 γ，记非缺失值对应的类别标签和特征分别为 $\tilde{X}$ 和 $\tilde{Y}$，则修正的信息增益为

$$\tilde{G}(X, Y) = (1 - \gamma)G(\tilde{X}, \tilde{Y})$$

当数据完全缺失时 $\gamma = 1$，信息增益为 0；当数据没有缺失值时 $\gamma = 0$，信息增益与原来的值保持一致。

分类树的节点分裂

C4.5 算法还能够处理带权重的样本：设所有节点的样本权重为 $\mathbf{w}^{(D)} = (w_1^{(D)}, ..., w_n^{(D)})$，其中 n 为全体样本的个数；当前分裂节点 N 的样本权重为 $\mathbf{w}^{(N)} = (w_1^{(N)}, ..., w_{n_N}^{(N)})$，其中 n_N 为当前节点的样本个数；当前节点按照特征 $\mathbf{y}$ 分裂的第 m 个子节点权重为 $\mathbf{w}^{(m)} = (w_1^{(m)}, ..., w_{n_m}^{(m)})$，其中 n_m 为该子节点的样本个数。

此时可得到带权的信息增益为

$$G_N(X,Y) = -\sum_{i=1}^{K} \tilde{p}(x_k) \log \tilde{p}(x_k) + \sum_{m=1}^{M} \frac{\sum_{i=1}^{n_m} w_i^{(m)}}{\sum_{i=1}^{n_N} w_i^{(N)}} \tilde{p}(y_m) \tilde{p}(x_k|y_m) \log \tilde{p}(x_k|y_m)$$

分类树的节点分裂

先前提到了 min_impurity_decrease 参数，$\frac{D_N}{D_{all}} G_N^{max}(X, Y)$ 的值会与此阈值进行对比以决定节点 N 是否分裂。对于带权样本，我们可以将相对最大信息增益修正为

$$\frac{\sum_{i=1}^{n_N} w_i^{(N)}}{\sum_{i=1}^{n} w_i^{(D)}} G_N^{max}(X, Y)$$

此处，我们需要注意到当样本的权重被设为全 1 时，加权信息增益和修正的阈值对比值与原来的定义完全一致，具有越高权重的样本就越容易对模型的分裂决策产生影响。

分类树的节点分裂

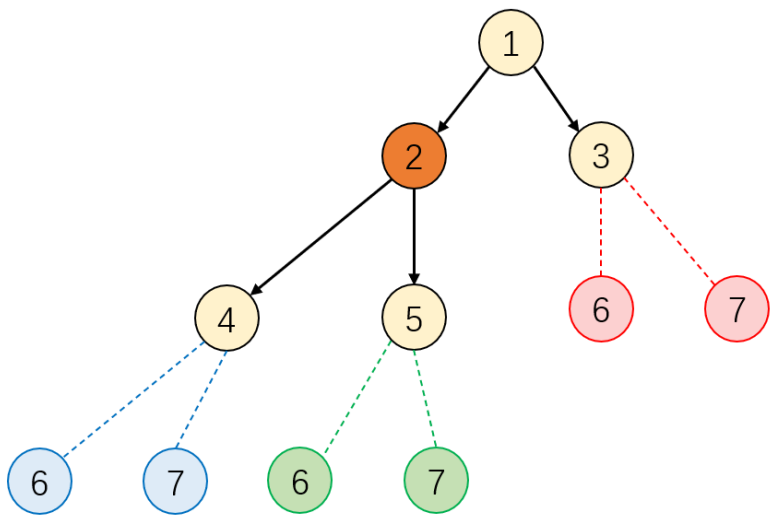
在 C4.5 算法中，使用了信息增益比来代替信息增益，其原因在于信息增益来选择的决策树对类别较多的特征具有天然的倾向性，例如当某一个特征 Y（身份证号码、学号等）的类别数恰好就是样本数量时，此时由于 $H(X|Y)=0$，即 $G(X,Y)$ 达到最大值，因此必然会优先选择此特征进行分裂，但这样的情况是非常不合理的。

我们在第 1 节已经证明了，在类别占比均匀的情况下，类别数越多则熵越高，因此我们可以使用特征对应的熵来进行惩罚，即熵越高的变量会在信息增益上赋予更大程度的抑制，由此我们可以定义信息增益比为

$$G^R(X,Y)=\frac{G(X,Y)}{H(Y)}$$

分类树的节点分裂

在前面的部分中，我们讨论了单个节点如何选取特征进行分裂，但没有涉及到树节点的分裂顺序。例如下图所示，假设当前已经处理完了节点 2 的分裂，所有黄色节点（包括 2 号节点）都是当前已经存在的树节点，那么我们接下来究竟应该选取叶节点 3 号、4 号和 5 号中的哪一个节点来继续进行决策以生成新的叶节点 6 号和 7 号？



