**CART算法简介及R实现**

目录

[1.决策树概述 1](#_Toc396060535)

[1.1.决策树模型学习 1](#_Toc396060536)

[1.2.常用决策树算法 1](#_Toc396060537)

[2.CART算法介绍 2](#_Toc396060538)

[2.1.决策树生成 2](#_Toc396060539)

[2.2.决策树剪枝 3](#_Toc396060540)

[2.3.CART算法优缺点 4](#_Toc396060541)

[3.CART算法R实现 5](#_Toc396060542)

[3.1.生成树(见CART\_algorithm.r) 5](#_Toc396060543)

[3.2.决策树剪枝(见CART\_algorithm.r) 5](#_Toc396060544)

[3.3.模型结果说明 5](#_Toc396060545)

[3.4.运用模型进行预测(见CART\_algorithm.r) 5](#_Toc396060546)

# 1.决策树概述

决策树算法是一类应用比较多的算法，主要用于解决分类(因变量为离散型)和回归(因变量为连续性)问题，生成的模型成树形结构，由结点和有向边构成，结点又包括内部结点和叶结点，内部结点是特征，叶结点是一个类标签。

## 1.1.决策树模型学习

决策树可以看做一个if-then规则的集合，由决策树根结点到叶结点的每一条路径都构成一条规则，这些规则在大部分算法中是互斥且完备的(C50算法一个样本可能命中多个规则，这个时候需要投票决定样本的归类)。这种规则的形式在我看来也是决策树最大的优势吧，因为除了可以从算法统计的角度去评估模型的效果外，还可以把规则拿出来结合自己的业务理解，去看这个规则是否靠谱，关键是这个规则的解释性极强，可以让业务运营人员一看就懂后面的逻辑，这也是为啥工作中我常选决策树算法的原因。

很多资料上将决策树的学习过程分为特征选择、树的生成和树的剪枝，但我的理解是特征选择的过程其实也就是决策树的生成过程。不同的算法有不同的特征选择度量标准，如ID3使用的是信息增益(Information gain)，C4.5和C5.0使用的是信息增益比(Information gain ratio)来衡量；而CART和随机森林则常用基尼指数(Gini index)作为特征选择的标准。

从另外一个角度来看，决策树表示的就是在某些特征条件下的类条件概率分布，将特征空间划分为互不相交的区域，在每个区域上求每个类的条件概率，哪个类的概率最大就将该区域分为哪类，具体内容可参看李航博士的《统计学习方法》，里面有很详细的描述。这一点儿想明白了也就基本清楚了单棵决策树为什么会有不稳定和有时候只能得到次优解这两大缺点，后面CART中会详述。

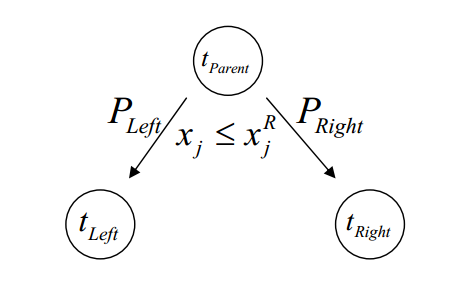
## 1.2.常用决策树算法

个人理解的决策树算法主要由两条主线构成，一是以Quinlan等人为主的基于信息熵为特征选择的标准的算法，这类里面先后有ID3算法、C4.5算法和C5.0算法，每一种算法都是在前一种算法的基础上进行改进和增加新功能，如C5.0里面已经有了boosting提升和特征预选的功能了；二是以Breiman等人为主的基于Gini index为特征选择标准的算法，也就是本文将重点介绍的CART算法，在此算法的基础上，Breiman又开发了随机森林算法。

上面的算法都具有解决分类和回归问题的能力，除了这些算法之外，还有专门解决回归问题的GBDT算法等，这里就不在一一列举。

# 2.CART算法介绍

分类与回归树(classification and regression tree,CART)是由Breiman在1984年提出，可用于解决分类和回归问题，CART是在给定输入变量X条件下输出随机变量Y的条件概率分布。CART较其他决策树算法的主要区别之一就是生成的是二叉树，如下图，而C5.0等可生成多叉树。内部结点产生判断逻辑后，左分支是回答为“是”的分支，右分支是回答为“否”的分支，如的判断逻辑为，那么就是满足该逻辑的，即，就是。



