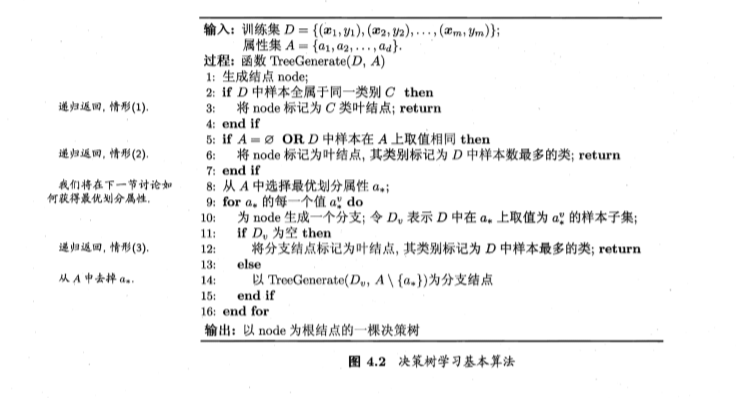
1.基本流程

决策树中根结点包含样本全集，内部结点表示一个属性测试，叶子节点代表一个决策的结果，整个的基本思想是分而治之

算法如下



决策树的生成是一个递归的过程，三种情况会有递归的返回值，（1）当前结点包含的样本全部属于一类，无需划分（2）当前的属性集为空，或者当前结点所有样本在属性集合上的取值相同（3）当前结点的样本集为空不可分

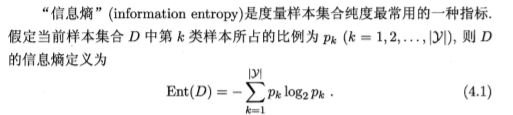
（2）情况将当前结点设为叶节点，其类别设为此节点所含样本最多的类别

（3）情况将当前结点设为叶节点，类别设置为父节点所含样本最多的类别，假设取得当前属性值的样本是父节点中含样本最多的类

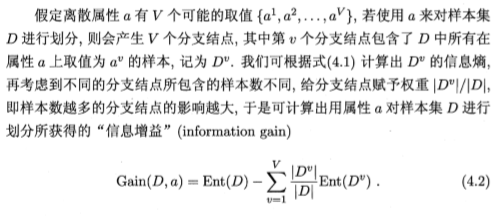
2.划分选择

在算法中我们可以发现第8步如何选择最优划分属性是很重要的一步，那么如何找到最优的划分属性进行划分呢，两个概念

信息熵：



信息增益：



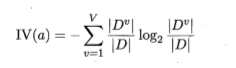
信息熵与信息增益根据定义都是正值，信息熵代表的是样本集合D的纯度，如果信息熵越小那么纯度越高，分类的目标就是为了获得更纯的子集来达到划分的目的，信息增益则是代表了选择属性a作为划分属性的纯度提升程度，信息增益越大说明用a来划分就越好

那么仅仅使用信息增益会导致一个问题，例如对于属性取值较多的属性，它纯度就可能越高，那么如果样本较少可能选择此属性可能就比较大，但是泛化性能就会降低很多，训练出来的模型也没什么用，所以在这里采取一个比较好的策略一种新的评判标准增益率

