

第16课：决策树——既能分类又能回归的模型

### 决策树

前面我们讲了线性回归和朴素贝叶斯分类模型。前者只能做回归，后者只能做分类。但本文中要讲的决策树模型，却既可以用于分类，又可以用于回归。

#### 什么是决策树

决策树是一种非常基础又常见的机器学习模型。

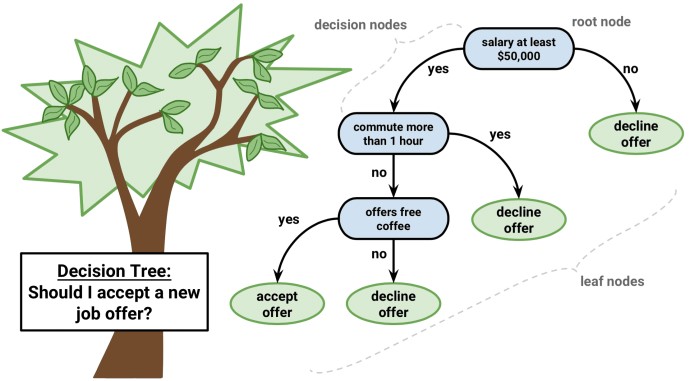
一棵决策树（Decision Tree）是一个树结构（可以是二叉树或非二叉树），每个非叶节点对应一个特征，该节点的每个分支代表这个特征的一个取值，而每个叶节点存放一个类别或一个回归函数。

使用决策树进行决策的过程就是从根节点开始，提取出待分类项中相应的特征，按照其值选择输出分支，依次向下，直到到达叶子节点，将叶子节点存放的类别或者回归函数的运算结果作为输出（决策）结果。

决策树的决策过程非常直观，容易被人理解，而且运算量相对小。它在机器学习当中非常重要。如果要列举“十大机器学习模型”的话，决策树应当位列前三。

#### 直观理解决策树

下图是一个决策树的例子：



这棵树的作用，是对要不要接受一个 Offer 做出判断。

我们看到，这棵树一共有7个节点，其中有4个叶子节点和3个非叶子节点。它是一棵分类树，每个叶子节点对应一个类别。

那么有4个叶子节点，是说一共有4个类别吗？当然不是！从图中我们也可以看出，总共只有2个类别：accept offer（接受）和 decline offer（拒绝）。

理论上讲，一棵分类树有 n 个叶子节点（n>1，只有一个结果也就不用分类了）时，可能对应2~n 个类别。不同判断路径是可能得到相同结果的（殊途同归）。

以上例而言，拿到一个 Offer 后，要判断三个条件：（1）年薪；（2）通勤时间；（3）免费咖啡。

这三个条件的重要程度显然是不一样的，最重要的是根节点，越靠近根节点，也就越重要——如果年薪低于5万美元，也就不用考虑了，直接 say no；当工资足够时，如果通勤时间大于一个小时，也不去那里上班；就算通勤时间不超过一小时，还要看是不是有免费咖啡，没有也不去。

这三个非叶子节点（含根节点），统称****决策节点****，每个节点对应一个条件判断，这个条件判断的条件，我们叫做****特征****。上例是一个有三个特征的分类树。

当我们用这棵树来做判断一个 Offer 的时候，我们就需要从这个 Offer 中提取年薪、通勤时间和有否免费咖啡三个特征出来，将这三个值（比如：[ $ 65,000，0.5 hour, don’t offer free coffee]）输入给该树。

该树按照根节点向下的顺序筛选一个个条件，直到到达叶子为止。到达的叶子所对应的类别就是预测结果。

#### 构建决策树

决策树的作用过程是很简单的，那么决策树是如何构造的呢？

前面讲了，获得一种模型的过程叫训练，那么我们如何训练可以得到一棵决策树呢？

简单讲，有以下几步：

1. 准备若干的训练数据（假设 m 个样本）；
2. 标明每个样本预期的类别；
3. 人为选取一些特征（即决策条件）；
4. 为每个训练样本对应所有需要的特征生成相应值——数值化特征；
5. 将通过上面的1-4步获得的训练数据输入给训练算法，训练算法通过一定的原则，决定各个特征的重要性程度，然后按照决策重要性从高到底，生成决策树。

