决策树 Decision Tree

1.决策树基本原理

决策树（Decision Tree）是一种分而治之的决策过程。一个困难的预测问题，通过树的分支节点，被划分成两个或多个较为简单的子集，从结构上划分为不同的子问题。将依规则分割数据集的过程不断递归下去（Recursive Partitioning）。随着树的深度不断增加，分支节点的子集越来越小，所需要提的问题数也逐渐简化。当分支节点的深度或者问题的简单程度满足一定的停止规则（Stopping Rule）时, 该分支节点会停止分裂，此为自上而下的停止阈值（Cutoff Threshold）法；有些决策树也使用自下而上的剪枝（Pruning）法。

2.决策树三要素

1. 特征选择：

   从训练数据的众多特征中选择一个特征作为当前节点的分裂标准。

2. 决策树生成：

   根据选择的特征评估标准，从上至下递归地生成子节点，直到数据集不可分则决策树停止生长。

3. 剪枝：

   缩小树结构规模，缓解过拟合。剪枝技术有预剪枝和后剪枝两种。

3. 决策树学习基本算法

机器学习算法 1 决策树学习基本算法

输入: 训练集$D=\{(\mathbf{x}_1,y_1),(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_m,y_m)\}$; 属性集$A=\{(a_1,a_2,\ldots,a_d)\}$

输出: 以$node$为根节点的一棵决策树

```
function TreeGenerate(D, A)
    生成节点node
    if D中的样本全属于同一类别C then
        将node标记为C类叶节点;return
    end if
    if A = ∅ OR D中样本再A上取值相同 then
        将node标记为叶节点，其类别标记为D中样本类最多的类;return
    end if
    从A中选择最优划分属性a_*
    for a_*的每一个值a_*^v do
        为node生成一个分支;令D_v表示D中在a_*上取值为a_*^v的样本子集;
        if D_v为空 then 将分支节点标记为叶节点，其类别标记为D中样本最多的类;return
        else 以TreeGenerate(D_v, A\{a_*})为分支节点
        end if
    end for
end function
```


4.决策树算法优缺点

优点	缺点
可用于小数据集	对连续性的字段比较难预测
时间复杂度较小，为用于训练决策树的数据点的对数	容易出现过拟合
可处理数字和数据的类别	当类别太多时，错误可能就会增加的比较快
能处理多输出问题	在处理特征关联性比较强的数据时表现得不是太好
对缺失值不敏感	对于各类别样本数量不一致的数据，在决策树当中，信息增益的结果偏向于那些具有更多数值的特征
可处理不相关特征数据	
效率高，决策树只需要一次构建，反复使用，每一次预测的最大计算次数不超过决策树的深度	

5.熵的概念

熵：度量随机变量的不确定性

定义：假设随机变量X的可能取值有$x_1, x_2, \ldots, x_n$，对于每一个可能的取值x_i，其概率为$P(X = x_i) = p_i, i = 1, 2..., n$。随机变量的熵为：
$H(X) = -\sum_{i=1}^{n} p_i log_2 p_i$ 对于样本集合，假设样本有k个类别，每个类别的概率为$\frac{|C_k|}{|D|}$，其中 $|C_k||D|$为类别为k的样本个数，$|D|$为样本总数。样本集合D的熵为：
$H(D) = -\sum_{k=1}^{k} \frac{|C_k|}{|D|} log_2 \frac{|C_k|}{|D|}$

6.信息增益

定义：

信息增益为以某特征划分数据集前后的熵的差值，用于衡量使用当前特征对于样本集合D划分结果的好坏。

假设划分前样本集合D的熵为$H(D)$。使用某个特征A划分数据集D，计算划分后的数据子集的熵为$H(D|A)$。
则信息增益为：

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

> *注：*在决策树构建的过程中我们总是希望集合往最快到达纯度更高的子集合方向发展，因此我们总是选择使得信息增益最大的特征来划分当前数据集D。

思想：

计算所有特征划分数据集D，得到多个特征划分数据集D的信息增益，从这些信息增益中选择最大的，因而当前结点的划分特征便是使信息增益最大的划分所使用的特征。

> **信息增益比**：
>
> 信息增益比 = 惩罚参数 × 信息增益
>
> 信息增益比本质：
>
> 在信息增益的基础之上乘上一个惩罚参数。特征个数较多时，惩罚参数较小；特征个数较少时，惩罚参数较大。
>
> 惩罚参数：
> 数据集D以特征A作为随机变量的熵的倒数。

7. 剪枝处理的作用及策略

剪枝处理可以解决决策树算法的过拟合问题。

可以采用剪枝处理来去掉一些分支来降低过拟合的风险。

枝的基本策略有预剪枝（pre-pruning）和后剪枝（post-pruning）。

预剪枝：

在决策树生成过程中，在每个节点划分前先估计其划分后的泛化性能， 如果不能提升，则停止划分，将当前节点标记为叶结点。

后剪枝：

生成决策树以后，再自下而上对非叶结点进行考察， 若将此节点标记为叶结点可以带来泛化性能提升，则修改之。