樹模型

Tree Models

VEAGER

2021 年 6 月 11 日

1 决策树 Decision Tree

1.1 分裂指标

1.1.1 离散特征分裂指标

```
import os
import numpy as np
```

在分类任务中，定义数据集 $D=\{\boldsymbol{x_n},y_n\}_{n=1}^{N}$，$N=|D|$ 表示样本容量（样本总数）；$x_n\in\mathbb{R}^P$ 为输入样本，样本维度为 P；$y_n\in\mathbb{R}$ 为输出样本，样本维度为 1；特征集 $\mathbb{A}=\{A_1,A_2,\cdots,A_p,\cdots,A_P\}$。

设输出样本 $y_n,n=1,2,...,N$ 有 K 个类，每一类的的集合为 C_k，其样本数量为 $|C_k|$，则有：

$$D=\bigcup_{k=1}^{K}C_k \tag{1.1}$$

$$N=|D|=\sum_{k=1}^{K}|C_k| \tag{1.2}$$

对于某一特征 $A_p(p=1,2,...,P)$（简写为 A），设特征 A 有 M 个不同的取值 $\{a_1,a_2,...,a_M\}$，根据特征 A 的取值将数据集 D 划分为 M 个子集 $D_1,D_2,...,D_M$，则有：

$$D=\bigcup_{m=1}^{M}D_m \tag{1.3}$$

$$N=|D|=\sum_{m=1}^{M}|D_m| \tag{1.4}$$

记划分子集 D_m 中，属于类 C_k 的样本的合集为 D_{mk}，即：

$$D_{mk}=D_m\cap C_k \tag{1.5}$$

$$D_m=\bigcup_{k=1}^{K}D_{mk} \tag{1.6}$$

$$|D_m|=\sum_{k=1}^{K}|D_{mk}| \tag{1.7}$$

1.1.1.1 信息熵

在信息论与概率统计中，熵（*entropy*）是表示随机变量不确定的度量。

设 X 是一个取有限个值的离散随机变量，其概率分布为：

$$P(X=x_i)=p_i,\quad i=1,2,...,n \tag{1.8}$$

则随机变量 X 的熵定义为：

$$H(X)=H(p)=-\sum_{i=1}^{n}p_i\log p_i \tag{1.9}$$

由定义可知，熵只依赖于 X 的分布，而与 X 的取值无关，所以也可将 X 的熵记作 $H(p)$。在式 1.9 中，若 $p_i=0$，则定义 $0\log 0=0$。通常，式 1.9 中的对数以 2 为底或以 e 为底（自然对数），这时熵的单位分别称作比特（bit）或纳特（nat）。

熵越大，随机变量的不确定性就越大。从定义可验证：

$$0 \leq H(p) \leq \log n \tag{1.10}$$

设有随机变量 (X, Y) ，其联合概率分布 $P(X = x_i, Y = y_j)$ 为：

$$P(X = x_i, Y = y_j) = p_{ij}, \quad i = 1, 2, ..., n; \quad j = 1, 2, ..., m \tag{1.11}$$

随机变量 X 的边际分布 $P(X)$ 为：

$$P(X = x_i) = p_i = p_{i\cdot} = \sum_{j=1}^{m} p_{ij}, \quad i = 1, 2, ..., n \tag{1.12}$$

条件熵（*conditional entropy*） $H(Y|X)$ 表示在已知随机变量 X 的条件下随机变量 Y 的不确定性，定义为 X 给定条件下 Y 的条件概率分布的熵对 X 的数学期望：

$$H(Y|X) = -\sum_{i=1}^{n} p_i H(Y|X = x_i) \tag{1.13}$$

当 熵和 条件熵中的概率由数据估计（特别是极大似然估计）得到时，所对应的熵与条件熵分别称为经验熵（*empirical entropy*）和经验条件熵（*empirical conditional entropy*）。此时，如果由 0 概率，令 $0 \log 0 = 0$ 。