那么训练算法到底是怎么样的？决定特征重要程度的原则又是什么呢？

### 几种常用算法

决策树的构造过程是一个迭代的过程。每次迭代中，采用不同特征作为分裂点，来将样本数据划分成不同的类别。被用作分裂点的特征叫做****分裂特征****。

选择分裂特征的****目标****，是让各个分裂子集尽可能地“纯”，即尽量让一个分裂子集中的样本都属于同一类别。

如何使得各个分裂子集“纯”，算法也有多种，这里我们来看几种。

#### ID3 算法

我们先来看看最直接也最简单的 ****ID3 算法****（Iterative Dichotomiser 3）。

该算法的****核心****是：****以信息增益为度量，选择分裂后信息增益最大的特征进行分裂****。

首先我们要了解一个概念——信息熵。

假设一个随机变量 x 有 n 种取值，分别为 {x1,x1,...,xn}{x1,x1,...,xn}，每一种取值取到的概率分别是 {p1,p2,...,pn}{p1,p2,...,pn}，那么 x 的信息熵定义为：

Entropy(x)=−∑ni=1pilog2(pi)Entropy(x)=−∑i=1npilog2⁡(pi)

熵表示的是信息的混乱程度，信息越混乱，熵值越大。

设 S 为全部样本的集合，全部的样本一共分为 n 个类，则：

Entropy(S)=−∑ni=1pilog2(pi)Entropy(S)=−∑i=1npilog2⁡(pi)

其中，pipi 为属于第 ii 个类别的样本，在总样本中出现的概率。

接下来要了解的概念是****信息增益****，信息增益的公式为（下式表达的是样本集合 S 基于特征 T 进行分裂后所获取的信息增益）：

InformationGain(T)=Entropy(S)−∑value(T)|Sv||S|Entropy(Sv)InformationGain(T)=Entropy(S)−∑value(T)|Sv||S|Entropy(Sv)

其中：

* SS 为全部样本集合，|S|为S|S|为S 的样本数；
* TT为样本的一个特征；
* value(T)value(T) 是特征 TT 所有取值的集合；
* vv 是 TT 的一个特征值；
* SvSv 是 SS 中特征 T 的值为 v 的样本的集合，|Sv|为Sv|Sv|为Sv 的样本数。

#### C4.5

前面提到的 ID3 只是最简单的一种决策树算法。

它选用信息增量作为特征度量，虽然直观，但却有一个很大的****缺点****：ID3一般会优先选择取值种类较多的特征作为分裂特征。

因为取值种类多的特征会有相对较大的信息增益——信息增益反映的是给定一个条件以后不确定性被减少的程度，必然是分得越细的数据集确定性更高。

被取值多的特征分裂，分裂成的结果也就容易细；分裂结果越细，则信息增益越大。

为了避免这个不足，在 ****ID3 算法****的基础上诞生了它的****改进版本：C4.5 算法****。

C4.5 选用****信息增益率****（Gain Ratio)——用比例而不是单纯的量——作为选择分支的标准。

信息增益率通过引入一个被称作****分裂信息（Split Information）****的项，来惩罚取值可能性较多的特征。

SplitInformation(T)=−∑value(T)|Sv||S|log|Sv||S|SplitInformation(T)=−∑value(T)|Sv||S|log⁡|Sv||S|

GainRatio(T)=InformationGain(T)SplitInformation(T)GainRatio(T)=InformationGain(T)SplitInformation(T)

