决策树

1. 一种重要的分类与回归的方法，可以看成一系列 if-else 集合

   1) 模型定义

   组成 ① 节点 { 内部节点：特征或属性；叶节点：表示一个类别 }

   ② 有向边

   2) if-then 集合

   从根节点到叶节点的每一条路径构成一个规则

   内部节点 → 条件

   叶节点 → 结论

   性质：互斥且完备，每个实例可以被一条规则覆盖，且只能被一个规则覆盖

   3) 条件概率分布

   ① 将特征空间分为互不相交的单元

   ② 在~~每~~个单元定义一个类的概率分布

   ③ 将实例强行分类到分布概率较大的类

   4) 决策树学习

   $$D=\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\} \quad y_i\in\{1,\cdots,k\}$$

   ① 构建根节点：将所有训练数据放在根节点

   ② 选择一个最优特征进行划分

   ③ { if 按照此划分可以将训练集基本正确分类，构建叶节点
   else 递归选择特征进行划分，直到停止条件 }

   ④ 决策树剪枝：提高泛化能力，防止过拟合

2、特征选择

2.1 特征选择问题

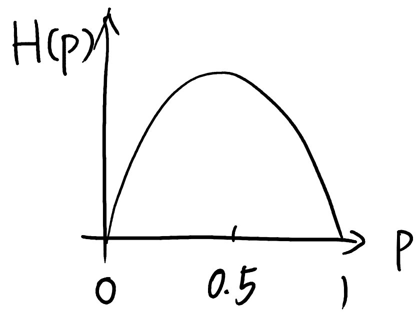
目的：选择对训练集有分类能力的特征

按照此特征分类后与随机分类差异不大，则认为没有分类能力，

一般选择信息增益大的分类

2.2 信息增益

熵(entropy)是随机变量不确定性的度量



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

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

以2为底(bit 比特)

以e为底(nat 纳特)

条件熵(Conditional Entropy)：$H(Y|X) = \sum_i p_i H(Y|X=x_i)$

信息增益：得知X的特征后，Y不确定性的减少

$g(D,A) = H(D) - H(D|A)$ 也称为互信息(mutual information)

· 训练集D：

K个类别：$\{C_1, C_2, \cdots, C_K\}$ $\quad \sum_k |C_k| = |D|$

某个特征：$A = \{a_1, a_2, \cdots, a_n\}$，$\quad D_{ik} = D_i \cap C_k$

$$\begin{matrix} a_1 & a_2 & \cdots & a_n \\ \downarrow & \downarrow & & \downarrow \\ D_1 & D_2 & & D_n \end{matrix}$$

经验熵：$H(D) = -\sum_k \frac{|C_k|}{|D|} \cdot \log \frac{|C_k|}{|D|}$

经验条件熵：$H(D|A) = \sum_i \frac{|D_i|}{|D|} \left\{ -\sum_k \frac{|D_{ik}|}{|D_i|} \log \frac{|D_{ik}|}{|D_i|} \right\}$

2.3 信息增益比

信息增益倾向于取值大的特征

$$g_R(D,A)=\frac{g(D,A)}{H_A(D)}, \quad H_A(D)=-\sum_i \frac{|D_i|}{|D|}\log\frac{|D_i|}{|D|}$$

3. 决策树的生成

3.1 ID3 算法

输入：训练集 D，特征集 A，阈值 ε

1) 若 D 中所有实例属于同一类 C_k，则标注 C_k 反回 T

2) 若 $A=\phi$，则 T 为单节点树，将 D 中实例数最大的类 C_k 作为该节点类的标记。

3) 否则，对 A 中各个特征计算信息增益，选择信息增益最大的特征 A_g.

