**3.决策树**

**3.1定义**

一种描述基于特征对实例进行分类（或回归）的树型结构。决策树由结点(node)和有向边(directed edge)组成。结点有两种类型：表示一个特征或属性的内部结点(internal node)和表示一个类的叶结点(leaf node)。

从根结点开始，根据某个特征选择指标选择当前最优特征将实例按照该特征的取值分配到子结点，每个子结点对应该特征的一个取值，如此递归分配，直至到达叶结点，所有实例都被分到叶结点的类中。

主要优点：1）模型具有可读性；2）计算复杂度不高，分类速度快；3）对中间值的缺失不敏感，可以处理不相关特征数据；4）输出结果易于理解。

缺点：容易过拟合。

它可以认为是if-then规则的集合，也可以认为是定义在特征空间与类空间上的条件概率分布。

**3.1.1 if-then规则**

决策树转换成if-then规则的过程：

a）由决策树根结点到叶结点的每一条路径构建一条规则；

b）路径上的内部结点的特征对应规则的条件，叶结点对应规则的结论。

决策树的路径（对应的if-then规则集合）一个重要性质：

互斥且完备，即每个实例都被且只被一条路径或规则覆盖。

**3.1.2条件概率分布**

决策树还表示给定特征条件下类的条件概率分布，由将特征空间划分成互不相交的单元（cell）或区域（region）给定条件下类的条件概率分布组成。决策树的一条路径对应划分中的一个单元，每个单元定义一个类的概率分布。

**3.2决策树学习**

决策树学习的**本质**是从训练数据集中归纳出一组分类规则。与训练数据集不相矛盾的决策树（即能对训练数据进行正确分类的决策树）可能有多个，也可能一个没有。我们需要寻找一个与训练数据矛盾较小（训练误差小）且具有很好泛化能力的决策树。从所有可能的决策树中选取最优决策树是NP完全问题，所以决策树学习通常采用**启发式方法**，近似求解最优化问题，得到次最优解。

决策树学习的损失函数通常是**正则化的极大似然函数**。决策树的学习包括三个步骤：1）**特征选择**；2）**决策树的生成**；3）**决策树的修剪**。由于决策树表示一个条件概率分布，所以深浅不同的决策树对应不同复杂度的概率模型。决策树的生成对应模型的局部选择，决策树的剪枝对应模型的全局选择。

传统的决策树算法有ID3、C4.5与CART。主要不同在于特征选择的方法。

**3.2.1特征选择**

特征选择在于选取对训练数据具有分类能力的特征，提高决策树学习的效率。通常特征选择的准则是信息增益（ID3算法）或信息增益比（C4.5），CART算法通常选择最小二乘法（回归）或基尼系数（分类）。

**（1）信息增益**

信息增益表示得知特征X的信息而使得类Y的信息不确定性减少的程度。

a）定义

特征A对训练数据集D的信息增益g(D，A)为集合D的经验熵H(D)与特征A给定条件下D的经验条件熵H(D|A)之差：

g (D, A) = H (D) – H (D | A)

这个差一般称为互信息（mutual information），决策树学习中的信息增益等价于训练数据集中类与特征的互信息。

信息增益依赖于特征，不同的特征往往具有不同的信息增益。根据信息增益准则的特征选择方法是：对训练数据集（或子集）D，计算其每个特征的信息增益并比较大小，选择信息增益最大的特征。

**b）信息增益算法：**

设训练集为D，|D|表示其样本容量。设有K个类Ck，k=1，2，……，K，|Ck|为属于Ck的样本个数，。设特征A有n个不同的取值{a1，a2，……，an}，根据特征A的取值将D划分为n个子集D1，D2，……，Dn，|Di|为Di的样本个数，。记子集Di中属于Ck的样本集合为Dik，即Dik = Di∩Ck，|Dik|为Dik的样本个数。

输入：训练数据集D和特征A；

输出：特征A对训练数据集D的信息增益g(D, A)。

1）计算数据集D的经验熵H(D)

2）计算特征A对数据集D的经验条件熵H(D|A)

3）计算信息增益

g (D, A) = H (D) – H (D | A)

