## 决策树（弱分类器）

介绍

1. 分治思想：把数据集分成两组
2. 不同数据点被完美区分了么
3. 不足重复之前两部
4. 满足收工

核心概念：

信息**熵**是代表随机变量的复杂度（不确定度），**条件熵**代表在某一个条件下，随机变量的复杂度（不确定度）。

而**信息增益**恰好是：信息熵-条件熵。换句话说，信息增益代表了在一个条件下，信息复杂度（不确定性）减少的程度。那么我们现在也很好理解了。

在决策树算法中，我们的关键就是每次选择一个特征，特征有多个，那么到底按照什么标准来选择哪一个特征。这个问题就可以用信息增益来度量。如果选择一个特征后，信息增益最大（信息不确定性减少的程度最大），那么我们就选取这个特征。

### 熵

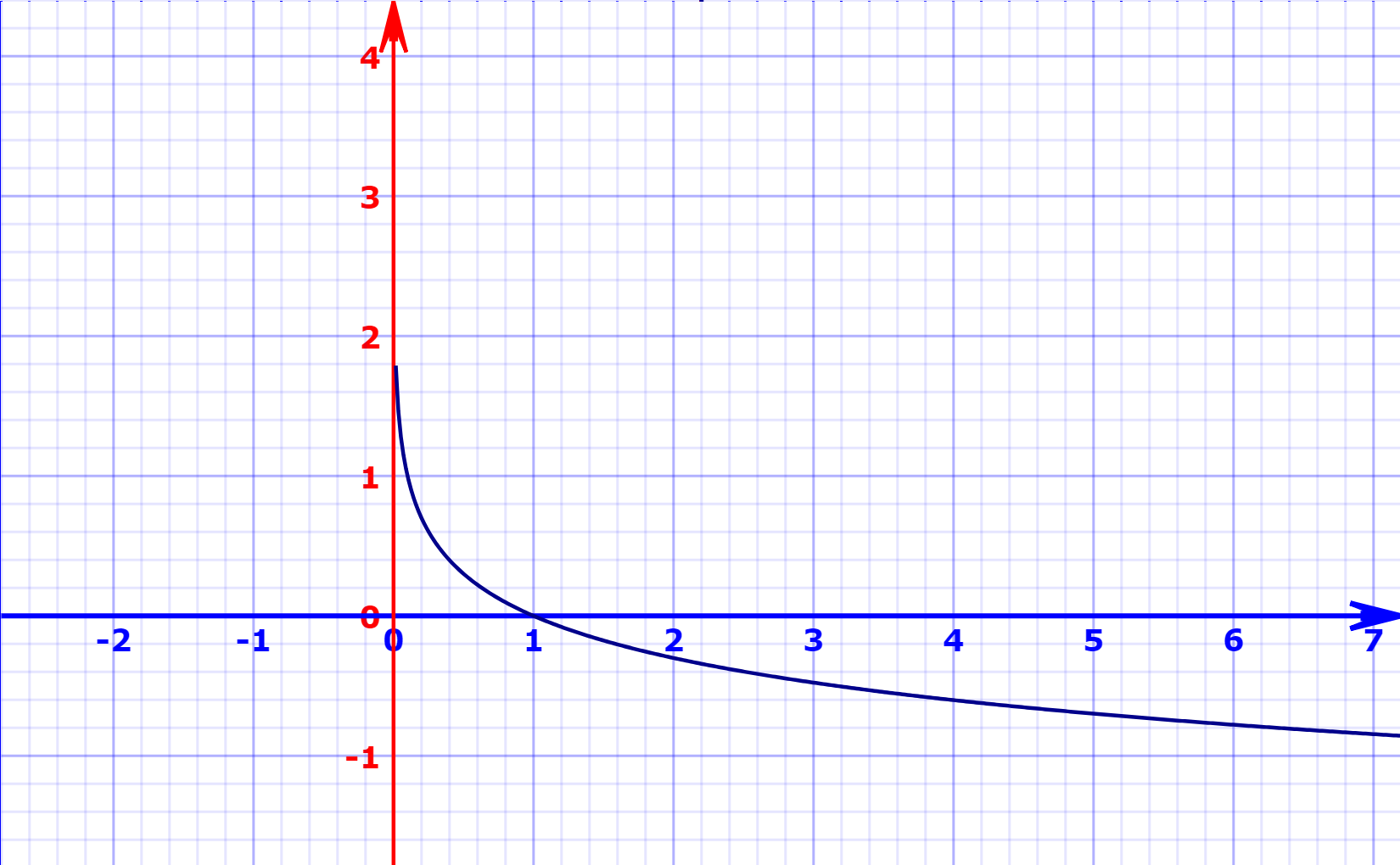
想要知道用那个条件（variable）区分数据，则要引入熵的概念

熵源于信息论，所以

1. 先说自信息

信息：

所绘制图形如下



从图中即可以看出，p(x)越接近于1，信息量越小，所以若果说概率p(x)是对确定性的度量，那么信息是对不确定性的度量。

1. 然后说熵（信息熵）

对比期望公式

熵就是对很多自信息求了期望。

熵是对平均不确定性的度量，本质是信息量的期望。（看公式可以看出是对不确定性求了期望，所以是平均）

计算示例：

比如一个分类分成了3个是，3个否(log底数2,10都行算出值一样)

1是高熵的，最差的分类，信息量是最低的。

### 信息增益

在决策树算法的学习过程中，信息增益是特征选择的一个重要指标，它定义为一个特征能够为分类系统带来多少信息，带来的信息越多，说明该特征越重要，相应的信息增益也就越大。信息增益也就是信息论中的平均互信息。信息增益公式如下：

也可以写作：

推导可详见（8）最大熵模型与EM算法23页

所以通俗的讲：信息增益=熵-条件熵。

1. 先说条件熵

读公式：Y条件下X的熵等于x事件发生的前提下，Y的信息熵再求期望。条件熵就是熵的熵。

