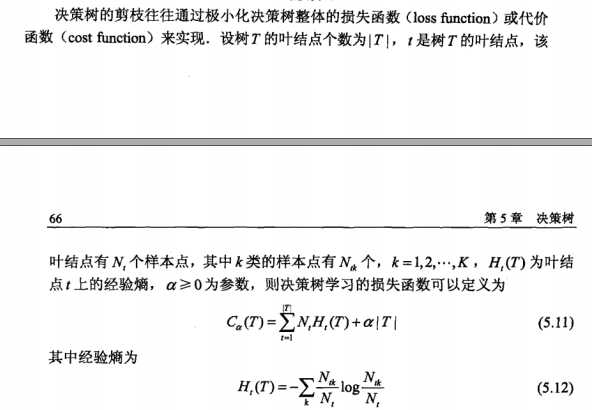
* **什么是决策树？**

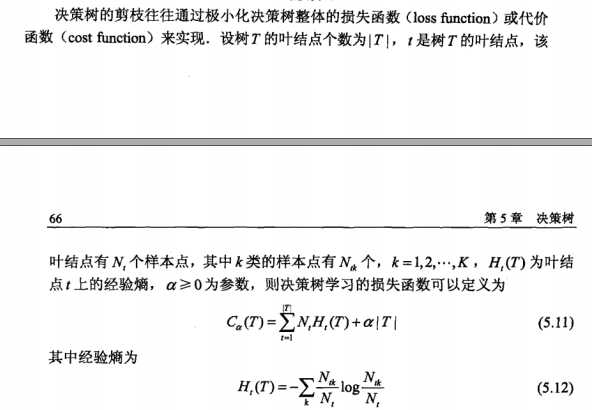
决策树是一种分类和回归的基本模型，可从三个角度来理解它，即：

* 一棵树
* **if-then规则的集合**，该集合是决策树上的所有从根节点到叶节点的路径的集合
* **定义在特征空间与类空间上的条件概率分布**，决策树实际上是将特征空间划分成了互不相交的单元，每个从根到叶的路径对应着一个单元。**决策树所表示的条件概率分布由各个单元给定条件下类的条件概率分布组成。**实际中，哪个类别有较高的条件概率，就把该单元中的实例强行划分为该类别。
* **和其他模型比，它的优点？**
* 模型具有可解释性
* 分类速度快
* **如何学习一棵决策树？**

**决策树的学习本质上就是从训练数据集中归纳出一组分类规则，使它与训练数据矛盾较小的同时具有较强的泛化能力。**

决策树的**损失函数通常是正则化的极大似然函数**，学习的策略是以**损失函数为目标函数的最小化**。所谓**正则化的极大似然函数**，就是如下的形式：





由于这个最小化问题是一个**NP完全问题**，现实中**，我们通常采用启发式算法（不求最后求可行解，遗传算法、退火、SMO…）来近似求解这一最优化问题，得到的决策树是次最优的。**

该启发式算法可分为三步：

**特征选择**

**模型生成**

**决策树的剪枝**

根据特征选择的差异，一般有ID3, C3.5, CART三种

* **ID3、C4.5、CART的区别?**
* ID3使用**信息增益**作为特征选择的标准、C4.5使用的是**信息增益比**、CART用的则是**GINI系数**或**平方误差最小**作为分类原则：

假设D是用于分类的样本集合，Di是其中第i类样本的个数。假设使用特征A对D进行划分。

**信息增益：**

http://latex.codecogs.com/gif.latex?gain(A)=info(D)-info_A(D)

http://latex.codecogs.com/gif.latex?info(D)=-%5Csum%20%5Em_%7Bi=1%7Dp_ilog_2(p_i)

http://latex.codecogs.com/gif.latex?info_A(D)=%5Csum%20%5Ev_%7Bj=1%7D%5Cfrac%7B%7CD_j%7C%7D%7B%7CD%7C%7Dinfo(D_j)