分类树的节点分裂

在 sklearn 中提供了两种生长模式，它们分别被称为深度优先生长和最佳增益生长，当参数 max_leaf_nodes 使用默认值 None 时使用前者，当它被赋予某个数值时使用后者。

深度优先生长采用深度优先搜索的方法：若当前节点存在未搜索过的子节点，则当前节点跳转到子节点进行分裂决策；若当前节点为叶节点，则调转到上一层节点，直到根节点不存在未搜索过的子节点为止。对上图而言，当前节点为 2 号，它的两个子节点 4 号和 5 号都没有被搜索过，因此下一步则选择两个节点中的一个进行跳转。在底层数据结构上，由于深度优先生长采用了“先进后出”的节点搜索模式，故使用栈（Stack）结构。当决策树使用最佳增益生长时，每次总是选择会带来最大相对信息增益的节点进行分裂，直到叶节点的最大数量达到 max_left_nodes。

CART 树

CART（Classification And Regression Tree）是一棵二叉树，它既能处理分类问题，又能够处理回归问题。值得注意的是，在 sklearn 中并没有实现处理类别特征和处理缺失值的功能，前者是因为多个类别的特征会产生多叉树，后者是因为 sklearn 认为用户应当自己决定缺失值的处理而不是交给模型来决定。

对于回归树而言，每个叶节点输出的不再是类别而是数值，其输出值为该叶节点所有样本标签值的均值。在每次分裂时，我们希望不同的子节点之间的差异较大，但每个子节点内部的差异较小。此时，分割策略仍然可以采用随机分割法或最佳分割法，只是现在不再以熵（条件熵）来评价节点（子节点）的纯度。

CART 树

我们应当如何定义回归树的节点纯度？对于数值标签而言，我们可以认为节点间元素大小越接近则纯度越高，因此可以考虑使用均方误差（MSE）或平均绝对误差（MAE）来替换熵和条件熵的位置。

设节点 N 的样本标签为 $y_1^{(D)},...,y_N^{(D)}$，左右子节点的样本个数分别为 $y_1^{(L)},...,y_{N_R}^{(L)}$ 和 $y_1^{(R)},...,y_{N_R}^{(R)}$，记 $\bar{y}^{(D)}=\frac{1}{N}\sum_{i=1}^{N}y_i^{(D)}$、$\bar{y}^{(L)}=\frac{1}{N_L}\sum_{i=1}^{N_L}y_i^{(L)}$ 和 $\bar{y}^{(R)}=\frac{1}{N_R}\sum_{i=1}^{N_R}y_i^{(R)}$ 分别为节点 N 的样本标签均值、左子节点的样本标签均值和右子节点的样本标签均值，再记 $\tilde{y}^{(D)}$、$\tilde{y}^{(L)}$ 和 $\tilde{y}^{(R)}$ 分别为节点 N 的样本标签中位数、左子节点的样本标签中位数和右子节点的样本标签中位数。

CART 树

此时，两者的信息增益可以分别定义为

$$G^{MSE}(X,Y) = -\sum_{i=1}^{N}(y_i^{(D)} - \bar{y}^{(D)})^2 + \frac{N_L}{N}\sum_{i=1}^{N_L}(y_i^{(L)} - \bar{y}^{(L)})^2 + \frac{N_R}{N}\sum_{i=1}^{N_R}(y_i^{(R)} - \bar{y}^{(R)})^2$$

$$G^{MAE}(X,Y) = -\sum_{i=1}^{N}|y_i^{(D)} - \tilde{y}^{(D)}| + \frac{N_L}{N}\sum_{i=1}^{N_L}|y_i^{(L)} - \tilde{y}^{(L)}| + \frac{N_R}{N}\sum_{i=1}^{N_R}|y_i^{(R)} - \tilde{y}^{(R)}|$$

CART 树

当样本带有权重时，加权信息增益定义为

$$G_{\mathbf{w}}^{MSE}(X,Y) = -\sum_{i=1}^{N}(y_i^{(D)} - \bar{y}^{(D)})^2 + \frac{\sum_{i=1}^{N_L} w_i^{(N)}}{\sum_{i=1}^{N} w_i^{(N_L)}} \sum_{i=1}^{N_L}(y_i^{(L)} - \bar{y}^{(L)})^2 + \frac{\sum_{i=1}^{N_R} w_i^{(N)}}{\sum_{i=1}^{N} w_i^{(N_R)}} \sum_{i=1}^{N_R}(y_i^{(R)} - \bar{y}^{(R)})^2$$

$$G_{\mathbf{w}}^{MAE}(X,Y) = -\sum_{i=1}^{N}|y_i^{(D)} - \tilde{y}^{(D)}| + \frac{\sum_{i=1}^{N_L} w_i^{(N)}}{\sum_{i=1}^{N} w_i^{(N_L)}} \sum_{i=1}^{N_L}|y_i^{(L)} - \tilde{y}^{(L)}| + \frac{\sum_{i=1}^{N_R} w_i^{(N)}}{\sum_{i=1}^{N} w_i^{(N_R)}} \sum_{i=1}^{N_R}|y_i^{(R)} - \tilde{y}^{(R)}|$$

