0、概述

决策树是一种基于“规则”的机器学习模型，它既可以用于分类也可以用于回归（用于分类时称为分类树，用于回归时称为回归树）。决策树比较容易过拟合，但是以决策树为基学习器的集成学习模型（随机森林、GBDT）效果非常好，广泛地应用于实际的分类和回归问题中。

下面我们先以分类树为例介绍决策树的结构、训练过程和预测过程，然后再介绍回归树。

一、决策树的结构

决策树的结构与一般的树结构无异，由诸多的节点和连接节点的边组成。它从一个节点出发，分裂出多个子节点，每个子节点又分裂出它们的子节点，如此周而复始，直至达到预先设定的条件而停止分裂。分裂开始的节点称为根节点，根节点没有父节点；分裂结束的节点称为叶节点，叶节点没有子节点；其余节点称为中间节点，中间节点既有子节点也有父节点。

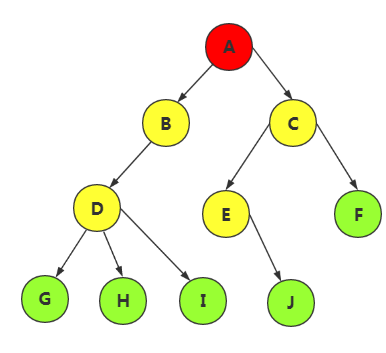


图1 树结构示意图

图1为一个树结构，红色节点为根节点，绿色为叶节点，黄色为中间节点。

基于树结构，分类树就可以实现对样本进行分类的功能。那么具体是如何实现的呢？与一般的机器学习分类模型一样，分类树需经历训练过程来构建具体的模型，并通过预测过程来对给定的样本进行分类。

二、分类树的训练和预测

**1、训练过程**

分类树的训练过程就是其分裂的过程（或者说“伸展”的过程）。

给定训练集，训练集中包含多个样本，每个样本有多维属性并有一个类别标签。首先将所有样本放在根节点，然后选择一个特征和基于该特征的一个分裂规则（特征和规则是依据某种策略而确定的），按照规则分裂出多个子节点，将根节点中的样本分到这些子节点中（相当于对集合进行一次划分）；对于每个子节点，重复上述过程，即针对该子节点，选择一个特征和基于该特征的一个分裂规则，…。这样递归地分裂分类树（相当于递归地划分集合），不断地将样本分入子节点，直至达到预先设定的某个条件，就停止继续分裂。这时，所有的样本都位于分类树的叶节点中。最后，将每个叶节点中多数样本的类别定义为该叶节点的类别，该类别也就作为模型对该叶节点中样本的输出。

**2、预测过程**

以上介绍了决策树训练的整个过程。训练结束后，我们就可以利用训练好的分类树对一个给定的样本进行分类：首先将该样本放在根节点，然后按照每个节点对应的分裂规则，将样本不断向下移动，最后移动到某个叶节点，则该样本的类别就设定为该叶节点的类别。

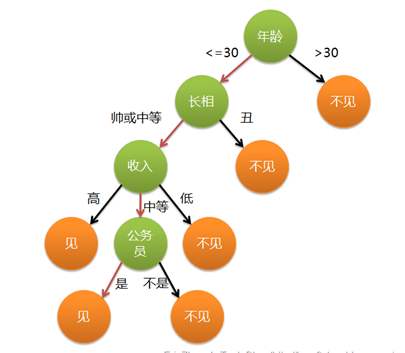


图2 决策树示意图

如图2所示是一个决策树示意图，这是一个拜金女相亲的决策树。从中可以清晰地看出对于一个样本（待相亲对象），决策树如何决定它的类别（见面或不见）。

**3、节点分裂的策略**

决策树的训练过程中最重要的步骤是选择特征和相应的规则来完成每个节点分裂。那么针对某个节点，我们如何选择相应的最优特征和规则呢？需要遵循什么标准？有什么依据？

首先我们应明确我们的目标是什么，然后才能基于目标确定如何操作。

根据上面介绍的决策树的思想可以看到，决策树本质上是对样本集合进行划分，最终将整个样本的集合划分成叶节点对应的多个集合。所以我们终极目标是确定一种最优的样本集合的划分，使得划分后的子集中样本尽量都属于同一类别。因为我们对一个待预测的样本进行分类时，该样本落入的叶节点中的样本尽量属于同一类别时，我们才有比较大的把握说该样本属于相应类别。

很自然地想到，可以将一个样本集合划分后的子集的“纯度”最大化作为目标。一个样本集合的纯度的意思是，该集合中样本类别越少，样本关于类别的分布越集中（即大多数样本都属于少数的几个类别，只有少量样本属于其他类别），则纯度越高；而集合中样本类别越多，且样本关于类别的分布越均匀，则纯度越低。