****ID3 还有一个问题****：就是不能处理取值在连续区间的特征。例如上面例子里，假设训练样本有一个特征是年龄，取值为（0，100）区间内的实数。ID3 就不知如何是好了。

****C4.5**** 在这方面也有****弥补****，具体做法是:

* 把需要处理的样本（对应整棵树）或样本子集（对应子树）按照连续变量的大小从小到大进行排序。
* 假设所有 m 个样本数据在特征上的实际取值一共有 k（k<=m）个，那么总共有 k−1 个可能的候选分割阈值点，每个候选的分割阈值点的值为上述排序后的特征值中两两前后连续元素的中点。根据这 k-1 个分割点把原来连续的一个特征，转化为 k-1 个 Bool 特征。
* 用信息增益率选择这 k-1 个特征的最佳划分。

但是，C4.5 有个问题：当某个 |Sv||Sv| 的大小跟 |S||S| 的大小接近的时候：

SplitInformation(T)→0,GainRatio(T)→∞SplitInformation(T)→0,GainRatio(T)→∞

为了避免这种情况导致某个其实无关紧要的特征占据根节点，可以采用启发式的思路，对每个特征先计算信息增益量，在其信息增益量较高的情况下，才应用信息增益率作为分裂标准。

C4.5 的优良性能和对数据和运算力要求都相对较小的特点，使得它成为了机器学习最常用的算法之一。它在实际应用中的地位，比 ID3 还要高。

#### CART

ID3 和 C4.5 构造的都是分类树。还有一种算法，在决策树中应用非常广泛，它就是 CART 算法。

****CART 算法****的全称是： Classification and Regression Tree 分类和回归树。从这个名字一望可知，它不仅可以用来做分类，还可以用来做回归。

CART 算法的运行过程和 ID3 及 C4.5 大致相同，不同之处在于：

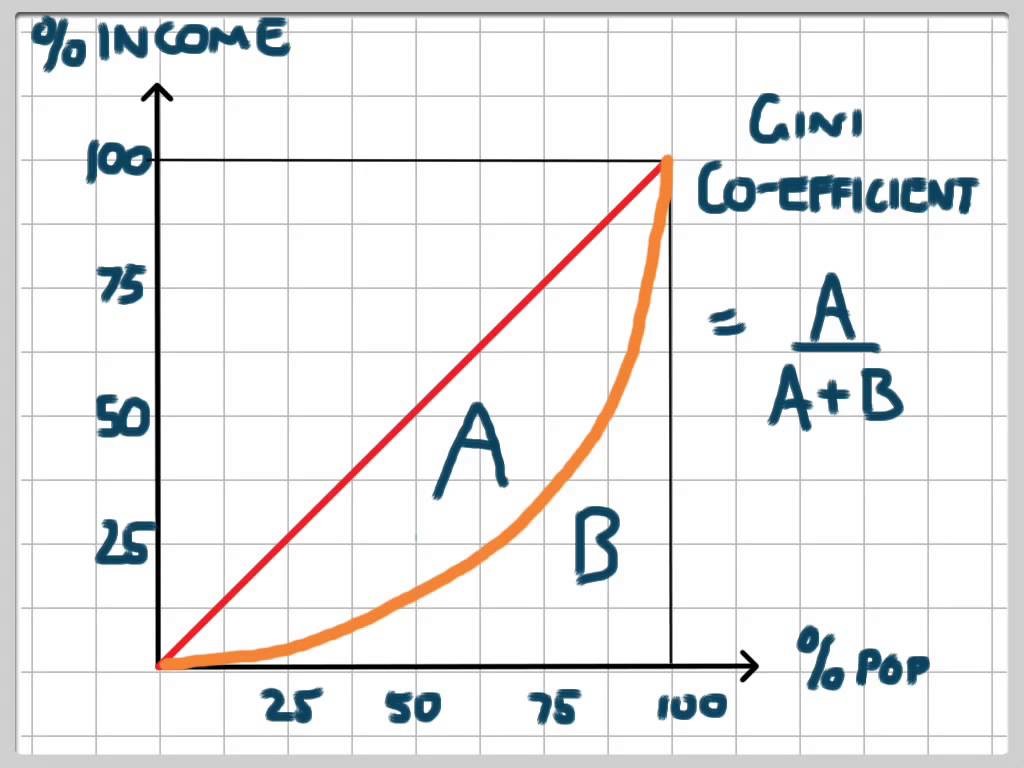
1. CART 的特征选取依据不是增益量或者增益率，而是 Gini 系数（Gini Coefficient）。每次选择 Gini 系数最小的特征作为最优切分点。
2. CART 是一棵严格二叉树。每次分裂只做二分。