CART主要由两步组成：

1. 决策树生成：基于训练数据集生成决策树，而且树要越大越好，这个过程中涉及到特征选择、缺失值处理、先验概率调整等内容；
2. 决策树剪枝：对已生成的树进行剪枝并选择最优子树，剪枝的度量标准是损失函数。

## 2.1.决策树生成

决策树的生成就是一个构造二叉树的过程，回归树用平方误差最小化进行特征选择，分类树用基尼指数最小化进行特征选择(CART还可以用二分类点进行特征选择)，本文重点介绍分类树的构建，回归树类似。

假设一个训练数据集D包括K个类，样本属于第k类的概率为，则基尼指数定义为

如果样本集合D根据某个特征t是否取某个值，将D分层了和两部分，则在特征t的条件下，集合D的基尼指数定义为，基尼指数越大，样本集合的不确定性就越大。

CART分类树生成算法(详细参考李航博士《统计学习方法》)：

输入：训练数据集D，停止计算的条件；

输出：CART决策树

根据训练数据集，从根节点递归的对每个节点进行以下操作来构造二叉树：

1. 在训练数据集上，对每个特征都计算基尼指数
2. 在所有可能的特征以及所有可能的切分点中，选择基尼指数最小的特征及其对应的切分点作为最优切分点，将训练数据集分割到两个子节点中去
3. 对两个子结点，递归的调用(1)、(2)，直至满足停止条件
4. 生成CART决策树

在运行上述算法的过程中，有几个地方需要注意：

1. 停止计算条件，主要是通过复杂度参数和结点最小样本数两种方式来设置；
2. 连续性特征是否需要离散化？对于连续性属性，CART必须考虑特征值的所有可能的划分点。利用Gini指数最小原则，选择最优划分点。实践中我倾向于将其离散，一方面是为了降低决策树的不稳定性，另一方面出于性能的考虑，在大数据量的时候，对每一个连续特征的每个取值都要去计算最佳的切分点，需要较长的时间。至于离散化的标准，我更喜欢基于业务的理解去进行有实际意义的数据切分，效果并不比基于统计出来的要差，而且基于统计出来的离散有时候还不好解释。
3. 离散性特征是否需要处理成0、1型的哑变量？做过尝试，两种在效果上没有太大的差异。
4. 关于选择基尼指数最小的特征作为分裂点，熟悉C4.5或者C50的朋友应该比较熟悉这两种算法是通过信息增益比来进行特征选择的，它表示在得知特征t的情况下而使得类的不确定性减少的程度，所以这两种算法选择信息增益比最大的特征作为最优切分点。而基尼指数最小，本质上和信息增益比最大是一样的道理：对应的就是最小哈。

## 2.2.决策树剪枝

CART生成的树一般都比较复杂，模型在训练样本上的准确率较高，但是在新数据上的表现可能较差。CART剪枝是采用的代价-复杂度剪枝(Cost-Complexity Pruning, CCP)方法进行剪枝，从生成的决策树中剪去一些子树，使决策树变小，从而在新数据上有更好的泛化能力，其主要目的是为了权衡模型复杂度以及错误率，该方法主要有两步：1)从生成的决策树底端开始不断剪枝，一直到根结点，形成一个子树序列;2)评价这些子树序列，选择最优的树。

1. 剪枝，形成子树序列

代价-复杂度剪枝中的代价(cost)指的是样本错分率，复杂度(complexity)指的是树的叶结点数，Breiman定义t的代价复杂度为：，其中是决策树训练样本个数，是决策树错分样本数，是树的叶结点个数，而用于衡量代价和复杂度之间的关系。

对树来说，减掉它的子树s，用t中最优叶结点代替，就得到了新树，新树的分类错误样本数可能比原来的树多M个，可是新树的叶结点数却比原树少了个。复杂度降低了，但是代价上升了，如何来评估二者之间的关系？为了使剪枝前后的cost-complexity不变，令剪枝前后二者相等，就可以推到出的取值：

 综上，形成子树序列的过程为：首先计算完全决策树的每个非叶结点的值，然后从下往上减掉具有最小值的子树，直到剩下根结点，就得到一系列的子树序列，其中是原决策树，只包括一个跟结点。