计算示例：



12组数，可以求得随机变量X（嫁与不嫁）的信息熵为：嫁的个数为6个，占1/2，那么信息熵为-1/2log1/2-1/2log1/2 = -log1/2=0.301

现在引入另一个变量身高。假如我知道了一个男生的身高信息。身高有三个可能的取值{矮，中，高}

矮包括{1,2,3,5,6,11,12}，嫁的个数为1个，不嫁的个数为6个

中包括{8,9} ，嫁的个数为2个，不嫁的个数为0个

高包括{4,7,10}，嫁的个数为3个，不嫁的个数为0个

套用条件熵公式：

H(Y|X = 矮) = -1/7log1/7-6/7log6/7=0.178

H(Y|X=中) = -2log1-0 = 0

H(Y|X=高） = -3log1-0=0

p(X = 矮) = 7/12,

p(X =中) = 2/12,

p(X=高) = 3/12

则可以得出条件熵为：

7/12\*0.178+2/12\*0+3/12\*0 = 0.103

1. 再说信息增益

信息增益=熵-条件熵

实力计算：

信息增益=0.301-0.103=0.198

我们可以知道，本来如果我对一个男生什么都不知道的话，作为他的女朋友决定是否嫁给他的不确定性有0.301这么大。当我们知道男朋友的身高信息后，不确定度减少了0.198.也就是说，身高这个特征对于我们广大女生同学来说，决定嫁不嫁给自己的男朋友是很重要的。至少我们知道了身高特征后，我们原来没有底的心里（0.301）已经明朗一半多了，减少0.198了（大于原来的一半了）

### 充分自由生长

#### ID3算法

1. 算法流程：
2. 对当前样本集合，计算所有属性的信息增益；
3. 选择信息增益最大的属性作为测试属性，把测试属性取值相同的样本划为同一个子样本集；
4. 若子样本集的类别属性只含有单个属性，则分支为叶子节点，判断其属性值并标上相应的符号，然后返回调用出；
5. 否则对子样本递归调用本算法。
6. 决策树停止的条件：

