文本分类算法之决策树.ID3实现_liuyongkanglinux的专栏-程序员宅基地_决策树 文本分类

技术标签： 算法 机器学习 ID3 决策树

文本分类，字面意思来说就是对文本进行分类。其实就是这样，文本分类是根据文本所具有的一些特征和属性来将其判别为事先确定的几个类别的过程。常用的分类方法有朴素贝叶斯方法，K近邻方法（KNN），支持向量机（SVM），决策树方法，神经网络方法等等。本文主要来介绍决策树的一种实现。

决策树是通过文本的一些属性来建立的一个模型。决策树采用自顶向下的方式，树中的每个结点都代表一个属性，该结点的每棵子树都代表这个属性的一个确定取值，叶结点上放着决策值。用决策树对目标属性进行判断时，将从树根出发，沿着已知的属性进入子树，知道找到叶结点，也就得到的预测结果。所以从跟结点到每一个叶结点的路径都是一条决策规则。

引用他人博文（①）中的一个例子：

通俗来说，决策树分类的思想类似于找对象。现想象一个女孩的母亲要给这个女孩介绍男朋友，于是有了下面的对话：

女儿：多大年纪了?

母亲：26。

女儿：长的帅不帅?

母亲：挺帅的。

女儿：收入高不?

母亲：不算很高，中等情况。

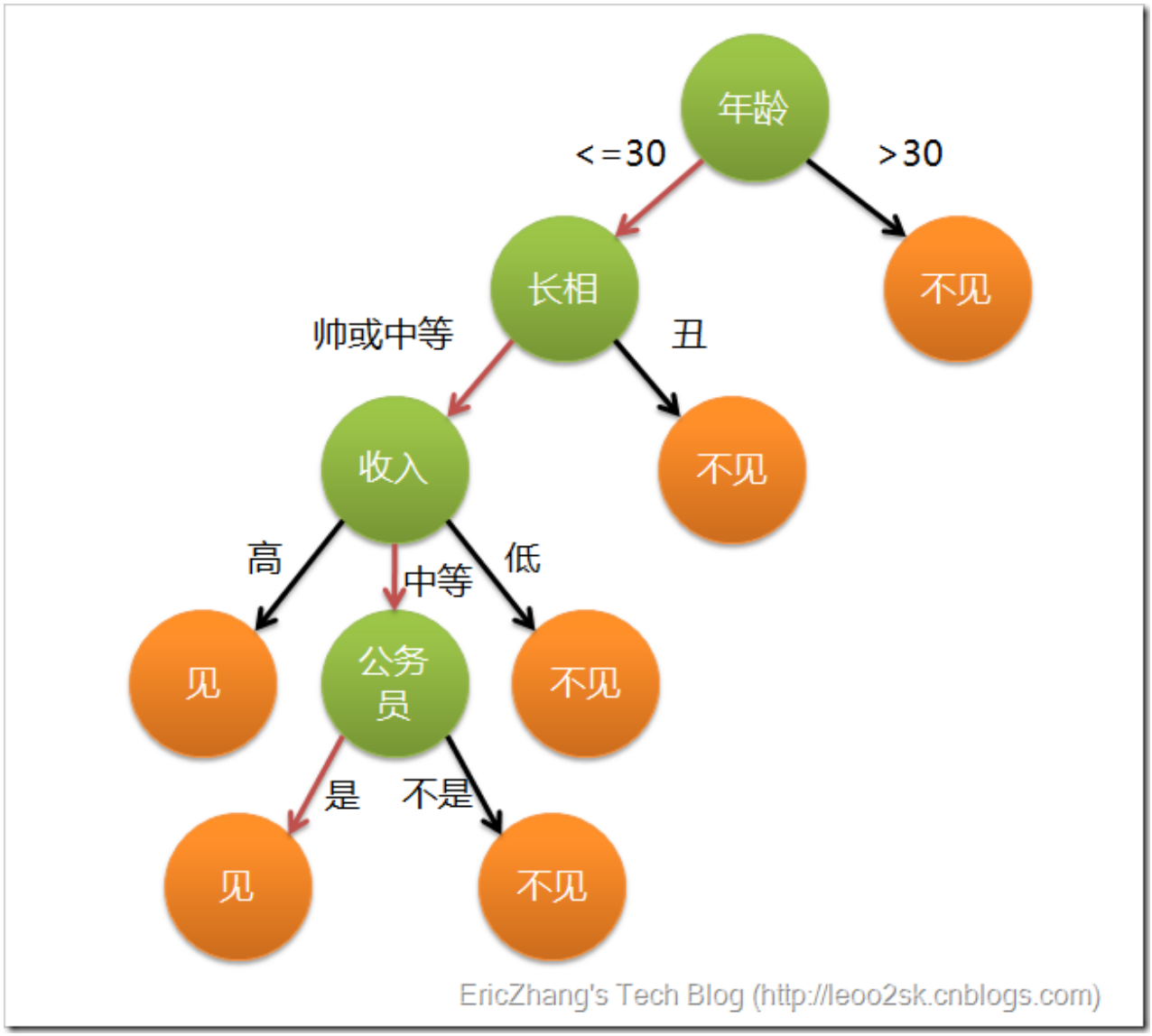
女儿：是公务员不?