与信息增益偏好可取值数目多的属性相反，增益率对可取值数目少的属性有所偏好，定义为



其中

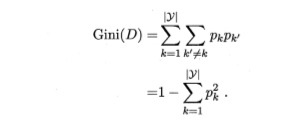


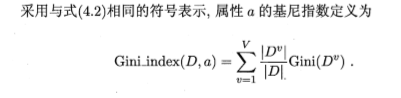
称为属性a的固有值，属性a的可取值数目越多固有之越

C4.5算法先从划分属性中找出信息增益高于平均水平的属性，再选增益率最高的可以有效的避免信息增益的缺点

CART决策树（分类回归决策树两者皆可使用的决策树算法）

使用的判别指数是基尼指数，符号同信息熵和信息增益





那么基尼指数代表的就是随机取样本集中两个样本其样本标记不同的概率，基尼指数越小纯度越高，所以对应第二个式子应该选使得基尼指数最小的属性作为划分属性

3.剪枝

对于在决策树的划分过程中，若是划分过多，对于样本集学得太好就会造成泛化能力差，那么解决的办法就是进行剪枝操作，分为预剪枝和后剪枝

预剪枝是在决策树的生成过程中，如果当前划分不能使得决策树的泛化性能提升，那么就不进行划分（奥卡姆剃刀）；后剪枝则是在整个决策树生成后，然后自底向上对非叶节点进行考察，如果将该节点对应子树替换为叶节点能够获得泛化性能的提升那么可将该子树替换为叶节点

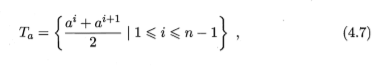
如何判断泛化性能提高呢，可采用留出法预留验证集

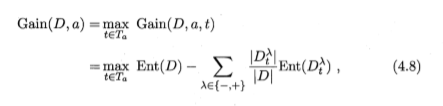
预剪枝就是在每轮划分之后用验证集计算验证集精度，如果划分使得验证集精度下降了那么就不对其进行划分，预剪枝可以有效的避免过拟合以及训练及测试的时间花销，但是有些划分也许当前会降低精度但是其后续划分可提高所以这样就会存在一个欠拟合的问题

那么对于后剪枝，自底向上的考察对应的非叶子结点，如果剪掉其子树能够提高精度，则剪掉那么对于这种方法欠拟合与过拟合都比较下，但是如果数很复杂则测验的时间花销会很大远超过预剪枝以及不剪枝

4.连续与缺失值

对于连续值的处理，例如对于一个属性a其属性值为n个为连续的，那么我们可以根据一个划分点t，将属性取值分为大于t和小于t的两部分再根据划分开的属性将样本集D划分为两部分，易知在ai+1和ai之间取任意t都可以，那么我们可以取两者的中间值，对于n个属性值就有n-1个候选划分点，对应划分点进行信息增益的求解，公式进行一些小变化



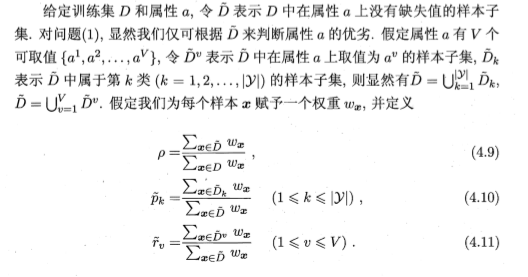


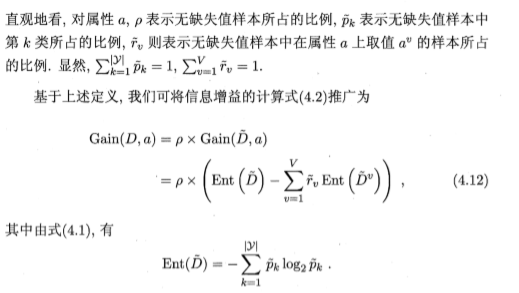
与离散型不同连续型的属性还可作为其后代节点的划分属性

对于缺失值要解决的问题一共有两个

（1）如何在属性值缺失的情况下进行属性划分选择

（2）给定划分属性，若样本在该属性上的值缺失，如何对样本进行划分





简而言之就是用无缺集与全集之间的比值来平衡无属性值样本的影响

那么对于第二个问题，划分问题是根据权值来进行的，一开始权值均设为1进行分化之后遇到某个属性值缺失的样本将其分类到所有子节点中，并且将样本权值更新为，对应子节点的，直观的看这就是让同一样本以不同概率划分到不同子节点中去

5.多变量决策树

多变量决策树的分类策略与普通决策树不同的是，普通决策树是根据对很多属性进行测试得出的分类结果，而多变量决策树是对属性的线性组合进行测试即其中是属性的权重,和t可在该结点所含的样本集和属性集上学得。于是,与传统的“单变量决策树”不同,在多变量决策树的学习过程中，不是为每个非叶结点寻找一个最优划分属性,而是试图建立一个合适的线性分类器