如果发生以下的情况，决策树将停止分割

1. 1.改群数据的每一笔数据已经归类到每一类数据中，即数据已经不能继续在分。
2. 2.该群数据已经找不到新的属性进行节点分割
3. 3.该群数据没有任何未处理的数据
4. 决策树的bug-过拟合：

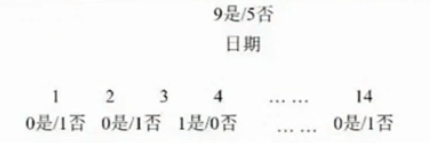
永远可以吧N个数据分成100%纯洁的N组

如果在【日期】这个条件下做分裂，就废了

#### C4.5算法

C4.5与ID3的区别：用信息增益比替换了信息增益。

因为信息增益存在缺陷，比如用【日期】分类，肯定可以分为非常纯洁的N组



1. 信息增益率

=

D是标签，A是特征。

信息增益率在信息增益基础上增加了一个惩罚系数，。即，特征**A本身的信息熵**的倒数。可以很好地抑制过细属性分类。

#### CART算法

基尼指数

基尼系数越高，两级分化越大。

基尼系数其实是信息增益率的变异。

### 修剪枝叶（CCP）

1. 没必要的分裂不要

在每次分枝前，做significant test，如果下分项怎样都是100%，（举极端例子比如说下分项一个是一个否，不管哪个都是100%，没必要分，返回重做），可以设置阈值75%这样。

1. 剪枝

决策树为什么(WHY)要剪枝？原因是避免决策树过拟合(Overfitting)样本。前面的算法生成的决策树非常详细并且庞大，每个属性都被详细地加以考虑，决策树的树叶节点所覆盖的训练样本都是“纯”的。因此用这个决策树来对训练样本进行分类的话，你会发现对于训练样本而言，这个树表现完好，误差率极低且能够正确得对训练样本集中的样本进行分类。训练样本中的错误数据也会被决策树学习，成为决策树的部分，但是对于测试数据的表现就没有想象的那么好，或者极差，这就是所谓的过拟合(Overfitting)问题。Quinlan教授试验，在数据集中，过拟合的决策树的错误率比经过简化的决策树的错误率要高。

预剪枝（PrePrune）判断停止树生长的方法可以归纳为以下几种：

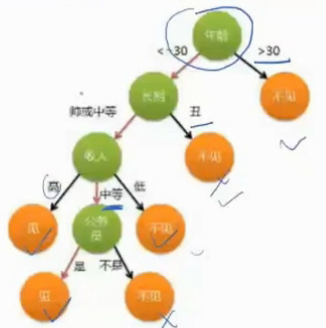
* 1. 达到最大数深度(Maximum Tree Depth)；
  2. 最优划分(Split)标增益小于某一个阈值也可停止树的生长；
  3. 到达此结点的实例具有相同的特征向量，而不必一定属于同一类，也可停止生长。
  4. 计算每次生长对系统性能的增益，如果这个增益值小于某个阈值则不进行生长。如果在最好情况下的生长增益都小于阈值，即使有些叶子结点的实例不属于同一类，也停止树的增长。

后剪枝（PostPrune）主要有以下几种方法：

1. 错误率降低剪枝 Reduced-Error Pruning(REP）
2. 悲观剪枝 Pessimistic Error Pruning(PEP）
3. 代价复杂度剪枝Cost-Complexity Pruning(CCP)CART用这种剪枝

#### 成本复杂度

假设已经通过ID3/C4.5/CART等算法生成了一充分自由生长的决策树Tmax，我们用T代表它的子树，用表示该子树中的**叶子节点数（下图中橘黄色的）**以衡量子树的复杂度，用R(T)表示T的误判成本（预测误差，平均误差/基尼指数），并引入一个α作为复杂参数。



