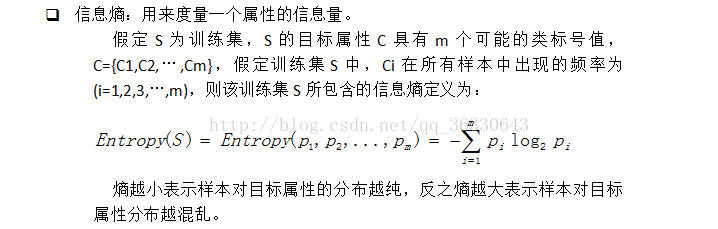
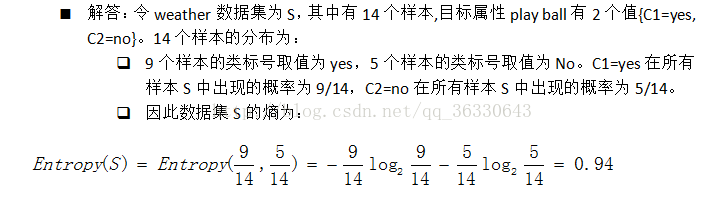
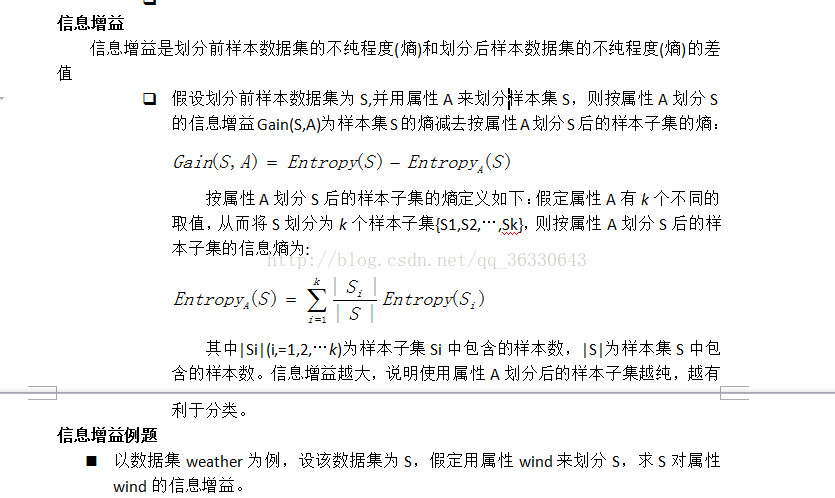
**1、C4.5决策树**

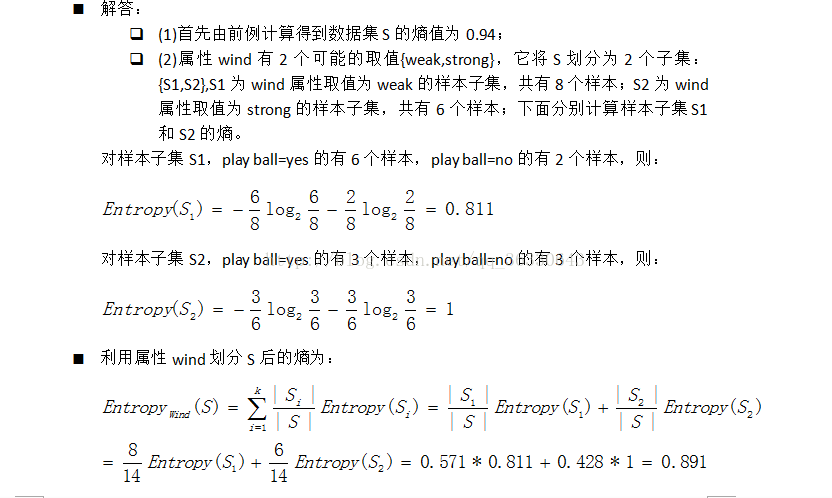
C4.5决策树的对ID3决策树的改进。





属性信息增益A的信息增益是样本集的熵减去按属性A划分S**样本子集的熵。**样本子集是指该属性的取值个数。样本子集的熵是所有样本子集的熵按仰恩数目的加权和。





样本增益是0.94-0.891。

最后按照信息增益最大的原则选Outlook为根节点。对子节点依次划分。

**ID3**优点是理论清晰、方法简单、学习能力较强，但也存在一些缺点：

（1）只能处理分类属性的数据，不能处理连续的数据；

（2）划分过程会由于子集规模过小而造成统计特征不充分而停止；

（3）ID3算法在选择根节点和各内部节点中的分支属性时，采用信息增益作为评价标准。信息增益的缺点是倾向于选择取值较多的属性，在有些情况下这类属性可能不会提供太多有价值的信息。

