Decision-Tree

决策树学习算法包括两个部分：建立决策树和对决策树进行剪枝，根据建立决策树的依据可以把学习算法分为 ID3 算法，C4.5 算法和 CART 算法，其中 CART 算法可以用来分类也可以用来回归。

ID3 算法

a. 用 H(T)表示训练集 T 的经验熵，K 是训练集的类别总数，则

$$H(T)=-\sum_{k}^{K}\frac{N_k}{|T|}\log\frac{N_k}{|T|}$$

其中 $N_k,|T|$ 分别表示训练集中类别是 k 的数量和训练集的总数，当训练集根据某一特征 A 进行划分时的条件熵表示为

$$H(T|A)=\sum_{i}^{|A|}\frac{D_i}{|T|}H(D_i)$$

其中 $|A|$ 表示为特征 A 可能值的数量，然后选择最优的特征作为训练集 T 的划分特征

$$p=\underset{A}{argmax}\{\boldsymbol{H(T)-H(T|A)}\}$$

b. 对决策树进行剪枝，使其更好的泛化

引入控制因子 α 改变它可以改变树的叶子的数量，假设树 T 的叶子节点数量为 T_f ，定义损失函数为

$$C_a(T)=\sum_{i}^{|T_f|}N_i H(N_i)+\alpha|T_f|$$

如果剪枝后的损失函数小于剪枝前的损失函数则进行剪枝，并且合并的节点由多数表决决定标签。

CART 算法

a. 回归树

假设有训练集 T $(x_1,y_1),(x_2,y_2),\ldots,(x_N,y_N)$ ，则划分点 p

$$p=\underset{x_i}{argmin}\left\{\sum_{x_j\in R(x_j<x_i)}(y_j-c_i^{(1)})^2+\sum_{x_j\in R(x_j\geq x_i)}(y_j-c_i^{(2)})^2\right\}$$

其中 $c_i^{(1)},c_i^{(2)}$ 表示为

$$c_i^{(1)}=\frac{1}{|R_1|}\sum_{x_j\in R(x_i<x_j)}y_j$$
$$c_i^{(2)}=\frac{1}{|R_2|}\sum_{x_j\in R(x_i\geq x_j)}y_j$$

b. 分类树

CART 算法分类树是使用基尼指数为特征选择的依据，假设有训练数据集 T 有 K 个类， p_k 表示第 k 类的样本概率，则

$$Gini(T)=\sum_{K} p_k(1-p_k)=1-\sum_{K} p_k^2$$

当特征 $A=a$ 把训练集分成 2 个子集 $D_1(A=a), D_2(A\neq a)$ 时，则基尼指数可以定义为

$$Gini(T,A=a)=\sum_{i}^{2}\frac{D_i}{|T|}Gini(D_i)$$

最优的特征划及特征的切分点

$$\{\boldsymbol{A},\boldsymbol{a}\}=\underset{A,a}{argmin}\{\boldsymbol{Gini}(\boldsymbol{T},\boldsymbol{A}=\boldsymbol{a})\}$$

c. CART 剪枝

引入因子 α 来 $C_\alpha(T_t)=Gini(T_t)+\alpha$ 控制决策树的叶子节点数量，树 T 的损失函数

$$C_a(T)=Gini(T)+\alpha|T_f|$$

当 T 被剪枝，则损失函数为

$$C_\alpha(T_t)=Gini(T_t)+\alpha$$

当 α 趋近零时， $C_\alpha(T)<C_\alpha(T_t)$,当 α 趋近无穷大时， $C_\alpha(T)>C_\alpha(T_t)$,则可能存在一个 α 使得 $C_\alpha(T)=C_\alpha(T_t)$,此时

$$\alpha=\frac{Gini(T_t)-Gini(T)}{|T_f|-1}$$

表明剪枝和没剪枝损失函数是一样的，因此进行剪枝