5_决策树

决策树是基于树结构来对实例进行决策的一种基本的分类和回归的机器学习方法。决策树由结点和有向边组成，结点分为内部结点（表示一个特征的划分）和叶子结点（表示一个类别或输出）。

决策树学习，训练数据集

$$D=\{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),\cdots,(\mathbf{x}_i,y_i),\cdots,(\mathbf{x}_N,y_N)\}$$

其中，$\mathbf{x}_i$为第i个特征向量（实例），$\mathbf{x}_i=\left(x_i^{(1)},x_i^{(2)},\ldots,x_i^{(j)},\ldots,x_i^{(n)}\right)^T\in\mathcal{X}\subseteq\mathbb{R}^n$；$y_i$为$\mathbf{x}_i$的类别标记，$y_i\in\{1,2,\cdots,K\}$。学习的目标数是根据给定的训练数据集构建一个决策树模型，使得可对实例进行正确的分类或回归。

决策树学习包括3个步骤：特征选择、决策树生成、决策树修剪。

5_1_特征选择

熵表示随机变量不确定性的度量。
设X是一个取有限个值的离散随机变量，其概率分布为

$$P(X=x_i)=p_i,\quad i=1,2,\cdots,n$$

则随机变量X的熵

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

其中，若$p_i=0$，则定义$0\log 0=0$

若

$$p_i=\frac{1}{n}$$

则

$$\begin{aligned}H(p)&=-\sum_{i=1}^{n}p_i\log p_i\\&=-\sum_{i=1}^{n}\frac{1}{n}\log\frac{1}{n}\\&=\log n\end{aligned}$$

由定义，得

$$0\le H(p)\le\log n$$

设有随机变量(X,Y)，其联合分布

$$P\left(X=x_i,Y=y_j\right)=p_{ij},\quad i=1,2,\cdots,n;\quad j=1,2,\cdots,m$$

随机变量X给定的条件下随机变量Y的条件熵

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

即，X给定条件下Y的条件概率分布的熵对X的数学期望。其中，$p_i=P(X=x_i),i=1,2,\cdots,n$。
条件熵$H(Y|X)$表示在已知随机变量X的条件下随机变量Y的不确定性。

特征A对训练集D的信息增益

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

即，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。
其中，当熵和条件熵由数据估计（极大似然估计）得到时，对应的熵和条件熵分别称为经验熵和经验条件熵。

设训练数据集为D，$|D|$表示其样本容量，即样本个数。
设有K个类$C_k,k=1,2,\cdots,K$，$|C_k|$为属于类C_k的样本的个数，$\sum_{k=1}^{K}|C_k|=|D|$。
设特征A有n个不同的特征取值$\{a_1,a_2,\cdots,a_n\}$，根据特征A的取值将D划分为n个子集$D_1,D_2,\cdots,D_n$，$|D_i|$为D_i的样本数，$\sum_{i=1}^{n}|D_i|=|D|$。
记子集D_i中属于类C_k的样本的集合为D_{ik}，即$D_{ik}=D_i\cap C_k$，$|D_{ik}|$为D_{ik}的样本个数。

信息增益算法：
输入：训练数据集D和特征A
输出：特征A对训练数据集D的信息增益$g(D,A)$

1. 计算数据集D的经验熵$H(D)$

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

2. 计算特征A对数据集D的经验条件熵$H(D|A)$

$$\begin{aligned}H(D|A)&=\sum_{i=1}^{n}\frac{|D_i|}{|D|}H(D_i)\\&=\sum_{i=1}^{n}\frac{|D_i|}{|D|}\sum_{k=1}^{K}\frac{|D_{ik}|}{|D_i|}\log_2\frac{|D_{ik}|}{|D_i|}\end{aligned}$$

3. 计算信息增益

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

特征A对训练集D的信息增益比

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

即，信息增益$g(D,A)$与训练数据集D关于特征A的经验熵$H_A(D)$之比。
其中，

$$H_A(D)=-\sum_{i=1}^{n}\frac{|D_i|}{|D|}\log_2\frac{|D_i|}{|D|}$$

5_2_决策树生成

ID3算法：
输入：训练数据集D，特征集合A，阈值ε
输出：决策树T

1. 若D中所有实例属于同一类C_k，则T为单结点树，并将类C_k作为该结点的类标记，返回T；
2. 若$A=\emptyset$，则T为单结点树，并将D中实例数最大的类C_k作为该结点的类标记，返回T；
3. 否则，计算A中各特征对D的信息增益，选择信息增益最大的特征A_g

$$A_g = \arg\max_A g(D,A)$$

4. 如果A_g的信息增益小于阈值ε，则置T为单结点树，并将D中实例数量最大的类C_k作为该结点的类标记，返回T；
5. 否则，对A_g的每一个可能值a_i，依$A_g=a_i$将D分割为若干非空子集D_i，将D_i中实例数对大的类作为标记，构建子结点，由结点及其子结点构成树T，返回T；
6. 对第i个子结点，以D_i为训练集，以$A-\{A_g\}$为特征集，递归地调用步1.～步5.，得到子树T_i，返回T_i。