CART 树

当处理分类问题时，虽然 ID3 或 C4.5 定义的熵仍然可以使用，但是由于对数函数 log 的计算代价较大，CART 将熵中的 log 在 $p = 1$ 处利用一阶泰勒展开，基尼系数定义为熵的线性近似，即由于

$$H(X) = \mathbb{E}_X I(p) = \mathbb{E}_X[-\log_2 p(X)] \approx \mathbb{E}_X[1 - p(X)]$$

从而定义基尼系数为

$$\begin{aligned}\mathrm{Gini}(X) &= \mathbb{E}_X[1 - p(X)] \\ &= \sum_{k=1}^{K} \tilde{p}(x_k)(1 - \tilde{p}(x_k)) \\ &= 1 - \sum_{k=1}^{K} \tilde{p}^2(x_k)\end{aligned}$$

CART 树

类似地定义条件基尼系数为

$$\begin{aligned}\text{Gini}(X|Y) &= \mathbb{E}_Y[\mathbb{E}_{X|Y}1 - p(X|Y)] \\ &= \sum_{m=1}^{M}\tilde{p}(y_m)\sum_{k=1}^{K}[\tilde{p}(x_k|y_m)(1-\tilde{p}(x_k|y_m))] \\ &= \sum_{m=1}^{M}\tilde{p}(y_m)[1-\sum_{k=1}^{K}\tilde{p}^2(x_k|y_m)]\end{aligned}$$

从而引出基于基尼系数的信息增益为

$$G(X,Y) = \text{Gini}(X) - \text{Gini}(X|Y)$$

决策树的剪枝

决策树具有很强的拟合能力，对于任何一个没有特征重复值的数据集，决策树一定能够在训练集上做到分类错误率或均方回归损失为 0，因此我们应当通过一些手段来限制树的生长，这些方法被称为决策树树的剪枝方法。其中，预剪枝是指树在判断节点是否分裂的时候就预先通过一些规则来阻止其分裂，后剪枝是指在树的节点已经全部生长完成后，通过一些规则来摘除一些子树。

在 sklearn 的 CART 实现中，一共有 6 个控制预剪枝策略的参数，它们分别是最大树深度 max_depth、节点分裂的最小样本数 min_samples_split、叶节点最小样本数 min_samples_leaf、节点样本权重和与所有样本权重和之比的最小比例 min_weight_fraction_leaf、最大叶节点总数 max_leaf_nodes 以及之前提到的分裂阈值 min_impurity_decrease。

决策树的剪枝

后剪枝过程又称作 MCCP 过程，即 Minimal Cost-Complexity Pruning，它由参数 ccp_alpha 控制，记其值为 α。一般而言，树的叶子越多就越复杂，为了抑制树的生长，我们定义以节点 N 为根节点的树 T^N 的复杂度为该树的叶节点数量 $|T^N|$。设树 T 的剪枝度量为

$$R_\alpha(T^N) = R(T^N) + \alpha|T^N|$$

其中，$R(T^N)$ 代表各个叶子节点的条件熵/条件基尼系数之和（分类问题）或均方误差/平均绝对误差之和（回归问题），即 MCCP 中的 Cost 部分，$\alpha|T^N|$ 对应的就是 Complexity 部分。

决策树的剪枝

对于树的单个节点而言，由于此时节点数为 1，故其剪枝度量为 $R_\alpha(Node^N) = R(Node^N) + \alpha$。树剪枝的思想在于，如果对于决策树某一个节点为根的子树，其根的剪枝度量低于该子树的剪枝度量，那么这个根节点就没有必要分裂，即砍掉这棵子树中除了根节点以外的所有节点。

此时，我们可以得到剪枝的依据为

$$R_\alpha(Node^N) \leq R_\alpha(T^N)$$

这等价于

$$R(Node^N) + \alpha \leq R(T^N) + \alpha|T^N|$$

决策树的剪枝

对上式进行移项后可得

$$E(Node^N) = \frac{R(Node^N) - R(T)}{|T^N| - 1} \leq \alpha$$

这个条件表明只要 $E(Node^N)$ 的值小于给定的参数 cpp_alpha，那么这个节点下的所有节点都会被删除。事实上在 sklearn 中，在树完全生成后就会把所有节点的 $E(Node^N)$ 值进行记录，每次剪枝都会分别查看所有非叶子节点的树节点对应的 $E(Node^N)$ 值，并且对具有最小 $E(Node^N)$ 值的非叶子节点进行剪枝，直到所有节点的 $E(Node^N)$ 值都大于给定的 cpp_alpha。