母亲：是，在税务局上班呢。

女儿：那好，我去见见。

这个女孩的决策过程就是典型的分类树决策。相当于通过年龄、长相、收入和是否公务员对将男人分为两个类别：见和不见。假设这个女孩对男人的要求是：30岁以下、长相中等以上并且是高收入者或中等以上收入的公务员，那么这个可以用下图表示女孩的决策逻辑（声明：此决策树纯属为了写文章而YY的产物，没有任何根据，也不代表任何女孩的择偶倾向，请各位女同胞莫质问我^^）：

_



上图完整表达了这个女孩决定是否见一个约会对象的策略，其中绿色节点表示判断条件，橙色节点表示决策结果，箭头表示在一个判断条件在不同情况下的决策路径，图中红色箭头表示了上面例子中女孩的决策过程。

这幅图基本可以算是一颗决策树，说它"基本可以算"是因为图中的判定条件没有量化，如收入高中低等等，还不能算是严格意义上的决策树，如果将所有条件量化，则就变成真正的决策树了。

决策树方法的起源是概念学习系统CLS，然后发展到ID3方法，最后又演化为能处理连续属性的C5.0.有名的决策树方法还有CART和Assistant。这里我的实现方法是ID3。

ID3算法的思想是采用贪心的策略选取分类能力最好的一个属性来，然后对于这个最佳属性的每一个取值产生一个分支，对于每一个分支都将有属于它的训练样例，重复前边的过程来完成建树。

ID3算法的伪代码实现如下（摘自《机器学习》 米歇尔（Mitchell，T.M.）：

```
ID3(Examples, Target_attribute, Attributes)
Examples即训练样例集。Target_attribute是这棵树要预测的目标属性。Attributes是除目标属性外供学习到得决策树测试的属性列表。
返回一棵能正确分类给定Examples的决策树



创建树的Root节点
如果Examples 都为正，那么返回label = + 的单节点树Root
如果Examples 都为反，那么返回label = - 的单节点树Root
如果Attributes为空，那么返回单节点树Root，label=Examples中最普遍的Target_attributes值
否则开始
    A <- Attributes中分类Examples能力最好的属性
    Root的决策属性 <- A
    对于A的每个可能值vi
        在Root下加一个新的分支对应测试A = vi
        令Examples vi为Examples中满足A属性值为vi的子集
        如果Examples vi为空
            在这个新分支下加一个叶子结点，结点的label=Examples中最普遍的Target_attribute值
        否则在这个新分支下加一个子树ID3(Examples vi, Target_attribute, Attributes - {A})
结束
返回Root
```

看起来这个算法还是很简单的，但是还有一个问题，如何选择分类能力最好的一个属性，这也是ID3算法要解决的核心问题。ID3算法使用统计学来实现的。

用一个例子来说明分类能力最好的属性是如何产生的。

现在有一组数据，数据中包括天气情况和某事是否去打网球的情况，如下。

outlook	temperature	humidity	windy	play
sunny	hot	high	weak	no
sunny	hot	high	strong	no
overcast	hot	high	weak	yes
rainy	mild	high	weak	yes
rainy	cool	normal	weak	yes
rainy	cool	normal	strong	no
overcast	cool	normal	strong	yes
sunny	mild	high	weak	no
sunny	cool	normal	weak	yes
rainy	mild	normal	weak	yes
sunny	mild	normal	strong	yes
overcast	mild	high	strong	yes
overcast	hot	normal	weak	yes
rainy	mild	high	strong	no

现在先来介绍一下熵，熵是信息论中广泛使用的一个度量标准，它是来衡量一个随机变量出现的数学期望，刻画了任意样例集的纯度。给定包含关于某个目标概念的正反样例的样例集S，那么S相对这个布尔型分类的熵为：

$$Entropy(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

比如上述数据的计算式为：

$$Entropy([9+, 5-]) = -(9/14)\ log_2(9/14) - (5/14)\ log_2(5/14) = 0.940$$

已经有了熵作为衡量训练样例集合纯度的标准，现在可以定义属性分类训练数据的效力的度量标准。这个标准被称为**信息增益**。简单的说，一个属性的信息增益就是由于使用这个属性分割样例而导致的期望熵降低(或者说，样本按照某属性划分时造成熵减少的期望)。更精确地讲，一个属性A相对样例集合S的信息增益Gain(S,A)被定义为：

$$Gain(S, A) = Entropy(S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

- V(A)是属性A的值域
- S是样本集合
- S_v 是S中在属性A上值等于v的样本集合

其中 Values(A)是属性A所有可能值的集合，是S中属性A的值为v的子集。换句话来讲，Gain(S,A)是由于给定属性A的值而得到的关于目标函数值的信息。当对S的一个任意成员的目标值编码时，Gain(S,A)的值是在知道属性A的值后可以节省的二进制位数。

为什么是二进制，很简单，计算机中的数据就是以二进制的形式存储的，信息量为n的数据就可以用log2(n) 位的二进制数来表示。

以上述信息中的wind属性为例，可以得到它的信息增益：

$$
\begin{aligned}
Values(Wind) &= Weak, Strong \\
S &= [9+, 5-] \\
S_{Weak} &\leftarrow [6+, 2-] \\
S_{Strong} &\leftarrow [3+, 3-] \\
Gain(S, Wind) &= Entropy(S) - \sum_{v \in \{Weak, Strong\}} \frac{|S_v|}{|S|} Entropy(S_v) \\
&= Entropy(S) - (8/14) Entropy(S_{Weak}) - (6/14) Entropy(S_{Strong}) \\
&= 0.940 - (8/14) 0.811 - (6/14) 1.00 \\
&= 0.048
\end{aligned}
$$

ID3算法基本上就已经介绍完了，对于上述数据我写了一个实现，由于水平有限，可能有些地方还有错误或不足，如果有人发现，请留言，谢谢。