C4.5算法：
输入：训练数据集D，特征集合A，阈值ε
输出：决策树T

1. 若D中所有实例属于同一类C_k，则T为单结点树，并将类C_k作为该结点的类标记，返回T；
2. 若$A=\emptyset$，则T为单结点树，并将D中实例数最大的类C_k作为该结点的类标记，返回T；
3. 否则，计算A中各特征对D的信息增益，选择信息增益比最大的特征A_g

$$A_g = \arg\max_A g_R(D,A)$$

4. 如果A_g的信息增益小于阈值ε，则置T为单结点树，并将D中实例数量最大的类C_k作为该结点的类标记，返回T；
5. 否则，对A_g的每一个可能值a_i，依$A_g=a_i$将D分割为若干非空子集D_i，将D_i中实例数对大的类作为标记，构建子结点，由结点及其子结点构成树T，返回T；
6. 对第i个子结点，以D_i为训练集，以$A-\{A_g\}$为特征集，递归地调用步1.～步5.，得到子树T_i，返回T_i。

5_3_决策树剪枝

决策树的剪枝通过极小化决策树整体的损失函数或代价函数来实现。
设树T的叶结点个数为$|T|$，t是树T的叶结点，该叶结点有N_t个样本点，其中k类的样本点有N_{tk}个，$k=1,2,\cdots,K$，$H_t(T)$为叶结点t上的经验熵，
则决策树的损失函数

$$C_\alpha(T)=\sum_{t=1}^{|T|}N_tH_t(T)+\alpha|T|$$

其中，$\alpha\geq 0$为参数，经验熵

$$H_t(T)=-\sum_k\frac{N_{tk}}{N_t}\log\frac{N_{tk}}{N_t}$$

损失函数中，记

$$C(T)=\sum_{t=1}^{|T|}N_tH_t(T)=-\sum_{t=1}^{|T|}\sum_{k=1}^{K}N_{tk}\log\frac{N_{tk}}{N_t}$$

则

$$C_\alpha(T)=C(T)+\alpha|T|$$

其中，$C(T)$表示模型对训练数据的预测误差，即模型与训练数据的拟合程度，$|T|$表示模型复杂度，参数$\alpha\geq 0$控制两者之间的影响。

树的剪枝算法：
输入：决策树T，参数α
输出：修剪后的子树T_α

1. 计算每个结点的经验熵
2. 递归地从树的叶结点向上回缩
   设一组叶结点回缩到其父结点之前与之后的整体树分别为T_B与T_A，其对应的损失函数值分别是$C_\alpha(T_B)$与$C_\alpha(T_A)$，如果
   $$C_\alpha(T_A)\leq C_\alpha(T_B)$$
   则进行剪枝，即将父结点变为新的叶结点。
3. 返回2.，直到不能继续为止，得到损失函数最小的子树T_α

5_4_分类与回归树CART

5_4_1_回归树的生成

假设X与Y分别为输入和输出变量，并且Y是连续变量，给定训练数据集

$$D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)\}$$

可选择第j个变量$x^{(j)}$及其取值s作为切分变量和切分点，并定义两个区域

$$R_1(j,s) = \left\{x|x^{(j)} \le s\right\}, \quad R_2(j,s) = \left\{x|x^{(j)} > s\right\}$$

最优切分变量x_j及最优切分点s

$$j, s = \arg\min_{j,s}\left[\min_{c_1}\sum_{x_i \in R_1(j,s)}(y_i - c_1)^2 + \min_{c_2}\sum_{x_i \in R_2(j,s)}(y_i - c_2)^2\right]$$

其中，c_m是区域R_m上的回归决策树输出，是区域R_m上所有输入实例x_i对应的输出y_i的均值

$$c_m = ave(y_i|x_i \in R_m), \quad m = 1, 2$$

对每个区域R_1和R_2重复上述过程，将输入空间划分为M个区域$R_1, R_2, \cdots, R_M$，在每个区域上的输出为$c_m, m = 1, 2, \cdots, M$，最小二乘回归树