考虑到样本集合划分得到的子集有多个，每个子集纯度不同，因此我们应当使用所有子集的“综合纯度”作为目标。对“集合纯度”的量化主要两个指标：信息熵和基尼系数。

确定了目标以后，我们如何在每一个节点的分裂中具体地选择相应的特征和规则呢？

如果只考虑分类树的整体目标的话，其实很容易实现，即将树分裂到每个叶节点只包含一个样本为止。这样在每次节点分裂时，随便选择即可（因为最终都会达到单样本的叶节点）。但是这样势必造成过拟合，所以我们还希望整棵树的规模越小越好。所以最好的策略是基于贪心策略：每次分裂（即一次集合划分）时，最大化该划分的综合纯度。也就是说，对于任何一个节点的分裂，我们以最大化这一次分裂后的子节点的综合纯度为目标（当前目标），选择最优的特征和相应的规则。按照这种策略进行分裂，分类树能够尽快停止分裂（或者说尽快收敛）。

对于离散型特征，我们选定最优特征后，相应的分裂规则就是：将节点中的样本按照最优特征的取值分到子节点中（比如最优特征A有N个取值，则节点分裂为N个子节点，每个子节点中样本在特征A上的取值分别是）；

对于连续型特征，一般使用二分的处理方法，即选定最优特征后，相应的分裂规则就是：根据特征的取值，设置一个分割点，然后将特征值小于分割点的样本划分到左子节点，特征值大于分割点的样本划分到右子节点。

可以看出离散型特征与连续型特征的区别：

（1）使用连续型特征进行节点分裂，不仅要选择最优特征，还要选择最优分割点，而使用离散型特征进行节点分裂，则（按照最简单的处理方式来说）只需要选择最优特征；

（2）按照最简单的处理方式来说，连续型特征形成二叉树，离散型特征形成多叉树（当然，连续型特征也可以形成多叉树（设置多个分割点），离散型特征也可以形成二叉树（按特征是否取某一可能值，可将集合分成两部分））；

（3）在决策树分裂的过程中，离散型特征不能多次使用，连续型特征可以多次使用。

这就是节点分裂的基本策略。

三、回归树

上面详细地介绍了分类决策树的工作机制，实际上，决策树也可以实现回归的任务，这就是“回归树”。

回归树的分裂过程与分类树相同，对每个节点，选取一个特征和相应的规则进行分裂，并将节点中的样本划分到子节点中（但是，回归树的节点分裂策略与分类树不同，它不以子节点综合纯度最大为目标确定每一个节点分裂的特征和相应规则，而是以最小化MSE为目标）。

与分类树每个叶节点都有一个“输出类别”相对应，回归树的每个叶节点都有一个“输出值”，输出值为该叶节点中所有样本的标签值的平均。

利用回归树对给定的样本作预测，就是先将样本放在回归树的根节点，然后按照回归树的节点分裂规则，将样本不断向下移动，最终移动到叶节点，然后将叶节点的输出作为该样本的输出。

四、决策树分裂性能指标

上面介绍了分类树和回归树的训练过程，重点是节点分裂的策略：对于分类树，节点分裂以最大化子节点的综合纯度为目标，对于回归树，则以最小化子节点的MSE为目标。

但只给出一般性的策略还不够，我们并未实现这些目标的具体形式。我们还需要根据一般性的策略，给出具体的指标，并基于指标导出具体的目标函数，才能进一步构建出具体的训练算法。

下面我们先将这些目标进行形式化，得到分类树和回归树的节点分裂性能指标：

**1、分类树的指标**

上面已经提到，使用信息熵和基尼系数可以衡量集合的“纯度”，则使用多个集合的综合信息熵或综合基尼系数可以衡量“综合纯度”，从而衡量集合划分的效果（也就是决策树的性能）。信息熵在其他笔记中已有记叙，此处不再详解，只给出基尼系数的定义：

样本集合D的基尼指数的定义为：，基尼指数越大，则样本集的纯度越小。

我们上面提到，决策树分裂的基本策略是贪心策略，因此我们不仅应给出决策树的整体指标，还应给出每一次节点分裂的指标。基尼系数不仅可以作为整体指标，也可以扩展为单次节点分裂指标：



上式表示集合D关于特征A的基尼系数，其中是用特征A对集合D划分后得到的子集。

对于单次分裂的性能指标，我们常用的还有“信息增益”和“信息增益率”。（实际上，这两个指标也可以扩展成为整体指标）

（1）信息增益

信息增益是基于信息熵的一个指标。

设样本集为D。定义样本的特征A给D带来的信息增益为：



其中H(D)是集合D的信息熵，，（表示集合D的样本个数，表示集合D中类别为k的样本个数）；

H(D|A)是条件熵，它衡量了基于特征A的取值对D划分后，子集们的综合纯度，，（N是A特征取值的个数），其取值越大，则纯度越低，取值越小，则纯度越高。

可以看出，信息增益g(D,A)衡量了使用特征A对集合D进行划分，使得划分后的集合的综合纯度相比集合D本身的纯度的增加量。因此，g(D,A)越大则用特征A对D进行划分越好。

（2）信息增益率

信息增益率是在信息增益的基础上进行的改进。信息增益作为衡量特征A对样本集合D划分效果的指标，存在一个缺陷，那就是它倾向于对取值较多的特征给出较高的评分（这是因为特征取值越多，则划分的子集就越多，每个子集中样本个数就越少，其纯度就越高），但是我们又不希望使用取值太多的特征，因为那会使划分的子节点较多，从而使树的规模太大。因此我们对信息增益这个指标进行改进，得到信息增益率：



