本周学习：决策树的其他生成算法（C4.5）

先回顾一下之前学的ID3算法

ID3

这是一个采用自顶向下的贪婪的搜索遍历可能的决策树空间算法（通过自顶向下的方法来构造决策树进行学习）：

首先，挑选一个属性，作为决策树的根结点。挑选规则是分类性能最好的属性作为根结点（统计学得到信息熵增[第九次报告]最大的结点）。然后根结点属性的每一个值都会产生一个分支（对应着不同的样本集），我们需要把所有的训练样本都分到对应的分支下。然后，每个分支下再选择出一个叶子的结点，规则与根结点一致。重复上述过程，用每个分支结点关联的训练样本来选取在该点被测试的最佳属性，直到我们可以把每一个训练样本分类为止。

算法终止条件：

1）所有的属性已经被路径包括了；

2）这个结点关联的所有训练样本都具有相同的目标属性值（即熵为0）。

算法特点：

该算法形成了对合格决策树的贪婪搜索（greedy search），该算法没有回溯的过程。换句话说，当选择好了根结点或者是叶子的结点之后，它就不会再回头考虑之前的选择

ID3算法的优缺点

1）ID3算法中的假设空间包含了所有的决策树，它是关于现有属性的有限离散值函数的一个完整空间。所以该算法避免了假设空间可能不包含目标函数的风险；

2）当遍历决策树空间时，ID3仅仅考虑了单一的当前假设，故该算法失去了表示所有一致假设所带来的优势，即不能找出所有符合条件的决策树；

3）基本的ID3算法在搜索中不进行回溯。在无回溯的爬山搜索中，最常见的分线就是：收敛到局部最优的答案，而不是全局最优。这里我们可以增加回溯，如修剪决策树（C4.5）；

4）ID3算法在搜索的每一步都使用当前的所有训练样本，以统计为基础决定怎样精化当前的假设。因为使用到了所有样本的统计信息，所以对训练样本中的噪声不敏感，具有较好的健壮性。

ID3归纳偏置：较短的树比较长的树优先，信息增益高的属性更靠近根结点的树优先。

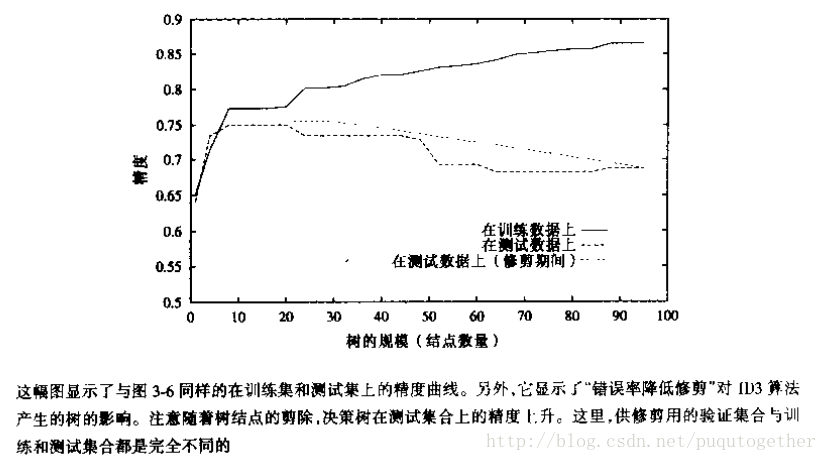
决策树中常见的问题

决策树学习过程中会经常遇到如下问题：确定决策树增长的深度；处理连续值的属性；选择一个适当的属性筛选度量标准；处理属性值不完整的训练样本；处理不同代价的属性；提高计算效率等等。那么C4.5就是在解决了如上这些问题之后得到的一种改进决策树算法。

相同点与不同点：同样属于贪心算法；分类决策的依据不同。

1）避免过度拟合数据

理想的决策树学习算法中，我们可以恰好的对训练样本完美的分类。但是，当数据中含有噪声，或者是训练样本的个数太少的时候，就会出现过度拟合的现象。换句话说，过度拟合想象就是存在另外一种假设，其在某个树的规模（深度）的时候错误率（分类效果）比最终求的假设大，但是整体来看，这样的另外一种假设错误率最小。这就是过度拟合。



两种方法解决过度拟合：

a、及早停止树的增长，在ID3算法完美分类训练数据之前就可以停止增长；

b、后修剪法，即允许树过度拟合数据，然后在得到的决策树上进行后续的修剪。

叶子裁剪

分析分类回归树的递归建树过程，不难发现它实质上存在着一个数据过度拟合问题。在决策树构造时，由于训练数据中的噪音或孤立点，许多分枝反映的是训练数据中的异常，使用这样的判定树对类别未知的数据进行分类，分类的准确性不高。因此试图检测和减去这样的分支，检测和减去这些分支的过程被称为树剪枝。树剪枝方法用于处理过分适应数据问题。通常，这种方法使用统计度量，减去最不可靠的分支，这将导致较快的分类，提高树独立于训练数据正确分类的能力。