4) 如果 A_g 的信息增益小于域值 ε，则设置 T 为单节点树，将 D 中实例数最大的类作为标记，返回 T

5) 否则，对 A_g 的每个特征值 a_i，依 $A_g=a_i$ 将 D 分为若干个非空子集 D_i，将 D_i 中实例最大的类作为标记，构建子节点，由节点和子节点构成树 T，返回 T。

6) 对第 i 个子节点，以 D_i 为训练集，以 $A\setminus\{A_g\}$ 为特征集，递归调用 1-5 步，得到子树 T_i，返回 T_i。

3.2 C45算法

使用信息增益比作为划分依据，ID3算法倾向于选择属性取值较多的特征，除以$H_A(D)$可以削弱这种作用

4. 决策树的剪枝 (pruning)

目标：预防过拟合，降低树的复杂度

树的叶节点个数 $|T|$

对于叶节点 t，设有 N_t 个样本，其中 k 类样本为 N_{tk}

损失函数

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

复杂度

$H_t(T)$ 叶节点 t 上的经验熵，越小代表拟合程度越好

$$H_t(T) = -\sum_{k} \frac{N_{tk}}{N_t} \log \frac{N_{tk}}{N_t},$$

α - tuning parameter，α 越大，树越简单

5. CART算法 (classification and regression tree)

CART是二叉树，由两步组成 $\begin{cases} \text{生成} \\ \text{剪枝} \end{cases}$

5.1 CART

回归树：平方误差最小化

分类树：基尼系数

(1) 回归树的形成

①给定划分输出预测值

假设特征空间被划分成 $R_1, R_2, \cdots, R_m$

$R_m \to C_m$, $f(x)=\sum_{m} C_m I(x\in R_n)$

$\min_{C_m} \sum_{x_i\in R_m}(y_i-C_m)^2$ $\therefore \hat{C}_m = ave(y_i|x_i\in R_m)$

②如何得到划分

先寻找切分变量，再寻找切分点

$R_1(j,s)=\{x|x^{(j)}\leq s\}$ $R_2(j,s)=\{x|x^{(j)}>s\}$ $j=1,2,\cdots,p$

$$\min_{j,s}\left\{\min_{C_1}\sum_{x\in R_1(j,s)}(y_i-C_1)^2+\min_{C_2}\sum_{x\in R_2(j,s)}(y_i-C_2)^2\right\}$$

$C_i = avg\{y|x\in R_i(s,j)\}$

(2) 分类树的生成

基尼系数：选择最优特征

定义：K个类 $\sim P_K$ $Gini(p)=\sum_{k=1}^{K} p_k(1-p_k)=1-\sum_{k=1}^{K}p_k^2$

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

若样本集合根据A是否取某个值分为 D_1 和 D_2

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

CART算法

(1) 训练集为 D，某个特征为 A，根据 A 的取值 $\begin{cases} A=a \rightarrow D_1 \\ A\neq a \rightarrow D_2 \end{cases}$

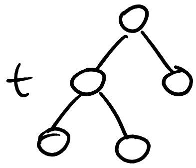
计算Gini index ①A可离散也可连续 ②二叉树

(2) 选择Gini指数最小的A和切分点

(3) 对两个子节点递归调用(1)~(2)

5.2 CART 剪枝

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



对于任意内部节点 t

$C_\alpha(t) = C(t) + \alpha$ → t 变成叶节点

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

if $C_\alpha(T_t) < C_\alpha(t)$ → 不剪枝

$C_\alpha(T_t) > C_\alpha(t)$ → 剪枝

使 $C_\alpha(t) = C_\alpha(T_t)$，$\alpha = \dfrac{C(t)-C(T_t)}{|T_t|-1}$

对 T_0 中每个内部节点 t，计算 $g(t) = \dfrac{C(t)-C(T_t)}{|T_t|-1}$

设最小的 $g(t)$ 为 α_1，T_1 为区间 $[\alpha_1, \alpha_2)$ 的最优子树，不断增大 α 的值，产生新的最优区间

$$|T_0| > |T_1| > |T_2|$$

在剪枝得到的子树中，通过交叉验证得到最优子树