**C4.5算法**是用于生成决策树的一种经典算法，是ID3算法的一种延伸和优化。C4.5算法对ID3算法主要做了一下几点改进：

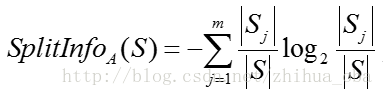
  （1）通过**信息增益率**选择分裂属性，克服了ID3算法中通过信息增益倾向于选择拥有多个属性值的属性作为分裂属性的不足；

  （2）能够处理**离散型和连续型**的属性类型，即将连续型的属性进行离散化处理；

  （3）构造决策树之后进行**剪枝**操作；

  （4）能够处理具有**缺失属性值**的训练数据。

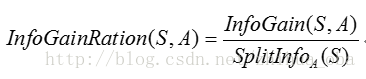
分裂属性选择的评判标准是决策树算法之间的根本区别。区别于ID3算法通过信息增益选择分裂属性，C4.5算法通过信息增益率选择分裂属性。   
  属性A的“分裂信息”(split information)：



其中，训练数据集S通过属性A的属性值划分为m个子数据集，|Sj||Sj| 表示第j个子数据集中样本数量，|S||S| 表示划分之前数据集中样本总数量。   
  通过属性A分裂之后样本集的信息增益：

这里写图片描述

通过属性A分裂之后样本集的信息增益率：



通过C4.5算法构造决策树时，信息增益率最大的属性即为当前节点的分裂属性，随着递归计算，被计算的属性的信息增益率会变得越来越小，到后期则选择相对比较大的信息增益率的属性作为分裂属性。

**连续型属性的离散化处理**

当属性类型为连续型，则需要对数据进行离散化处理。关键在于确定分割**阈值**。C4.5算法针对连续属性的离散化处理，核心思想：将属性A的N个属性值按照升序排列；通过二分法将属性A的所有属性值分成两部分（共有N-1种划分方法，二分的阈值为相邻两个属性值的中间值）；计算每种划分方法对应的信息增益，选取信息增益最大的划分方法的阈值作为属性A二分的阈值。

（1）将节点Node上的所有数据样本按照连续型属性A的具体取值，**由小到大**进行排列，得到属性A的属性值取值序列(xA1,...,xAN)(x1A,...,xNA)。

（2）在序列(xA1,...,xAN)(x1A,...,xNA)中共有N-1种二分方法，即共产生**N-1个分隔阈值**。对于第i种二分方法，其二分阈值θi=xAi+xAi+12θi=xiA+xi+1A2。它将该节点上的数据集划分为2个子数据集(xA1,...,xAi)(x1A,...,xiA)(xAi+1,...,xAN)(xi+1A,...,xNA)。计算此种二分结果下的信息增益。

（3）分别计算N-1种二分结果下的信息增益，选取信息增益最大的二分结果作为对属性A的划分结果，并记录此时的二分阈值。

**剪枝——PEP(Pessimistic Error Pruning)剪枝法**

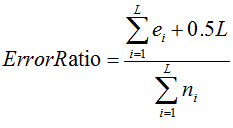
剪枝方法分为预剪枝和后剪枝两大类。预剪枝是在构建决策树的过程中，提前终止决策树的生长，从而避免过多的节点产生。预剪枝方法虽然简单但实用性不强，因为很难精确的判断何时终止树的生长。

**后剪枝**是在决策树构建完成之后，对那些置信度不达标的节点子树用叶子结点代替，该叶子结点的类标号用该节点子树中频率最高的类标记。后剪枝方法又分为两种，一类是把训练数据集分成树的生长集和剪枝集；另一类算法则是使用同一数据集进行决策树生长和剪枝。常见的后剪枝方法有CCP(Cost Complexity Pruning)、REP(Reduced Error Pruning)、PEP(Pessimistic Error Pruning)、MEP(Minimum Error Pruning)。