所以，在复杂参数为α时，T的成本复杂测量度：

注意和是有咬合关系的

从公式可以看出，是误判成本和复杂参数的线性组合。当决策树自由生长时，误判成本可以很低，但此时的树枝繁叶茂，也大。单追求误判成本最小化必然产生过大的决策树。

复杂参数α可以理解为**每增加一个叶子节点而带来的复杂成本**。

实际上实在误判成本上增加了一个惩罚因子，加入考虑了复杂成本。

剪枝就是希望在误判成本和复杂成本之间追求一个平衡，希望得到想、较小的成本复杂测度。

#### 子树序列生成

，对于每个α，我们都可以找到使得达到最小子树T。当α很小的时候，**每增加一个叶子节点而带来的复杂成本**也很小，所以为了追求最小化，子树T会比较茂盛；随着α增大，**每增加一个叶子节点而带来的复杂成本**也加大，所以为了追求最小化，子树T会凋零。最极端的情况是α足够大时，最后T只剩下根结点。

基于以上，我们要做的是找到一种有效的算法，高效的寻找这一系列子树。

1. **正式开始子树序列生成：**
2. 子树T1生成

子树序列生成的起点是T1。由于非叶子节点t与其子树tL和子树tR存在关系：

即，t节点的误判成本大于等于其下两个子树的误判成本之和。

所以如果是=等号的时候，不分裂会更简洁的，此时就可以减掉这两个子树，子树tL和子树tR。按照这个条件，先把Tmax修剪一遍，就是子树T1。这里与成本复杂度无关。

1. 其他子树生成

剪枝实际上在非子叶节点t和以t为根节点的子树之间进行选择。

1. 剪枝前的状态：有个叶子节点，误差成本是；

以t为根节点的子树，其成本复杂测量度：

1. 剪枝后的状态：只有本身一个叶子节点，误差成本是

对于t本身，其成本复杂测度：

如果=，那么基于简洁性，t的枝干应该剪掉，此时就有：

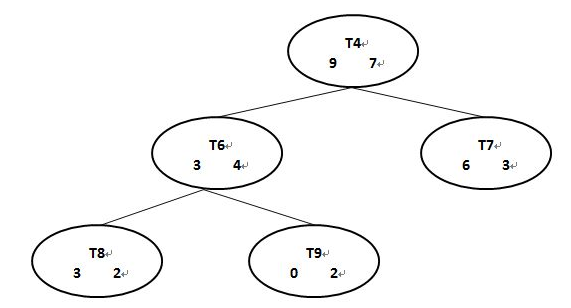
——节点t上的数据占所有数据的比例\*节点t的误差率

——子树Tt的误差成本=子树Tt上所有**叶子节点**的误差成本之和；

理论上讲，>，所以α满足大于0的条件的。于是，让α从0开始增加，依次将满足以上条件的枝干剪掉，并且记录对应的复杂参数α。（此处可以想到每个子树并不是针对某一个α值，而是一个区间）

至此，我们得到了子树序列T1> T2> T3…>{t1}，究竟用哪个子树作为最终的模型，还需要利用验证数据集来进行选择。

1. **示例计算：**



#### 最佳子树选择

1. 选择规则：
2. 预测概率：

某个观测的因变量为j，但模型将其判别为i的概率。

——验证集中j类别的观测个数；

——某一个子树将类别为j的误判为i的个数；

1. 类别的误判成本：

用表示将属于j的个体判别为I的误判成本，类别的误判成本为

表示类别为j的观测的平均误差成本。

1. 模型的误判成本：

用P表示类别的先验概率，整个模型的误判成本为：

1. 独立样本验证：

假设数据训练集为L，将L拆分为两个独立样本L1和L2，根据以上思路，用L1得到一系列子树，然后用其中每一个子树预测L2，由于L2中因变量已知，所以误判成本可以按以上规则计算出，每个子树对应一个误判成本，选误判成本最低的作为最佳子树。

1. 交叉验证法：

决策树究极进化

bagging

random forest

boosting

GBDT

XGBoost