信息增益选择特征倾向于选择**取值较多**的特征。使用信息增益比（information gain ratio）可以对这一问题进行校正。

**（2）信息增益比**

特征A对训练数据集D的信息增益比gR(D, A)定义为其信息增益g(D, A)与训练数据集D关于特征A的熵HA(D)之比：

其中，，n是特征A的取值个数。算法流程与信息增益相同。

**3.2.2决策树的生成**

**（1）ID3算法**

ID3算法核心就是用信息增益准则选择特征，相当于用极大似然估计进行概率模型的选择。

输入：训练数据集D，特征集A，阈值ε；

输出：决策树T。

1）若D中所有实例属于同一类Ck，则T为单结点树，并将类Ck作为该结点的类标记，返回T；

2）若A=Ø，则T为单结点树，并将D中实例数最大的类Ck作为该结点的类标记，返回T；

3）否则，按照信息增益算法计算A中各特征对D的信息增益，选择信息增益最大的特征Ag；

4）如果Ag的信息增益小于阈值ε，则置T为单结点树，并将D中实例数最大的类Ck作为该结点的类标记，返回T；

5）否则，对Ag的每一个可能值ai，依Ag=ai将D分割为若干非空子集Di，将Di中实例数最大的类作为标记，构建子结点，由结点及其子结点构成树T，返回T；

6）对第i个子结点，以Di为训练集，以A-{Ag}为特征集，递归调用步1）~5），得到子树Ti，返回Ti。

ID3算法只有树的生成，所以该算法生成的树容易产生过拟合。

**（2）C4.5算法**

C4.5在生成过程中，用信息增益比来选择特征。

输入：训练数据集D，特征集A，阈值ε；

输出：决策树T。

1）若D中所有实例属于同一类Ck，则T为单结点树，并将类Ck作为该结点的类标记，返回T；

2）若A=Ø，则T为单结点树，并将D中实例数最大的类Ck作为该结点的类标记，返回T；

3）否则，按照信息增益比算法计算A中各特征对D的信息增益，选择信息增益最大的特征Ag；

4）如果Ag的信息增益比小于阈值ε，则置T为单结点树，并将D中实例数最大的类Ck作为该结点的类标记，返回T；

5）否则，对Ag的每一个可能值ai，依Ag=ai将D分割为若干非空子集Di，将Di中实例数最大的类作为标记，构建子结点，由结点及其子结点构成树T，返回T；

6）对第i个子结点，以Di为训练集，以A-{Ag}为特征集，递归调用步1）~5），得到子树Ti，返回Ti。

**3.2.3决策树的剪枝**

决策树生成直到不能继续下去为止，这样生成的树容易过拟合，需要在学习中将已生成的树进行简化（剪掉一些子树或叶结点，将其根结点或父结点作为新的叶结点），这个过程称为剪枝。剪枝往往通过极小化决策树整体的损失函数或代价函数来实现。树的剪枝有预剪枝和后剪枝。

设树T的叶结点个数为|T|，t是树T的叶结点，该叶结点有Nt个样本，其中k类的样本点有Ntk个，k=1，2，……，K，Ht(T)为叶结点t上的经验熵，α ≥ 0为参数，则决策树学习的损失函数

其中经验熵为

在损失函数中，将右端的第一项记作

这时有

Cα(T) = C(T) + α|T|

式子右端第一项表示训练误差，第二项是正则项，控制模型的复杂度。利用损失函数最小原则进行剪枝就是用正则化的极大似然估计进行模型选择。

**树剪枝算法：**

输入：生成算法产生的整个树T，参数α；

输出：修剪后的子树Tα。

（1）计算每个结点的经验熵；

（2）递归地从树的叶结点向上回缩。

设一组叶结点回缩到其父结点之前和之后的整体树分别为TB和TA，其对应的损失函数值分别为Cα(TB)和Cα(TA)，如果Cα(TA) ≤ Cα(TB)则进行剪枝，将父结点变成新的叶结点。

（3）返回（2），直到不能继续为止，得到损失函数最小的子树Tα。

**3.3 CART算法**

分类与回归树（classification and regression tree，CART）既可以用于分类也可以用于回归。CART是在给定输入随机变量X条件下输出随机变量Y的条件概率分布的学习方法，假设决策树是二叉树，内部结点特征取值为“是”和“否”，左分支是取值为“是”的分支，右分支是取值为“否”的分支。

**3.3.1 CART生成**

CART生成就是递归地构建二叉决策树的过程，回归树用**平方误差最小化准则**，分类树用**基尼指数最小化准则**进行特征选择。

（1）回归树的生成

假设X与Y分别为输入和输出变量，并且Y是连续变量，给定训练数据集

D = {(x1, y1), (x2, y2), ......, (xN, yN)}

一个回归树对应着输入空间（即特征空间）的一个划分以及在划分的单元上的输出值。假设已将输入空间划分为M个单元R1，R2，…，RM，并且在单元Rm上有一个固定输出值cm，回归树模型可以表示为

当输入空间划分确定时，可以用平方误差表示回归树对于训练数据的预测误差，用最小平方误差最小化原则求解每个单元上的最优输出值。

划分输入空间采用启发式的方法：选择第j个变量x(j)和它的取值s作为切分变量（splitting variable）和切分点（splitting point），由此定义了两个区域：