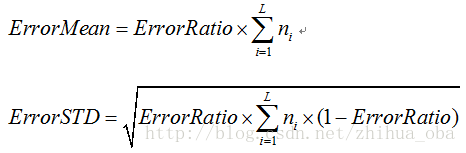
C4.5算法采用PEP(Pessimistic Error Pruning)剪枝法。PEP剪枝法由Quinlan提出，是一种自上而下的剪枝法，根据剪枝前后的错误率来判定是否进行子树的修剪，因此不需要单独的剪枝数据集。接下来详细介绍PEP(Pessimistic Error Pruning)剪枝法。

对于一个叶子节点，它覆盖了n个样本，其中有e个错误，那么该叶子节点的错误率为(e+0.5)/n(e+0.5)/n，其中0.5为惩罚因子（惩罚因子一般取值为0.5）。

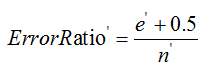
对于一棵子树，它有L个叶子节点，那么该子树的误判率为：



其中，eiei表示子树第i个叶子节点错误分类的样本数量，nini表示表示子树第i个叶子节点中样本的总数量。   
  假设一棵子树错误分类一个样本取值为1，正确分类一个样本取值为0，那么子树的误判次数可以认为是一个伯努利分布，因此可以得到该子树误判次数的均值和标准差：



把子树替换成叶子节点后，该叶子节点的误判率为



同时，该叶子结点的误判次数也是一个伯努利分布，因此该叶子节点误判次数的均值为：

这里写图片描述

剪枝的条件为：

这里写图片描述

满足剪枝条件时，则将所得叶子节点替换该子树，即为剪枝操作。

**5. 缺失属性值的处理**

  训练样本集中有可能会出现一些样本缺失了一些属性值，待分类样本中也会出现这样的情况。当遇到这样的样本集时该如何处理呢？含有缺失属性的样本集会一般会导致三个问题：

  （１）在构建决策树时，每一个分裂属性的选取是由训练样本集中所有属性的信息増益率来决定的。而在此阶段，如果训练样本集中有些样本缺少一部分属性，此时该如何**计算该属性的信息増益率；**

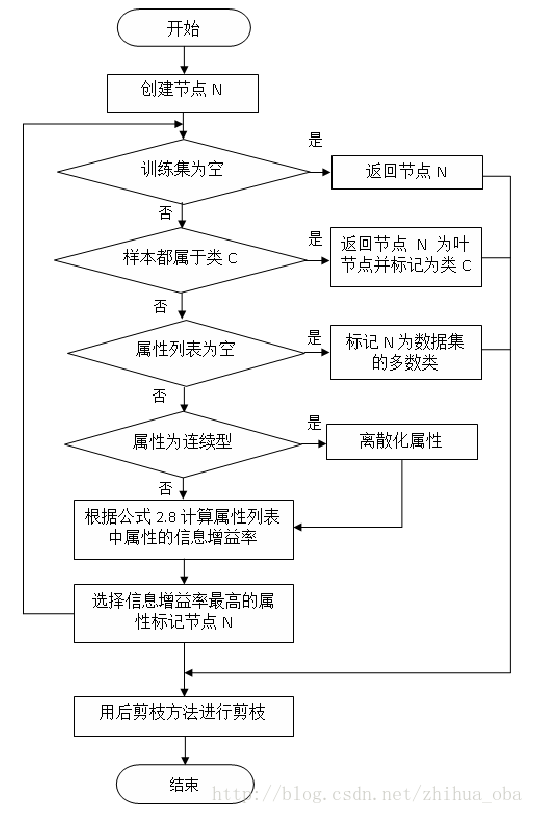
  （２）当已经选择某属性作为分裂属性时，样本集应该根据该属性的值来进行分支，但对于那些**该属性的值为未知的样本，应该将它分支到哪一棵子树上；**