这里面要特别提到概念：****Gini 系数****（Gini Coefficient）。

Gini 系数原本是一个统计学概念，20世纪初由意大利学者科拉多·基尼提出，是用来判断年收入分配公平程度的指标。Gini 系数本身是一个比例数，取值在0到1之间。

当 Gini 系数用于评判一个国家的民众收入时，取值越小，说明年收入分配越平均，反之则越集中。当 Gini 系数为0时，说明这一个国家的年收入在所有国民中平均分配，而当 Gini 系数为1时，则说明该国该年所有收入都集中在一个人手里，其余的国民没有收入。

在 Gini 系数出现之前，美国经济学家马克斯·劳伦茨提出了“收入分配的曲线”（又称****劳伦茨曲线****）的概念。下图就是一条劳伦茨曲线：



图中横轴为人口累计百分比，纵轴为该部分人的收入占全部人口总收入的百分比，红色线段表示人口收入分配处于绝对平均状态，而橘色曲线就是劳伦茨曲线，表现的是实际的收入分配情况。

我们可以看出，横轴75%处，如果依据红色线段，对应的纵轴也是75%，但是按照橘色曲线，则对应纵轴只有不到40%。

A 是红色线段和橘色曲线所夹部分面积，而 B 是橘色曲线下部分的面积。****Gini 系数实际上就是****AA+BAA+B****的比值****。这个概念在经济学领域远比在机器学习中有名。

Gini 系数的计算方法是：

Gini(p)=∑ni=1pi(1−pi)=1−∑ni=1p2iGini(p)=∑i=1npi(1−pi)=1−∑i=1npi2

对于二分类问题，若样本属于第一类的概率是 pp，则：

Gini(p)=2p(1−p)Gini(p)=2p(1−p)

这时，如果 p = 0.5， 则 Gini 系数为0.5；如果 p = 0.9， 则 Gini 系数为0.18。0.18 < 0.5，根据 CART 的原则，当 p=0.9 时，这个特征更容易被选中作为分裂特征。

由此可见，对二分类问题中，两种可能性的概率越不平均，则越可能是更佳优越的切分点。

上面的例子虽然用的是二分类，但实际上，对于多分类，趋势是一样的，那些概率分布在不同可能性之间越不平均的特征，越容易成为分裂特征。

到了这里，可能有朋友会误会，认为我们一直说的都是用 CART 做分类时的做法。但是实际上，无论是做分类还是做回归，都是一样的。

回归树和分类树的区别在于最终的输出值到底是连续的还是离散的，每个特征——也就是分裂点决策条件——无论特征值本身是连续的还是离散的，都要被当作离散的来处理，而且都是被转化为二分类特征，来进行处理：

* 如果对应的分裂特征是连续的，处理与 C4.5 算法相似；
* 如果特征是离散的，而该特征总共有 k 个取值，则将这一个特征转化为 k 个特征，对每一个新特征按照是不是取这个值来分 Yes 和 No。

****注意：****还有一个词——Gini 指数（Gini Index），经常在一些资料中被提及，并在 CART 算法中用来代替 Gini 系数，其实 Gini 指数就是基尼系数乘100倍作百分比表示，两者其实是一个东西。[邀请好友一起学，获得 25% 返现奖励](https://gitbook.cn/m/mazi/columns/5bc6ac7442d7d32f50f19a98/topics/5bf2823ffd72950cafdccd94)

[IMG_259](https://gitbook.cn/m/mazi/comp/column?columnId=5bc6ac7442d7d32f50f19a98%26tag=2#catalog)