**信息增益比：**

http://latex.codecogs.com/gif.latex?gain%5C_ratio(A)=%5Cfrac%7Bgain(A)%7D%7Bsplit%5C_info(A)%7D

http://latex.codecogs.com/gif.latex?split%5C_info_A(D)=-%5Csum%20%5Ev_%7Bj=1%7D%5Cfrac%7B%7CD_j%7C%7D%7B%7CD%7C%7Dlog_2(%5Cfrac%7B%7CD_j%7C%7D%7B%7CD%7C%7D)

Split\_infoA(D)是数据集关于特征A的熵，其中v是特征A取值的个数

**CART回归树**：

假设已经将输入空间划分为M个单元R1，R2，…, RM, 并且在每个单元上有一个固定的输出值cm，于是回归树模型可以表示为：

则其**分类误差（损失）**为：

易知单元上的最佳取值是中对应所有实例的输出的均值。

所以CART回归树最核心的问题是，如何去划分区间，同样的需要用到启发式算法，选取第j个变量x(j)和它的取值s，作为切分变量和切分点，并定义两个区域：

寻找最优的切分变量和切分点，即求解：



CART回归树也称为最小二乘回归树，因为，回归树以**误差平方和**为准则选择最优二分切点。

**CART分类树**：

在分类问题中，假设有K类，样本点属于第k类的概率为,则概率分布的基尼指数定义为：



对于分类问题：设Ck为D中属于第k类的样本子集，则基尼指数为:



设条件A将样本D切分为D1和D2两个数据子集，则在条件A下的样本D的基尼指数为:



算法实现步骤：

1）计算现有样本D的基尼指数，之后利用样本中每一个特征A，及A的每一个可能取值a，根据A>=a与A<a将样本分为两部分，并计算Gini(D,A)值

2）找出对应基尼指数最小Gini(D,A)的最优切分特征及取值，并判断是否切分停止条件，否，则输出最优切分点

3）递归调用1）2）

* ID3和C4.5都只能用于分类，而CART可以做回归和分类
* ID3和C4.5生成的不一定是二叉树，而CART生成的是二叉树
* **从信息增益到信息增益率？**

使用信息增益进行划分，对于特征取值较多的特征，其信息增益会更大，因此会更偏向于这些特征，信息增量率是在信息增益的基础上，除以了数据集在相应特征上的熵，该值能反映特征的取值的多少。

* **决策树算法停止的条件？**
  + 当前节点的所有样本都属于同一类，则取相应类别作为该节点的类别；
  + 对于当前节点，没有更多的特征能用于划分，则结束算法，取最多类为类别；
  + 对于当前节点，如果树的深度达到设定的阈值，则停止计算，取最多类；
  + 对于当前节点，如果信息增益/信息增益比/GINI系数小于设定的阈值，则停止计算；
* **决策树剪枝算法？**

决策树剪枝的方法有两类，一种称为**预剪枝**，一种称为**后剪枝**。

预剪枝就是在每一步进行划分时，选择最优的划分后，需要用划分前后模型对验证数据的分类效果来进行评估，如果划分后验证准确率提高，则进行划分；所以这是一个从上到下的剪枝过程；

而后剪枝是在模型生成之后再进行的，它从下到上，逐个考察，每个节点，考察其变换为叶子节点前后模型在验证集上的分类效果，如果分类效果提升，则剪枝为叶子节点。

预剪枝因为很多分支没有展开，有可能会造成欠拟合。后剪枝的欠拟合的可能性更低，泛化性能一般更好。但是后剪枝的训练开销会比预剪枝要大。

* **决策树对特征缺失的处理？**

决策树中加入了对缺失值处理的策略，如果某个特征没有用于分类，则特征缺失没有影响，如果某个特征用于分类了，则如果该特征的某一样本出现了特征缺失，那么在划分的时候**，需要把这些样本划分到每个子节点中**。