习题（A）

1. ID3 树算法、C4.5 树算法和 CART 算法之间有何异同？
2. 什么是信息增益率？它衡量了什么指标？它有什么缺陷？
3. sklearn 决策树中的 random_state 参数控制了哪些步骤的随机性？
4. 决策树如何处理连续变量和缺失变量？
5. 基尼系数是什么？为什么要在 CART 中引入它？
6. 什么是树的预剪枝和后剪枝？具体分别是如何操作的？

习题（B）

1. 在一般的机器学习问题中，我们总是通过一组参数来定义模型的损失函数，并且在训练集上以最小化该损失函数为目标进行优化。请问对于决策树而言，模型优化的目标是什么？
2. 如何理解 min_samples_leaf 参数能够控制回归树输出值的平滑程度？
3. 我们在第 4 节提到决策树具有很强的拟合能力，对于任何一个没有特征重复值的数据集，它一定能够在训练集上做到分类错误率或均方回归损失为 0。为什么？
4. 为什么采用深度优先生长策略的决策树应当使用“先进先出”的策略？

习题（B）

5. 对信息熵中的 log 函数在 $p=1$ 处进行一阶泰勒展开可以近似为基尼系数，那么如果在 $p=1$ 处进行二阶泰勒展开我们可以获得什么近似指标？请写出对应指标的信息增益公式。
6. 除了信息熵和基尼系数之外，我们还可以使用节点的

$$1-\max_k p(X=x_k)$$

和第 m 个子节点的

$$1-\max_k p(X=x_k|Y=y_m)$$

来作为衡量纯度的指标。请解释其合理性并给出相应的信息增益和加权信息增益公式。

习题（B）

7. 在讨论缺失值对信息增益的惩罚时，我们直接使用了 $1-\gamma$ 作为惩罚系数，其中 γ 为缺失值的比例。如果将系数替换为 $1-\gamma^2$，请问对缺失值而言是加强了惩罚还是削弱了惩罚？
8. 如果将树的生长策略从深度优先生长改为广度优先生长，假设其他参数保持不变的情况下，两个模型对应的结果输出可能不同吗？
9. 请实现参数与 sklearn 一致的 DecisionTreeClassifier 类，其成员函数包括 fit、predict 和 predict_proba，同时需给出 feature_importances_ 指标。
   - predict_proba 返回的是测试样本所在叶节点的各类别比例。
   - feature_importances_ 指每个特征的重要性，对于某个特征而言，其特征重要性等于决策树中根据该特征分裂而产生的相对信息增益之和。

习题（B）

10. 假设当前我们需要处理一个分类问题，请问对输入特征进行归一化会对树模型的类别输出产生影响吗？请解释原因。
11. sklearn 提供了 class_weight 参数来处理非平衡样本。设每个类别的样本数量为 $n_1, ..., n_K$，第 i 个样本的类别、样本权重和类别权重分别为 k、w_i 和 w_i^c。当 class_weight 值是形式为 {class_label: class_weight} 的字典时，样本权重被调整为 $w_i \cdot w_i^c$；当 class_weight 值是字符串“balanced”，样本权重被调整为 $w_i \cdot \frac{\sum_{k'=1}^{K} n_{k'}}{K \cdot n_k}$；否则 w_i 不变。现有样本 x_1、x_2 和 x_3 的样本权重为 $[20, 30, 10]$，类别分别是 0、0 和 1，且给定 class_weight={0:40, 1:60}，请计算调整后的样本权重。

习题（B）

12. X, Y 的联合熵为 $H(X,Y) = \mathbb{E}_{(X,Y)\sim p(x,y)}[-\log_2 p(X,Y)]$
    - 请证明如下关系：
      - $G(X,Y) = H(Y) - H(Y|X)$
      - $G(X,Y) = H(X) + H(Y) - H(X,Y)$
      - $G(X,Y) = H(X,Y) - H(X|Y) - H(Y|X)$
    - 下图被分为了 A、B 和 C 三个区域。若 AB 区域代表 X 的不确定性，BC 区域代表 Y 的不确定性，那么 $H(X)$、$H(Y)$、$H(X|Y)$、$H(Y|X)$、$H(X,Y)$ 和 $G(X,Y)$ 分别指代的是哪片区域？