R1(j, s) = {x | x(j) ≤ s} 和R2(j, s) = {x | x(j) > s}

单元Rm上的cm的最优值m是Rm上的所有输入实例xi对应的输出yi的均值：

求解以下式子

对固定输入变量j可以找到最优切分点s，其中和。通过遍历所有输入变量，可以找到最优的切分变量j，构成（j，s）对，将输入空间划分为两个区域。

**算法流程（最小二乘回归树生成算法）：**

输入：训练数据集D；

输出：回归树f(x)。

在训练数据集所在输入空间，递归地将每个区域划分为两个子区域并决定每个子区域的输出值，构建二叉决策树：

1）遍历变量j，对固定切分变量j求解

扫描切分点s，选择最优切分变量j与切分点s，使上式达到最小值的(j, s)；

2）用选定的(j，s)对划分区域并决定相应的输出值：

R1(j, s) = {x | x(j) ≤ s} 和R2(j, s) = {x | x(j) > s}

，x∈Rm，m=1，2

3）继续对两个子区域调用步骤1）、2），直至满足停止条件；

4）将输入空间划分为M个区域R1，R2，…，RM，生成决策树：

（2）分类树的生成

分类树用基尼指数选择最优特征，同时决定该特征的最优二值切分点。分类问题中，假设有K个类，样本点属于第k类的概率为pk，则概率分布的基尼指数定义为：

在特征A的条件下，集合D的基尼指数定义为：

基尼指数Gini(D)表示集合D的不确定性，基尼指数Gini(D，A)表示经A=a分割后集合D的不确定性。基尼指数越大，样本集合的不确定性也就越大。基尼指数表示每次选择不同的概率。

**算法流程：**

输入：训练数据集D，停止计算的条件；

输出：CART决策树。

1）根据结点训练数据集D现有的每个特征A可能的每个取值a测试为“是”或“否”将数据集D切分成D1和D2两个部分，并计算A=a时的基尼指数。

2）在所有可能的特征A以及它们所有的可能切分点a中，选择基尼指数最小的特征及其对应的切分点作为最优特征与最优切分点，将现结点依特征分配到两个子结点。

3）对两个子结点递归调用1）、2），直至满足停止条件；

4）生成CART决策树。

算法停止计算条件是结点中的样本个数小于预定阈值，或样本集的基尼指数小于预定阈值（样本基本属于同一类），或者没有更多特征。

**3.3.2 CART剪枝**

（1）从生成算法产生的决策树T0底端开始不断剪枝，直到T0的根结点，形成一个子树序列{ T0，T1，…，Tn}；

可以用递归的方法对树进行剪枝。将α从0增加到正无穷大，产生一系列区间α∈[αi，αi+1)，i=1，2，…，n，剪枝得到的子树序列对应着区间[αi，αi+1)的最优子树序列{ T0，T1，…，Tn}：

从整体树T0开始剪枝，对T0的任意内部结点t，以t为单结点树的损失函数为

Cα(t) = C(t) + α

以t为根结点的子树Tt的损失函数是

Cα(Tt) = C(Tt) + α|Tt|

当α=0或α充分小时，有不等式Cα(Tt) < Cα(t)；当α增大时，在某一α有Cα(Tt) = Cα(t)；当α再增大时，不等式变成Cα(Tt) > Cα(t)。当时，Tt与t有相同的损失函数值，而t的结点少，因此t比Tt更可取，对Tt进行剪枝。

因此，对T0中每一个内部结点t，计算，g(t)表示剪枝后整体损失函数减少的程度。在T0中剪去g(t)最小的Tt，将得到子树作为T1，同时将最小的g(t)设为α1。T1为区间[α1，α2)的最优子树。

（2）通过交叉验证法在独立的验证数据集上对子树序列进行测试，从中选择最优子树。

**CART剪枝算法：**

输入：CART算法生成的决策树T0；

输出：最优决策树Tα。

1）设k = 0，T = T0；

2）设α = +∞；

3）自下而上地对各内部结点t计算C(Tt)，|Tt|以及

α=min(α，g(t))

其中Tt表示以t为根结点的子树，C(Tt)是对训练数据的预测误差，|Tt|是Tt的叶结点个数。

4）对g(t) = α的内部结点t进行剪枝，并对叶结点t以多数表决法决定其类，得到树T；

5）设k=k+1，αk=α，Tk=T；

6）如果Tk不是由根结点及两个叶结点构成的树，回到3）；否则令Tk=Tn；

7）采用交叉验证法在子树序列中T0，T1，…，Tn选取最优子树Tα。