**[决策树模型 ID3/C4.5/CART算法比较](http://www.cnblogs.com/wxquare/p/5379970.html)**

<https://www.cnblogs.com/wxquare/p/5379970.html>

   决策树模型在监督学习中非常常见，可用于分类（二分类、多分类）和回归。虽然将多棵弱决策树的Bagging、Random Forest、Boosting等tree ensembel 模型更为常见，但是“完全生长”决策树因为其简单直观，具有很强的解释性，也有广泛的应用，而且决策树是tree ensemble 的基础，值得好好理解。一般而言一棵“完全生长”的决策树包含，**特征选择、决策树构建、剪枝**三个过程，这篇文章主要是简单梳理比较ID3、C4.5、CART算法。《统计学习方法》中有比较详细的介绍。

**一、决策树的优点和缺点**

    优点：

1. 决策树算法中学习简单的决策规则建立决策树模型的过程非常容易理解，
2. 决策树模型可以可视化，非常直观
3. 应用范围广，可用于分类和回归，而且非常容易做多类别的分类
4. 能够处理数值型和连续的样本特征

    缺点：

1. 很容易在训练数据中生成复杂的树结构，造成过拟合（overfitting）。剪枝可以缓解过拟合的负作用，常用方法是限制树的高度、叶子节点中的最少样本数量。
2. 学习一棵最优的决策树被认为是NP-Complete问题。实际中的决策树是基于启发式的贪心算法建立的，这种算法不能保证建立全局最优的决策树。Random Forest 引入随机能缓解这个问题

**二、ID3算法**

      ID3由Ross Quinlan在1986年提出。ID3决策树可以有多个分支，但是不能处理特征值为连续的情况。决策树是一种贪心算法，每次选取的分割数据的特征都是当前的最佳选择，并不关心是否达到最优。在ID3中，每次根据“最大信息熵增益”选取当前最佳的特征来分割数据，并按照该特征的所有取值来切分，也就是说如果一个特征有4种取值，数据将被切分4份，一旦按某特征切分后，该特征在之后的算法执行中，将不再起作用，所以有观点认为这种切分方式过于迅速。ID3算法十分简单，核心是根据“最大信息熵增益”原则选择划分当前数据集的最好特征，信息熵是信息论里面的概念，是信息的度量方式，不确定度越大或者说越混乱，熵就越大。在建立决策树的过程中，根据特征属性划分数据，使得原本“混乱”的数据的熵(混乱度)减少，按照不同特征划分数据熵减少的程度会不一样。在ID3中选择熵减少程度最大的特征来划分数据（贪心），也就是“最大信息熵增益”原则。下面是计算公式，建议看[链接](http://leijun00.github.io/2014/09/decision-tree/)计算信息上增益的实例。

**三、C4.5算法**

      C4.5是Ross Quinlan在1993年在ID3的基础上改进而提出的。.ID3采用的信息增益度量存在一个缺点，它一般会优先选择有较多属性值的Feature,因为属性值多的Feature会有相对较大的信息增益?(信息增益反映的给定一个条件以后不确定性减少的程度,必然是分得越细的数据集确定性更高,也就是条件熵越小,信息增益越大).为了避免这个不足C4.5中是用信息增益比率(gain ratio)来作为选择分支的准则。信息增益比率通过引入一个被称作分裂信息(Split information)的项来惩罚取值较多的Feature。除此之外，C4.5还弥补了ID3中不能处理特征属性值连续的问题。但是，对连续属性值需要扫描排序，会使C4.5性能下降，有兴趣可以参考[博客](http://leijun00.github.io/2014/09/decision-tree/)

**五、CART算法**

**参考：[CART使用GINI指数分类](http://blog.csdn.net/u011067360/article/details/24871801" \t "_blank)**

CART（Classification and Regression tree）分类回归树由L.Breiman,J.Friedman,R.Olshen和C.Stone于1984年提出。ID3中根据属性值分割数据，之后该特征不会再起作用，这种快速切割的方式会影响算法的准确率。CART是一棵二叉树，采用二元切分法，每次把数据切成两份，分别进入左子树、右子树。而且每个非叶子节点都有两个孩子，所以CART的叶子节点比非叶子多1。相比ID3和C4.5，CART应用要多一些，既可以用于分类也可以用于回归。CART分类时，使用基尼指数（Gini）来选择最好的数据分割的特征，gini描述的是纯度，与信息熵的含义相似。CART中每一次迭代都会降低GINI系数。下图显示信息熵增益的一半，Gini指数，分类误差率三种评价指标非常接近。回归时使用均方差作为loss function。基尼系数的计算与信息熵增益的方式非常类似，公式如下

**六、分类树 VS 回归树**

提到决策树算法，很多想到的就是上面提到的ID3、C4.5、CART分类决策树。其实决策树分为分类树和回归树，前者用于分类，如晴天/阴天/雨天、用户性别、邮件是否是垃圾邮件，后者用于预测实数值，如明天的温度、用户的年龄等。

         作为对比，先说分类树，我们知道ID3、C4.5分类树在每次分枝时，是穷举每一个特征属性的每一个阈值，找到使得按照feature<=阈值，和feature>阈值分成的两个分枝的熵最大的feature和阈值。按照该标准分枝得到两个新节点，用同样方法继续分枝直到所有人都被分入性别唯一的叶子节点，或达到预设的终止条件，若最终叶子节点中的性别不唯一，则以多数人的性别作为该叶子节点的性别。

         回归树总体流程也是类似，不过在每个节点（不一定是叶子节点）都会得一个预测值，以年龄为例，该预测值等于属于这个节点的所有人年龄的平均值。分枝时穷举每一个feature的每个阈值找最好的分割点，但衡量最好的标准不再是最大熵，而是最小化均方差--即（每个人的年龄-预测年龄）^2 的总和 / N，或者说是每个人的预测误差平方和 除以 N。这很好理解，被预测出错的人数越多，错的越离谱，均方差就越大，通过最小化均方差能够找到最靠谱的分枝依据。分枝直到每个叶子节点上人的年龄都唯一（这太难了）或者达到预设的终止条件（如叶子个数上限），若最终叶子节点上人的年龄不唯一，则以该节点上所有人的平均年龄做为该叶子节点的预测年龄。