1. 从子序列中选择最优子树

这里主要包括了两种方法：一种是基于训练数据集进行的交叉验证，一种基于独立验证数据集进行的标准误差筛选。

使用交叉验证方法，需要先将训练数据集D划分成V个子集，然后从中生成V个子树。的真实误差用的平均值来得到，最终取误差最小的作为最优子树。

如果使用独立的剪枝验证集，就是对第一步中子树序列的每棵树进行评估，获取最佳剪枝树。首先定义标准错误SE(standard error):



为剪枝集大小，为树对剪枝集的错分样本数，则,那么最佳的剪枝树是满足以下条件并且包含的节点数最小的那颗剪枝树：

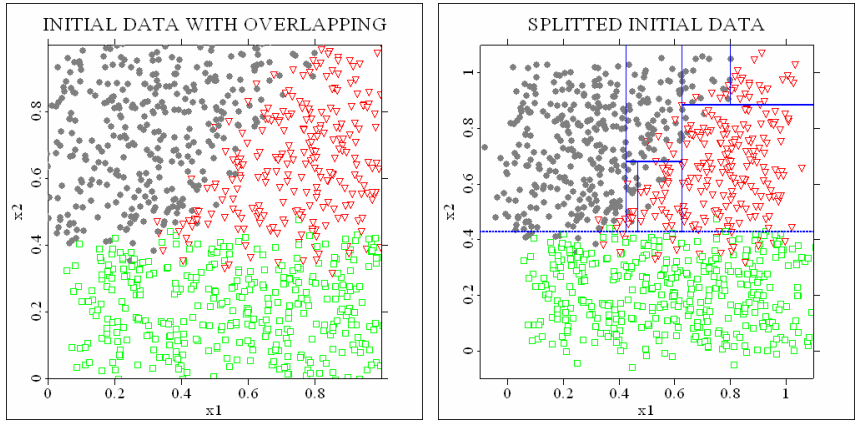
## 2.3.CART算法优缺点

优点：

1. CART很容易处理数值型和分类型变量，不需要过多的特征预处理，即使很多相关性比较大的特征同时放入模型也能够自动进行特征选择；
2. 对异常值有很强的鲁棒性
3. 解释性强
4. 决策树属于非参数方法，不需要任何先验建设，不假定类和其他属性服从一定的概率分布

缺点：

1. 和其他单棵决策树一样，CART也可能产生不稳定的树，训练数据稍微有一些变化就可能产生不同的树，这和决策树对特征空间的划分思想有关。比如一个连续变量t的最佳划分点取值是3，而两个样本在t上的取值分别是2.9和3.1，它们就会划分到完全不同的空间里面去，所以我一般习惯在决策树算法中将连续变量离散化;
2. 决策树可能产生次优解，决策树只能做垂直于坐标轴的划分，而有些数据是不适合进行这样的划分的，如下图:



正是由于单棵决策树存在上面的一些缺点，C5.0、随机森林这些组合树算法就慢慢出现来解决这些问题

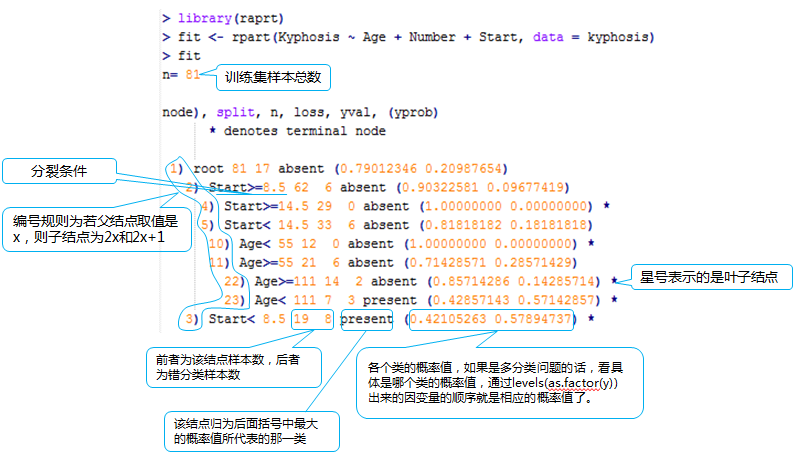
# 3.CART算法R实现

R中实现CART算法的主要package是rpart，本文主要讲述该包里面函数rpart的参数含义，以及对输出结果进行解读。

## 3.1.生成树(见CART\_algorithm.r)

## 3.2.决策树剪枝(见CART\_algorithm.r)

## 3.3.模型结果说明



## 3.4.运用模型进行预测(见CART\_algorithm.r)