决策树常用的剪枝常用的简直方法有两种：预剪枝(Pre-Pruning)和后剪枝(Post-Pruning)。

预剪枝

预剪枝是根据一些原则及早的停止树增长，如树的深度达到用户所要的深度、节点中样本个数少于用户指定个数、不纯度指标下降的最大幅度小于用户指定的幅度等。预剪枝的核心问题是如何事先指定树的最大深度，如果设置的最大深度不恰当，那么将会导致过于限制树的生长，使决策树的表达式规则趋于一般，不能更好地对新数据集进行分类和预测。除了事先限定决策树的最大深度之外，还有另外一个方法来实现预剪枝操作，那就是采用检验技术对当前结点对应的样本集合进行检验，如果该样本集合的样本数量已小于事先指定的最小允许值，那么停止该结点的继续生长，并将该结点变为叶子结点，否则可以继续扩展该结点。

规则后修剪法：

Step 1：从训练样本集中推导出决策树，增长决策树直到尽可能好地拟合训练样本，这里允许过度拟合的发生；

Step 2：将决策树转化为等价的规则集合，方法是为从根结点到叶子及诶但的每一条路径创建一条规则；

Step 3：通过删除任何能导致估计精度的前件来修剪）泛化 每一条规则；

step 4：按照修剪过的规则的估计精度对它们进行排序，并按照这样的顺序应用这些规则来分类后来的实例。

2）合并连续值属性

对于连续值属性，我们只需要把连续值属性采样称为离散的区间就可以了，之后构建决策树的步骤和前面一样。

对于连续分布的特征，其处理方法是：

先把连续属性转换为离散属性再进行处理。虽然本质上属性的取值是连续的，但对于有限的采样数据它是离散的，如果有N条样本，那么我们有N-1种离散化的方法：<=vj的分到左子树，>vj的分到右子树。计算这N-1种情况下最大的信息增益率。另外，对于连续属性先进行排序（升序），只有在决策属性（即分类发生了变化）发生改变的地方才需要切开，这可以显著减少运算量。经证明，在决定连续特征的分界点时采用增益这个指标（因为若采用增益率，splittedinfo影响分裂点信息度量准确性，若某分界点恰好将连续特征分成数目相等的两部分时其抑制作用最大），而选择属性的时候才使用增益率这个指标能选择出最佳分类特征。

在C4.5中，对连续属性的处理如下：

1.      对特征的取值进行升序排序

2.      两个特征取值之间的中点作为可能的分裂点，将数据集分成两部分，计算每个可 能的分裂点的信息增益（InforGain）。优化算法就是只计算分类属性发生改变的那 些特征取值。

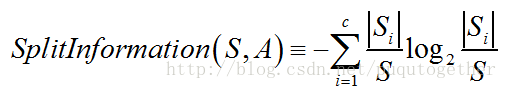
3.      选择修正后信息增益(InforGain)最大的分裂点作为该特征的最佳分裂点

4.      计算最佳分裂点的信息增益率（Gain Ratio）作为特征的Gain Ratio。注意，此处需对最佳分裂点的信息增益进行修正：减去log2(N-1)/|D|（N是连续特征的取值个数，D是训练数据数目，此修正的原因在于：当离散属性和连续属性并存时，C4.5算法倾向于选择连续特征做最佳树分裂点）

3）属性选择的其他度量标准

信息增益度量标准的一个缺陷就是：偏袒了具有较多值的属性。避免这个不足的一种方法就是用增益比率（gain ratio）作为度量标准，这也是C4.5算法中用来选择属性的度量标准。

相对于信息增益，增益比率增加了一个分裂信息（split information），它是用来惩罚具有较多值的属性的。分裂信息用来衡量属性分裂数据的广度和均匀性，其计算公式如下：



我们可以看出，分裂信息其实就是S（当前样本集）关于属性A的各值的熵。

那么增益比率就是：



（Gain（S，A）就是样本中分类中的熵增：样本的经验熵-A属性的信息熵）

（Splitinformation（S，A）就是样本集中A属性的不同值的-概率\*log概率和）

使用增益比率得到的选择属性的效果显然更好。

在ID3中用信息增益选择属性时偏向于选择分枝比较多的属性值，即取值多的属性，在C4.5中由于除以SplitInformation(S,A)=H(X)，可以削弱这种作用

4）处理缺少属性值的训练样本

树立缺少属性值的一种策略就是赋给它结点n的训练样本中该属性的最常见的值。或者是赋给它结点n的被分类为c(x)的训练样本中该属性的最常见的值。

5）处理不同代价的属性

引入代价敏感项（cost-sensitive term），这个是教研室中经常用到的technique。

参考资料：

[1].http://blog.csdn.net/gamer\_gyt/article/details/51242815

[2].http://blog.csdn.net/puqutogether/article/details/25455031