1.1.1.2 信息增益

信息增益（*information gain*）表示得知特征 X 的信息而使得类 Y 得信息的不确定性减少的程度。对于数据集 D ，特征 A 的信息增益 $g(D, A)$ ，定义为集合 D 的经验熵 $H(D)$ 与特征 A 给定条件下 D 的经验条件熵 $H(D|A)$ 之差，即：

$$g(D, A) = H(D) - H(D|A) \tag{1.14}$$

一般地，熵 $H(Y)$ 与条件熵 $H(Y|X)$ 之差为 互信息（*mutual information*）。

计算特征 A 对数据集 D 的信息增益 $g(D, A)$ 包括以下三个步骤：

步骤 1：计算数据集 D 的熵 $H(D)$ ：

$$H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} \log_2 \frac{|C_k|}{|D|} \tag{1.15}$$

步骤 2：计算特征 A 对数据集 D 的经验条件熵 $H(D|A)$ ：

$$\begin{aligned} H(D|A) &= \sum_{m=1}^{M} \frac{|D_m|}{|D|} H(D_m) \\ &= -\sum_{m=1}^{M} \frac{|D_m|}{|D|} \sum_{k=1}^{K} \frac{|D_{mk}|}{|D_m|} \log_2 \frac{|D_{mk}|}{|D_m|} \end{aligned} \tag{1.16}$$

步骤 3：计算信息增益 $g(D, A)$ ：

$$g(D, A) = H(D) - H(D|A) \tag{1.17}$$

1.1.1.3 信息增益比

信息增益比（*information gain ratio*），也称为信息增益率：特征 A 对数据集 D 的信息增益比 $g_R(D,A)$ 定义为其信息增益 $g(D,A)$ 与数据集 D 关于特征 A 的值的熵 $H_A(D)$ 之比，即：

$$g_R(D,A)=\frac{g(D,A)}{H_A(D)} \tag{1.18}$$

上式中，$H_A(D)$ 表示数据集 D 关于特征 A 的值的熵，计算公式如下：

$$H_A(D)=-\sum_{m=1}^{M}\frac{|D_m|}{|D|}\log_2\frac{|D_m|}{|D|} \tag{1.19}$$

1.1.1.4 基尼指数

基尼指数（*Gini index*），也被称为基尼不纯度（*Gini impurity*）。用于衡量数据集 D 的不纯度（*impurity*）。直观来说，基尼指数反映了从数据集 D 中随机抽取两个样本，其类别标记不一致的概率。基尼指数越小，表明数据集 D 的纯度越高（不纯度越低），样本的不确定性也就越小，这与熵相似。计算公式如下：

$$\begin{aligned}\mathrm{Gini}(p)&=\sum_{k=1}^{K}p_k(1-p_k)\\&=1-\sum_{k=1}^{K}p_k^2\end{aligned} \tag{1.20}$$

对于数据集 D，其基尼指数为：

$$\mathrm{Gini}(D)=1-\sum_{k=1}^{K}\left(\frac{|C_k|}{|D|}\right)^2 \tag{1.21}$$

在特征 A 的条件下，集合 D 的基尼指数 $\mathrm{Gini}(D,A)$ 定义为：

$$\mathrm{Gini}(D,A)=\sum_{m=1}^{M}\frac{|D_m|}{|D|}\mathrm{Gini}(D_m) \tag{1.22}$$

基尼指数一般用于 **CART 算法**完成分类任务，并且采用二分法分裂。因此，数据集根据特征 A 是否取某一值 a 被划分为 D_1 和 D_2 两个部分，即：

$$D_1=\{(\boldsymbol{x},y)\in D|A(\boldsymbol{x})=a\} \tag{1.23a}$$

$$D_2=D-D_1 \tag{1.23b}$$

此时，在特征 A 的条件下，集合 D 的基尼指数 $\mathrm{Gini}(D,A)$ 定义为：

$$\mathrm{Gini}(D,A)=\frac{|D_1|}{|D|}\mathrm{Gini}(D_1)+\frac{|D_2|}{|D|}\mathrm{Gini}(D_2) \tag{1.24}$$

