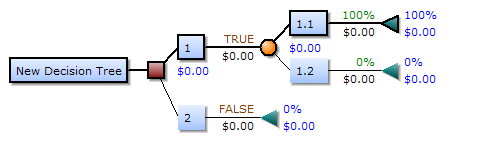
# 树形分类器

介绍两种树形分类算法的基本原理：决策树算法和随机森林算法。

## 决策树

**决策树**是一个预测模型；他代表的是对象属性与对象值之间的一种映射关系。树中每个节点表示某个对象，每个分叉路径则代表的某个可能的属性值，每个叶结点对应从根节点到该叶节点所经历的路径所表示的对象的值。从数据产生决策树的机器学习技术叫做**决策树学习**, 通俗说就是**决策树**。

一个决策树包含三种类型的节点： 1.决策节点——通常用矩形框来表式 2.机会节点——通常用圆圈来表式 3.终结点——通常用三角形来表示 

每个决策树都表述了一种树型结构，由它的分支来对该类型的对象依靠属性进行分类。每个决策树可以依靠对源数据库的分割进行数据测试。这个过程可以递归式的对树进行修剪。当不能再进行分割或一个单独的类可以被应用于某一分支时，递归过程就完成了。另外，**随机森林**分类器将许多决策树结合起来以提升分类的正确率。

### 决策树算法

#### 1. ID3算法

ID3算法是一个由Ross Quinlan发明的用于决策树的算法。这个算法便是建立在上述所介绍的奥卡姆剃刀的基础上：越是小型的决策树越优。该算法是一个启发式算法（不是总能得到最优解）。

汤姆.米歇尔《机器学习》中对ID3算法的描述：



**终止条件**

（简单来说就是每次构建节点（root）使得对当前集合的分类最佳，然后对当前节点条件判断产生的两个子集合再递归地进行该过程，构造子树。一种贪心风格的算法。）

信息增益的度量标准可以用**熵**（entropy）表示。例如布尔型分类的熵为：

http://hi.csdn.net/attachment/201201/8/0_1326017614WvVU.gif，

表示了样例集的纯度（purity）。

#### 2. C4.5算法

Quilan提出了C4.5算法，是ID3的一个改进算法。主要步骤：

a. 读取文件信息，统计数目

b. 建立决策树

* 如果样本集为空，则生成一个信息数目都为0的树节点返回
* 如果样本均为同一类别，则生成一个叶子节点返回
* 计算节点正负样本的数目
* 如果属性值只有那个类别的属性，则生成一个叶子节点，并赋值类型索引
* 如果以上都不是，则选择一个增益率最大的属性（连续属性要用增益率离散化），按那个属性的取值情况从新定义样本集和属性集，建造相关子树

c. 剪枝（采用悲观错误率估算）

d. 输出决策树

因此和ID3算法的主要区别在于：

* 用信息增益率来选择属性，克服了用信息增益选择属性时偏向选择取值多的属性的不足。
* 在树构造过程中进行剪枝；
* 能够完成对连续属性的离散化处理；
* 能够对不完整数据进行处理。

ref：<http://www.cnblogs.com/biyeymyhjob/archive/2012/07/23/2605208.html>

## 随机森林**(Random Forest)**

决策树算法有着很多良好的特性，比如说训练时间复杂度较低，预测的过程比较快速，模型容易展示等。但是同时， 单决策树又有一些不好的地方，比如说容易over-fitting，虽然有一些方法，如剪枝，但是还是不够的。

模型组合（比如说有Boosting，Bagging等）与决策树相关的算法比较多，这些算法最终的结果是生成N(可能会有几百棵以上）棵树，这样可以大大的减少单决策树带来的问题。虽然每一棵决策树都很简单（相对于C4.5这种单决策树来说），但它们组合起来很强大。模型组合+决策树相关的算法有两种比较基本的形式 - 随机森林与GBDT(Gradient Boost Decision Tree)，其他的比较新的模型组合+决策树的算法都是来自这两种算法的延伸。

### 随机森林算法

随机森林顾名思义，是用随机的方式建立一个很多个决策树组成的森林，每一棵决策树之间是没有关联的。当有一个新的输 入样本进入的时候，就让森林中的每一棵决策树分别进行判断，然后看哪一类被选择最多，就预测这个样本为那一类。

建立决策树的过程主要有两个关键点：采样与完全分裂。

**1）随机采样**：对输入的数据要进行行、列的采样。对于行采样，采用有放回的方式，也就是在采样得到的样本集合中，可能有重复的样本。假设输入样本为N个，那么采样的样本也为N个。这样在训练的时候，每一棵树的输入样本都不是全部的样本，使得相对不容易出现over-fitting。然后进行列采样，从M 个feature中，选择m个(m << M)。

**2）建立决策树**：之后就是对采样之后的数据使用完全分裂的方式建立出决策树（决策树的某一个叶子节点要么是无法继续分裂的，要么里面的所有样本的都是指向的同一个分类）。

由于两个随机采样的过程保证了随机性，所以就算不剪枝，也不会出现over-fitting。

随机森林有很多的优点：

* 在数据集上表现良好
* 在当前的很多数据集上，相对其他算法有着很大的优势
* 它能够处理很高维度（feature很多）的数据，并且不用做特征选择
* 在训练完后，它能够给出哪些feature比较重要
* 在创建随机森林的时候，对generlization error使用的是无偏估计
* 训练速度快
* 在训练过程中，能够检测到feature间的互相影响
* 容易做成并行化方法
* 实现比较简单

ref： <http://www.cnblogs.com/wentingtu/archive/2011/12/13/2286212.html>