  （３）在决策树已经构建完成后，如果待分类样本中有些属性值缺失，则该样本的分类过程如何进行。

面对问题一，在**计算各属性的信息増益率**时，若某些样本的属性值未知，那么可以这样处理：计算某属性的信息増益率时忽略掉缺失了此属性的样本；或者通过此属性的样本中出现频率最高的属性值，賦值给缺失了此属性的样本。

面对问题二，假设属性Ａ已被选择作为决策树中的一个分支节点，**在对样本集进行分支的时候，对于那些属性Ａ的值未知的样本**，可以送样处理：不处理那些属性Ａ未知的样本，即简单的忽略它们；或者根据属性Ａ的其他样本的取值，来对未知样本进行赋值；或者为缺失属性Ａ的样本单独创建一个分支，不过这种方式得到的决策树模型结点数显然要増加，使模型更加复杂了。

面对问题三，根据**己经生成的决策树模型，对一个待分类的样本进行分类时**，若此样本的属性Ａ的值未知，可以这样处理：待分类样本在到达属性Ａ的分支结点时即可结束分类过程，此样本所属类别为属性Ａ的子树中概率最大的类别；或者把待分类样本的属性Ａ赋予一个最常见的值，然后继续分类过程。



7. C4.5算法优缺点分析

优点：

（1）通过信息增益率选择分裂属性，克服了ID3算法中通过信息增益倾向于选择拥有多个属性值的属性作为分裂属性的不足；

（2）能够处理离散型和连续型的属性类型，即将连续型的属性进行离散化处理；

（3）构造决策树之后进行剪枝操作；

（4）能够处理具有缺失属性值的训练数据。

缺点：

（1）算法的**计算效率较低**，特别是针对含有连续属性值的训练样本时表现的尤为突出。

（2）算法在选择分裂属性时**没有考虑到条件属性间的相关性**，只计算数据集中每一个条件属性与决策属性之间的期望信息，有可能影响到属性选择的正确性。

**2、CART决策树Classification and Regression Tree**

CART是Classification and Regression Tree分类回归树的简称，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。主要由特征选择，树的生成和剪枝三部分组成。它主要用来处理分类和回归问题，下面对分别对其进行介绍。

（1）CART既能是分类树，又能是分类树；

（2）当CART是分类树时，采用GINI值作为节点分裂的依据；当CART是回归树时，采用样本的最小方差作为节点分裂的依据；

（3）CART是一棵二叉树。

|  |  |  |  |
| --- | --- | --- | --- |
| 看电视时间 | 婚姻情况 | 职业 | 年龄 |
| 3 | 未婚 | 学生 | 12 |
| 4 | 未婚 | 学生 | 18 |
| 2 | 已婚 | 老师 | 26 |
| 5 | 已婚 | 上班族 | 47 |
| 2.5 | 已婚 | 上班族 | 36 |
| 3.5 | 未婚 | 老师 | 29 |
| 4 | 已婚 | 学生 | 21 |

**分类树？回归树？**

分类树的作用是通过一个对象的特征来预测该对象所属的类别，而回归树的目的是根据一个对象的信息预测该对象的属性，并以数值表示。

CART既能是分类树，又能是决策树，如上表所示，如果我们想预测一个人是否已婚，那么构建的CART将是分类树；如果想预测一个人的年龄，那么构建的将是回归树。

一棵分类树，其叶子节点的输出结果为一个实际的类别，在这个例子里是婚姻的情况（已婚或者未婚），选择叶子节点中数量占比最大的类别作为输出的类别；

一棵回归树，预测用户的实际年龄，是一个具体的输出值。怎样得到这个输出值？一般情况下选择使用中值、平均值或者众数进行表示。

**CART如何选择分裂的属性？**

分裂的目的是为了能够让数据变纯，使决策树输出的结果更接近真实值。那么CART是如何评价节点的纯度呢？如果是分类树，CART采用GINI值衡量节点纯度；如果是回归树，采用样本方差衡量节点纯度。节点越不纯，节点分类或者预测的效果就越差。对节点内的样本计算。

GINI值的计算公式：

https://images2015.cnblogs.com/blog/833682/201601/833682-20160116140130616-1716397508.png

节点越不纯，GINI值越大。

https://images2015.cnblogs.com/blog/833682/201601/833682-20160116140203741-208915813.png