1.1.2 连续特征分裂指标

由于决策树一般要求特征变量为连续变量，因此，在数据预处理阶段，需要使用连续数离散化技术对连续变量进行预处理。

在决策树模型中，**C4.5 算法**使用二分法（*bi-partition*）对连续数据离散化处理。

对于数据集 D 和某一特征的连续特征 A，假设特征 A 在数据集 D 上有 N 个值（即有 $N=|D|$ 个样本）。首先，将这些值从小到大进行排序，得到 $\{a_1,a_2,\cdots,a_N\}$。对于某一划分点 $t(a_1\leq t<a_N)$，可以将数据集 D 分成两个子集：

$$D_1=\{(\boldsymbol{x},y)\in D|A(\boldsymbol{x})\leq t\} \tag{1.25a}$$

$$D_2=\{(\boldsymbol{x},y)\in D|A(\boldsymbol{x})>t\} \tag{1.25b}$$

其中，对于相邻的特征取值 a_n 和 a_{n+1}，t 在区间 $[a_n,a_{n+1})$ 中取任意值所产生的划分结果相同。因此，在实际的操作中，切分点 t 往往取区间下界 a_n 或区间中点 $(a_n,a_{n+1})/2$。

进而，连续特征 A 被转换成二值化的离散特征，从而根据离散特征计算分裂指标。

1.1.3 回归问题

CART 算法可以用于实现回归任务。在回归任务中，每个叶子结点的值 c_m 为落入该结点样本的平均值，即：

$$c_m=\frac{1}{|R_m|}\sum_{(\boldsymbol{x}_i,y_i)\in R_m}y_i \tag{1.26}$$

上式中，R_m 为落入第 m 个叶子结点的样本合集。

1.1.3.1 均方误差（MSE）

用于回归任务的决策树，对于每个集合 D_m ，其 MSE 度量指标 $H(D_m)$ 的计算公式为：

$$H(D_m)=\frac{1}{|D_m|}\sum_{(\boldsymbol{x}_i,y_i)\in D_m}(y_i-c_m)^2 \tag{1.27}$$

上式中，c_m 为数据子集的 D_m 的均值：

$$c_m=\frac{1}{|D_m|}\sum_{(\boldsymbol{x}_i,y_i)\in D_m}y_i \tag{1.28}$$

可以看出，MSE 的度量指标 $H(D_m)$ 实际上为集合 D_m 的方差，即：

$$H(D_m)=\operatorname*{VAR}_{(\boldsymbol{x}_i,y_i)\in D_m}y_i \tag{1.29}$$

1.1.3.2 改进的均方误差（Friedman MSE）

1.1.3.3 平方绝对误差（MAE）

$$H(D_m)=\frac{1}{|D_m|}\sum_{(\boldsymbol{x}_i,y_i)\in D_m}|y_i-c_m| \tag{1.30}$$

1.1.3.4 Half Poisson Deviance

$$H(D_m)=\frac{1}{|D_m|}\sum_{(\boldsymbol{x}_i,y_i)\in D_m}\left(y_i\log\frac{y_i}{c_m}-y_i+c_m\right) \tag{1.31}$$

$$\mathrm{MSE}(D,A,t)=\frac{1}{|D_1|}\sum_{(\boldsymbol{x}_i,y_i)\in D_1}(y_i-c_1)^2+\frac{1}{|D_2|}\sum_{(\boldsymbol{x}_j,y_j)\in D_2}(y_j-c_2)^2 \tag{1.32}$$

