决策树（Decision Tree）

作者：刘伟杰 日期：2015-11-27
参考：
[1]《统计学习方法》 李航 2012年3月第一版
[2]《机器学习实战》 Peter Harrington

1. 理论

1. 概述：

   决策树的内部节点表示一个特征或属性，叶子节点表示一个类别。输入一个新样本，从根节点开始按照节点说示的特征划分，直到划分到叶子节点，该叶子节点即为类别。

2. 关于熵的基础知识

   熵：定义式如下，衡量随机变量X的不确定性，也可以用于衡量一个集合的混乱度。

   $$H(X) = \sum(p_i log p_i)$$

   条件熵：H(Y|X)表示在已知随机变量X的条件下随机变量Y的不确定性，也可以理解为集合按照特征X划分以后的混乱度。

   $$H(Y|X) = \sum_{i=1}^{n} p_i H(Y|X = x_i)$$

   信息增益：表示在得知X的信息而使得Y的信息的不确定性减少的程度, 也就是按照特征X划分以后的混乱度。

   g(Y,X) = H(Y) - H(Y|X)

3. 关于决策树

   决策树用好的话是一个很好的分类算法，还可以归纳出一组划分规则。但是决策树对数据的预处理要求较高，最好能够提前进行特征降维，选取关键特征。如果决策树过拟合，还需要用剪枝算法进行剪枝。当然，除了用熵，还有一个衡量集合混乱度的指标是基尼不纯度(gini)

2. 算法

1. 学习算法

ID3算法

> 从根结点开始，对训练集计算所有特征的信息增益，选取信息增益最大的特征作为该节点的分类特征，

C4.5算法

> 使用信息增益划分存在偏向于选择取值较多特征的问题。C4.5与ID3算法相同，只是将信息增益换成信
>
> gr(Y,X)=g(Y,X)/H(X)

2. 剪枝算法

   确定一个函数作为剪枝的指标（例如选用代价函数），尝试将一组叶子节点退缩回到父节点，如果退回后的指标优于退回前，者进行退回。剪枝的算法和指标很多，这里不做讨论。

3. 实现

1. 我的实现：

   https://github.com/autoliuweijie/MachineLearning/tree/master/trees (https://github.com/autoliuweijie/MachineLearning/tree/master/trees)

2. scikit-learn:

   示例：

```
#Import Library
#Import other necessary libraries like pandas, numpy...
from sklearn import tree
#Assumed you have, X (predictor) and Y (target) for training data set and x_test(p
# Create tree object
model = tree.DecisionTreeClassifier(criterion='gini') # for classification, here y
# model = tree.DecisionTreeRegressor() for regression
# Train the model using the training sets and check score
model.fit(X, y)
model.score(X, y)
#Predict Output
predicted= model.predict(x_test)
```

4. 扩展

决策树还有一个同根的算法，称为回归树(CART)，可以用来做回归。决策树与回归树的界限不是很明显，有人经常把这两个概念混用。我喜欢把用来分类的树称为决策树，用于回归的树称做回归树。