方差越大，表示该节点的数据越分散，预测的效果就越差。

因此，无论是分类树还是回归树，CART都要选择使子节点的GINI值或者回归方差最小的属性作为分裂的方案。即最小化（分类树）：

https://images2015.cnblogs.com/blog/833682/201601/833682-20160116140229210-576757285.png

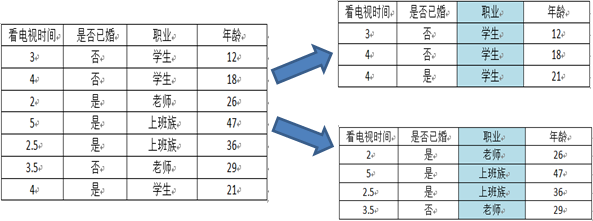
或者（回归树）：

https://images2015.cnblogs.com/blog/833682/201601/833682-20160116140250444-1000189848.png

**CART如何分裂成一棵二叉树？**

    节点的分裂分为两种情况，连续型的数据和离散型的数据。

    CART对连续型属性的处理与C4.5差不多，通过最小化分裂后的GINI值或者样本方差寻找最优分割点，将节点一分为二，在这里不再叙述，详细请看**C4.5**。



   对于离散型属性，理论上有多少个离散值就应该分裂成多少个节点。但CART是一棵二叉树，每一次分裂只会产生两个节点，怎么办呢？很简单，只要将其中一个离散值独立作为一个节点，其他的离散值生成另外一个节点即可。这种分裂方案有多少个离散值就有多少种划分的方法，举一个简单的例子：如果某离散属性一个有三个离散值X，Y，Z，则该属性的分裂方法有{X}、{Y，Z}，{Y}、{X，Z}，{Z}、{X，Y}，分别计算每种划分方法的基尼值或者样本方差确定最优的方法。

   以属性“职业”为例，一共有三个离散值，“学生”、“老师”、“上班族”。该属性有三种划分的方案，分别为{“学生”}、{“老师”、“上班族”}，{“老师”}、{“学生”、“上班族”}，{“上班族”}、{“学生”、“老师”}，分别计算三种划分方案的子节点GINI值或者样本方差，选择最优的划分方法，如下图所示：

**第一种划分方法：{“学生”}、{“老师”、“上班族”}**

预测是否已婚（分类）：

https://images2015.cnblogs.com/blog/833682/201601/833682-20160116140514319-212672665.png

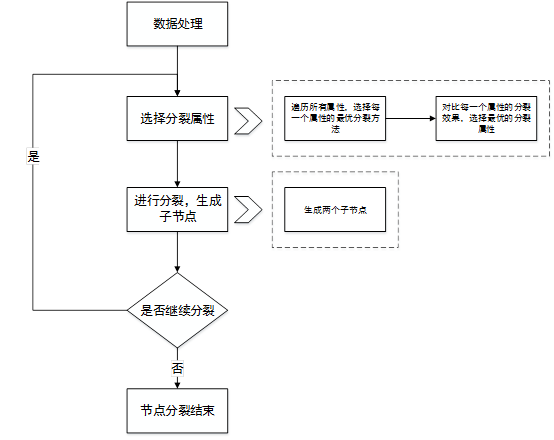
预测年龄（回归）：

https://images2015.cnblogs.com/blog/833682/201601/833682-20160116140542975-1587608963.png

**第二种划分方法：{“老师”}、{“学生”、“上班族”}**

https://images2015.cnblogs.com/blog/833682/201601/833682-20160116140635835-1539341444.png

https://images2015.cnblogs.com/blog/833682/201601/833682-20160116140642710-1920641216.png



**总结：**

（1）CART是一棵二叉树，每一次分裂会产生两个子节点，对于连续性的数据，直接采用与C4.5相似的处理方法，对于离散型数据，选择最优的两种离散值组合方法。

（2）CART既能是分类数，又能是二叉树。如果是分类树，将选择能够最小化分裂后节点GINI值的分裂属性；如果是回归树，选择能够最小化两个节点样本方差的分裂属性。

（3）CART跟C4.5一样，需要进行剪枝，采用CCP（代价复杂度的剪枝方法）。