在上式中，c_1 和 c_2 分别为数据子集 D_1 和 D_2 的均值：

$$c_1 = \frac{1}{|D_1|} \sum_{(\boldsymbol{x}_i, y_i) \in D_1} y_i \tag{1.33a}$$

$$c_2 = \frac{1}{|D_2|} \sum_{(\boldsymbol{x}_i, y_j) \in D_2} y_j \tag{1.33b}$$

$$\mathrm{MAE}(D, A, t) = \frac{1}{|D_1|} \sum_{(\boldsymbol{x}_i, y_i) \in D_1} |y_i - c_1| + \frac{1}{|D_2|} \sum_{(\boldsymbol{x}_j, y_j) \in D_2} |y_j - c_2| \tag{1.34}$$

在上式中，c_1 和 c_2 分别为数据子集 D_1 和 D_2 的中位数：

$$c_1 = \frac{1}{|D_1|} \mathrm{median}\{y_i | (\boldsymbol{x}_i, y_i) \in D_1\} \tag{1.35a}$$

$$c_2 = \frac{1}{|D_2|} \mathrm{median}\{y_j | (\boldsymbol{x}_j, y_j) \in D_2\} \tag{1.35b}$$

1.1.4 代码实现

1.1.4.1 信息熵

1.2 ID3

1.2.1 数学原理

ID3 算法是以 信息增益为准则来划分属性。

ID3 算法流程：

输入：数据集 $D = \{\boldsymbol{x_n}, y_n\}_{n=1}^{N}$ ，$x_n \in \mathbb{R}^P$ ，$y_n \in \mathbb{R}$ ；特征集 $\mathbb{A} = \{A_1, A_2, \cdots, A_p, \cdots, A_P\}$ ；划分阈值 ε。

输出：决策树 $\mathcal{T}$

步骤 1：若数据集 D 中所有样本属于同一类 C_k，并将类 C_k 作为该结点的类标记，返回 $\mathcal{T}$。

步骤 2：选择划分属性 A^* ：

(1) 特征集 $\mathbb{A} = \varnothing$，则 $\mathcal{T}$ 为单节点树，并将类 D 中实例数最大的类 C_k 作为该结点的类标记，返回 $\mathcal{T}$。

(2) 否则，计算各特征 A_p 对 D 的 信息增益 $g(D, A_p)$，选择 信息增益最大的特征 A^*，即：

$$A^* = \arg\max_{A_p \in \mathbb{A}} g(D, A_p) \tag{1.36}$$

步骤 3：生成叶子结点：

(1) 若 A^* 的 信息增益小于阈值 ε，则将数据集 D 中样本数最大的类 C_k 作为该结点的类标记，返回 $\mathcal{T}$。

(2) 否则，对 A^* 的每一取值 a_i，根据 $A^* = a_i$ 将数据集 D 分割为若干非空子集 D_i，将每一子集 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 $\mathcal{T}$，返回 $\mathcal{T}$。

步骤 4：对第 i 个子结点，以 D_i 为数据集，以 $\mathbb{A} - \{A^*\}$ 为特征集，递归地调用 **步骤 1** ~ **步骤 3**，得到子树 $\mathcal{T}_i$，返回 $\mathcal{T}_i$。

1.2.2 代码实现

1.2.3 实例结果

1.3 C4.5

1.3.1 数学原理

C4.5 算法与 **ID3 算法**过程相似，唯一的区别在于 **C4.5 算法**是以**信息增益比**为准则来划分属性。**信息增益**准则对可取值数目较多得属性有所偏好，使用**信息增益比**可以较少这种偏好带来得不利影响。

1.3.2 代码实现

1.3.3 实例结果

1.4 分类回归树 CART

分类与回归树（classification and regression tree, CART）即可用于 *分类*任务，又可用于 *回归*任务。**CART** 假设决策树是二叉树，内部结点特征的取值为“是”和“否”，左分支是取值为“是”的分支，右分支是取值为“否”的分支。因此，相比于上述两种的决策树模型，对于 **CART 算法**，在叶子结点每次分裂（树成长）的过程中，不仅需要确定最优划分特征，还需要确定最优划分特征的划分取值，对于*分类*任务和*回归*任务均是如此。