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

最小二乘回归树生成算法：
输入：训练数据集D
输出：回归树$f(x)$

1. 选择最优切分变量$x^{(j)}$与切分点s

$$j, s = \arg\min_{j,s}\left[\min_{c_1}\sum_{x_i \in R_1(j,s)}(y_i - c_1)^2 + \min_{c_2}\sum_{x_i \in R_2(j,s)}(y_i - c_2)^2\right]$$

2. 用最优切分变量$x^{(j)}$与切分点s划分区域并决定相应的输出值

$$R_1(j,s) = \left\{x|x^{(j)} \le s\right\}, \quad R_2(j,s) = \left\{x|x^{(j)} > s\right\}$$
$$c_m = \frac{1}{N}\sum_{x_i \in R_m(j,s)} y_i, \quad m = 1, 2$$

3. 继续对两个子区域调用步骤1.和2.，直到满足停止条件
4. 将输入空间划分为M个区域$R_1, R_2, \cdots, R_M$，生成决策树

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

5_4_2_分类树的生成

分类问题中，假设有K个类，样本点属于第k类的概率为p_k，则概率分布的基尼指数

$$Gini(p) = \sum_{k=1}^{K} p_k(1 - p_k) = 1 - \sum_{k=1}^{K}$$

对于二分类问题，若样本点属于第1类的概率为p，则概率分布的基尼指数

$$Gini\left(p\right)=\sum_{k=1}^{2}p_k\left(1-p_k\right)=2p\left(1-p\right)$$

对于给定样本集和D，其基尼指数

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

其中，C_k是D中属于第k类的样本自己，K是类别个数。

如果样本集合D根据特征A是否取某一可能值a被分割成D_1和D_2两个部分，即

$$D_1=\left\{\left(x,y\right)|A\left(x\right)=a\right\},\quad D_2=D-D_1$$

则在特征A的条件下，集合D的基尼指数

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

基尼指数$Gini\left(D\right)$表示集合D的不确定性，基尼指数$Gini\left(D,A\right)$表示经$A=a$分割后集合D的不确定性。基尼指数值越大，样本集合的不确定性也越大。

CART生成算法：
输入：训练数据集D，特征A，阈值ε
输出：CART决策树T

1. 设结点的训练数据集为D，对每一个特征A，对其可能取的每个值a，根据样本点对$A=a$的测试为“是”或“否”将D分割成D_1和D_2两部分，并计算$Gini\left(D,A\right)$
2. 在所有可能的特征A以及其所有可能的切分点a中，选择基尼指数最小的特征及其对应的切分点作为最优特征与最优切分点。依此从现结点生成两个子结点，将训练数据集依特征分配到两个子结点中去。
3. 对两个子结点递归地调用1.和2.，直至满足停止条件
4. 生成CART决策树T

5_4_3_CART树剪枝

对整体树T_0任意内部结点t，以t为单结点树的损失函数

$$C_\alpha(t) = C(t) + \alpha$$

以t为根结点的子树T_t的损失函数

$$C_\alpha(T_t) = C(T_t) + \alpha|T_t|$$

当$\alpha = 0$及α充分小时，有不等式

$$C_\alpha(T_t) < C_\alpha(t)$$

当α增大时，在某一α有

$$C_\alpha(T_t) = C_\alpha(t)$$
$$C(T_t) + \alpha|T_t| = C(t) + \alpha$$
$$\alpha = \frac{C(t) - C(T_t)}{|T_t| - 1}$$

即T_t与t有相同的损失函数值，而t的结点少，因此对T_t进行剪枝。

CART剪枝算法：
输入：CART决策树T_0
输出：最优决策树T_α

1. 设$k = 0, T = T_0$
2. 设$\alpha = +\infty$
3. 自下而上地对各内部结点t计算$C(T_t)$，$|T_t|$，以及

$$g(t) = \frac{C(t) - C(T_t)}{|T_t| - 1}$$
$$\alpha = \min(\alpha, g(t))$$

   其中，T_t表示以t为根结点的子树，$C(T_t)$是对训练数据的预测误差，$|T_t|$是T_t的叶结点个数。
4. 自下而上地访问内部结点t，如果有$g(t) = \alpha$，则进行剪枝，并对叶结点t以多数表决法决定其类别，得到树T
5. 设$k = k + 1, \alpha_k = \alpha, T_k = T$
6. 如果T不是由根结点单独构成的树，则回到步骤4.
7. 采用交叉验证法在子树序列$T_0, T_1, \cdots, T_n$中选取最优子树T_α