其中（N是特征A的取值个数），它表示用特征A的取值（而不是使用集合D中样本本身的分类）对集合D进行划分后的子集们的综合纯度，显然，特征A取值越多，越大。

注意，上面介绍的指标只适用于特征为离散特征的情况。对于每个节点，依照指标最大的目标选定特征后，其分裂规则为：对应于特征的每个取值，分裂出一个子节点。

这样，给定任何一种指标，我们就能顺利地在训练集上训练一棵分类树：对于树的每个节点，我们选择能使分裂后的子节点的指标最大的特征和规则进行分裂。

**2、回归树的指标**

我们上面已经提到回归树分裂的目标是最小化MSE，具体来说，就是：当一个节点分裂时，可以选择不同的特征和分裂规则进行分裂，分裂得到的每个样本标签值与模型对该样本的输出（也就是样本所在的叶节点的输出）之间存在误差，然后求所有样本的误差平方的均值，即为MSE。最小化MSE，其实就是最小化当前的模型输出与样本真实标签之间的误差。所以MSE是衡量回归树分裂性能的一个简单有效的指标。

五、决策树的目标函数

基于具体的指标，我们很自然地给出决策树的目标函数。前面已经多次说到，我们采用贪心策略训练决策树，因此这里我们不介绍决策树的整体目标函数，只介绍单次分裂的目标函数。

1、信息增益的目标函数

最大化信息增益，则优化问题为：



其中D是当前节点样本集，A是样本的一个特征。

2、信息增益率的目标函数

最大化信息增益率，则优化问题为：



3、基尼系数的目标函数

最大化基尼系数，则优化问题为：



注意，这里的优化问题只给出了优化特征A，若想对特征进行二分处理，则可以加入特征分割点（连续型特征）或特征取值（离散型特征）的最优化。

六、决策树训练算法

基于目标函数，就能构建具体的决策树训练算法。

常用的决策树训练算法有以下三种：

**1、ID3**

基于信息增益目标函数的算法。每次节点分裂时，选择最大化信息增益的特征A，并按相应的规则进行分裂。

**2、C4.5**

基于信息增益率目标函数的算法。注意，信息增益率准则对可取值数目较少的特征有所偏好，所以C4.5算法并不是直接选择使信息增益率最大的特征，而是使用了一个启发式：先从候选划分属性中找出信息增益高于平均水平的属性，再从中选择增益率最高的。

**3、CART**

ID3算法和C4.5算法都是只适用于分类树。CART则也适用于回归树。

CART全名是“分类和回归树”，它是二叉树结构，是基于基尼系数和MSE目标函数的算法。

（1）分类树生成（假设特征为离散特征）

对每个节点（设其样本集为D），根据优化问题：



求出最优特征A和取值a，并按A和a进行节点分裂。

其中和是根据A和a将D分成的两个集合，其中是D中特征A取值为a的样本，是D中特征A取值不为a的样本。

（2）回归树生成（假设特征为连续特征）

对每个节点（设其样本集为D），根据优化问题：



求出最优的变量j和最优分割点s，并按j和s进行分裂。

其中R1和R2是样本集D按照j和s分割成的两个子集，c1和c2是两个子集上的输出（也就是两个子集中样本标签值的平均）。

七、决策树的剪枝

决策树的剪枝就是从已经生成的树上剪掉一些子树或者叶节点。它的目的是对决策树进行简化，从而减轻过拟合。

设决策树为T，则其剪枝的目标函数为：



其中，是叶节点的个数，Nt是第t个叶节点中的样本个数，Ht(T)是第t个叶节点的信息熵。表示对各叶节点的信息熵的加权求和，可以衡量该决策树的整体分类性能；则表示决策树的规模。所以该目标函数平衡了决策树的性能和规模。

剪枝时，递归地进行以下操作：对每个叶节点，计算剪枝前后的目标函数，若目标函数减小，则剪掉该叶节点。

八、决策树的超参数

决策树有很多的超参数，它们控制着决策树的训练过程和剪枝过程，以使它的规模不至于变得太复杂而造成过拟合。

主要的超参数有以下几个：

1、最大深度（max\_depth）：即从根节点到各叶节点需要经过的中间节点的最大个数，若节点使决策树深度；

2、节点分裂所需的最小样本个数（min\_samples\_split）:当节点中的样本个数小于该参数时，节点停止分裂。

3、节点分裂所需的其子节点的最小样本个数（min\_samples\_leaf）：若节点继续分裂产生的任何子节点中样本个数小于该参数，则节点停止分裂。

九、决策树用于特征选择

决策树实际上相当于用来划分样本集的一组规则。决策树的训练过程，也就是它的生成过程，实际上是这组规则生成的过程。这些规则其实就是选定一个当前最优特征后，在该特征上选择一个最优特征分割。

由于这些规则是采用贪心策略而生成，因此相当于对特征的好坏进行了排序，越接近根节点的规则对应的特征，越是好特征。所以决策树可以用来对数据进行特征选择。