1.4.1 分类问题

1.4.2 回归问题

CART 算法也可以实现回归任务，用于实现回归任务的决策树，也被称为*回归树*（*regression tree*）。在 **CART 算法**中，树模型被要求为是*二叉树*结构，因此分裂指标的为左右叶子结点的指标之和。

例如，若采用 MSE 分裂指标，则分裂指标的计算方法如式 1.37 所示。这样的*回归树*通常被称为*最小二乘回归树*（*least squares regression tree*）。

$$H(D,A,t)=\frac{1}{|D_1|}\sum_{(\boldsymbol{x}_i,y_i)\in D_1}(y_i-c_1)^2+\frac{1}{|D_2|}\sum_{(\boldsymbol{x}_j,y_j)\in D_2}(y_j-c_2)^2 \tag{1.37}$$

CART 回归树流程：

输入：数据集 $D=\{\boldsymbol{x_n},y_n\}_{n=1}^N$，$x_n\in\mathbb{R}^P$，$y_n\in\mathbb{R}$；特征集 $\mathbb{A}=\{A_1,A_2,\cdots,A_p,\cdots,A_P\}$；划分阈值 ε。

输出：决策树 $\mathcal{T}$。

在数据集所在的输入空间种，递归地将每个区域划分为两个子区域并决定每个子区域地输出值，构建二叉决策树：

步骤 1：寻找最优划分特征 A^* 及其划分值 a^*。

(1) 确定每个特征 $A_p\in\mathbb{A}$ 的最优划分值 a^*。对于每个特征 A_p，遍历其所有的取值，选择使得划分指标 $H(D,A,a)$ 或 $H(D,A,t)$ 最小的特征值 a^* 或 t^*。注意，离散特征和连续特征的特征值有所差异。

$$a^*=\underset{a\in\{a_1,a_2,\ldots\}}{\arg\min}\ H(D,A_p,a) \tag{1.38}$$

(2) 确定最优划分特征 A^*。根据每个特征 A_p 所计算得到的最优划分指标 $H(D, A_p)$，选择最优的划分特征，即：

$$A^* = \arg\max_{A_p \in \mathbb{A}} H(D, A_p) \tag{1.39}$$

步骤 2：使用选定的划分特征 A^* 和划分值 a^* 划分区域 R_m 并决定相应的输出值 c_m：

$$R_1(A^*, a^*) = \{(\boldsymbol{x}, y) \in D | A(\boldsymbol{x}) = a^*\}, \quad c_1 = \frac{1}{|R_1|} \sum_{(\boldsymbol{x}_i, y_i) \in R_1} y_i \tag{1.40a}$$

$$R_2(A^*, a^*) = \{(\boldsymbol{x}, y) \in D | A(\boldsymbol{x}) \neq a^*\}, \quad c_2 = \frac{1}{|R_2|} \sum_{(\boldsymbol{x}_i, y_i) \in R_2} y_i \tag{1.40b}$$

步骤 3：对于叶子结点 R_1 和 R_2 递归地调用**步骤 1** ~ **步骤 2**，直到满足终止条件。

步骤 4：最终，数据集被划分为 M 个区域 $R_m (m = 1, 2, \cdots, M)$，每个区域（叶子结点）的输出为落入该区域样本的平均值。因此，可以得到决策树最终的输出为：

$$f(\boldsymbol{x}) = \sum_{m=1}^{M} c_m \mathrm{I}\left((\boldsymbol{x}, y) \in R_m\right) \tag{1.41}$$

上式中，$\mathrm{I}(x)$ 为指数函数，当括号内 x 为真，函数返回 1，否则返回 0。

1.4.3 实例结果

1.5 泛化技术

1.5.1 预剪枝

1.5.2 后剪枝

1.6 连续值与缺失值处理