实验环境

PyCharm-162.1237.1

TextMate ， python 2,7

一． 实验题目

写一个训练 ID3 判定树的程序，其中树的节点分支率 B 等同于每个属性装填的离散值的个数，采用增益比不纯度。

1. 利用你的程序对表中的 w1 和 w2 训练一棵树。

2. 通利用你的树分类如下数据：{B,G,I,K,N}， {C,D,J,L,M}。

3. 写出（2）的分类逻辑表达式，并化简。

4. 写出描述类别 w1 和 w2 的逻辑表达式。

二． 实验数据

类别	A-D	E-G	H-J	K-L	M-N
w1	A	E	H	K	M
w1	B	E	I	L	M
w1	A	G	I	L	N
w1	B	G	H	K	M
w1	A	G	I	L	M
w2	B	F	I	L	M
w2	B	F	J	L	N
w2	B	E	J	L	N
w2	C	G	J	K	N
w2	C	G	J	L	M
w2	D	H	J	K	M
w2	B	D	I	L	M

三． 实验过程

【一】1. 利用你的程序对表中的 w1 和 w2 训练一棵树。

（1） ID3 算法解释

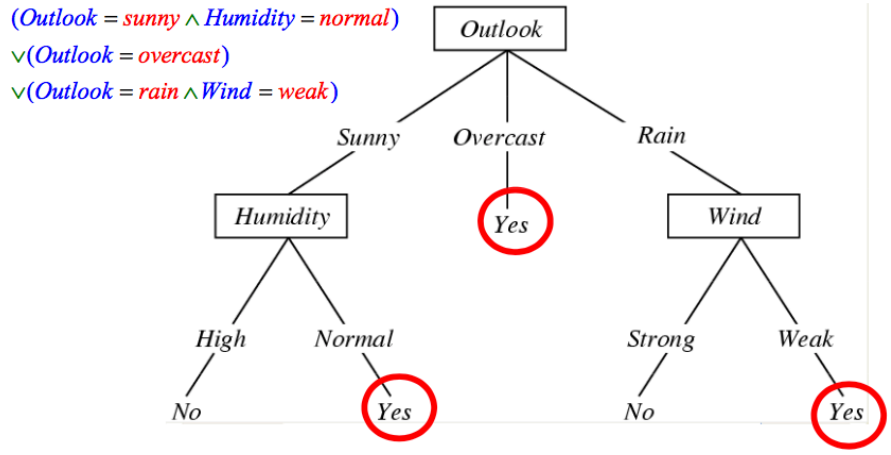
ID3 是一种自顶向下增长树的贪婪算法，在每个结点选取能最好地分类样例的属性。继续这个过程指导这棵树能完美分类训练样例，或所有的属性都已被使用过。

ID3 算法可以归纳为以下几点：

1. 使用所有没有使用的属性并计算与之相关的样本熵值
2. 选取其中熵值最小的属性
3. 生成包含该属性的节点

（2） 决策树

决策树是一种树形结构，其中每个内部节点表示一个属性上的测试，每个分支代表一个测试输出，每个叶节点代表一种类别。根据我们已经知道的属性值进行分支，知道找到符合全部要求的叶子节点，而这个叶子结点的值就是我们对于该物体的最终的决策分类。下面就是一颗划分天气的决策树，通过 Outlook,Humidity,Wind 来判断是否是好天气。和人做决策一样一步步看，先看 Outlook 再看其他。



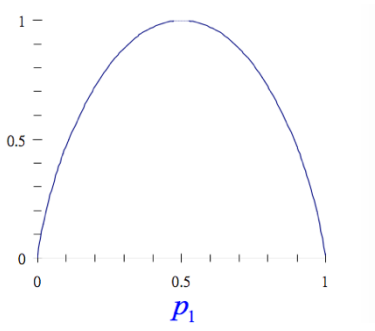
（3） 公式

ID3 算法构建出来的决策树，就是根据我们计算出来的信息熵的大小来判断当前应该用哪一个属性进行分类。

公式如下：

$$Entropy(S) = -p_1 \log_2 p_1 - p_2 \log_2 p_2$$

然后看它的图像可以得知，若 p 越平均则 Entropy 的值越大，若 Entropy 为 0 就不用再往下分了。此时的 p1 是由当前样本集中的 1 类的样本除以总样本数得到的，而 $p2$ 是由属于 2 类的样本数除以总样本数得到。



每一种属性的贡献的信息熵由如下公式计算：

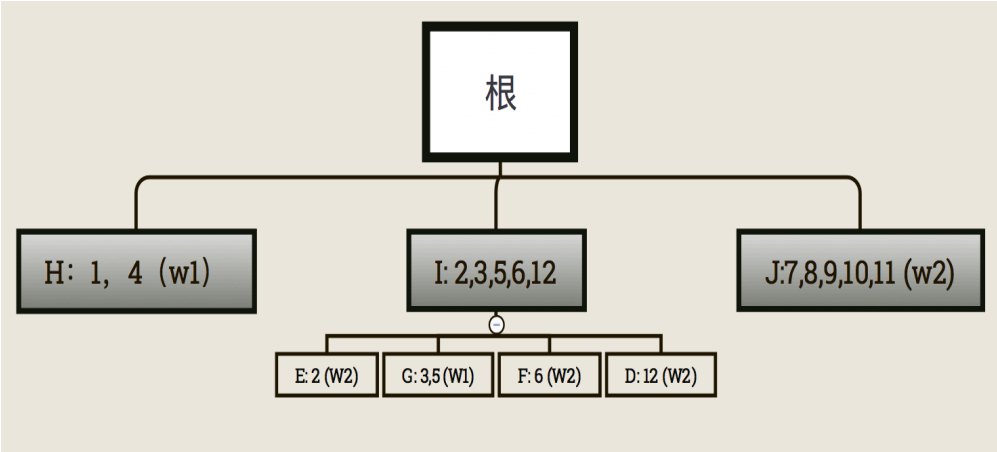
$$\text{Gain}(S,A) = \text{Entropy}(S) - \sum_{v \in A} \frac{|S_v|}{|S|} Entropy(S_v)$$

其中|S|指的是样本的个数。这样得到的 GAIN 值最大的为当前情况下分类拥有最大贡献熵的属性 A，所以我们就用 A 属性作为分支，即可。然后重复上述过程，直到决策树构建完毕为止。

（4） 实验结果

```
.....
分支
['H', 'I', 'J']
节点对应的Gain
[-0.0, 0.97095059445466858, -0.0]
I向下分支
.....
分支
['E', 'G', 'F', 'D']
节点对应的Gain
[-0.0, -0.0, -0.0, -0.0]
.....
```

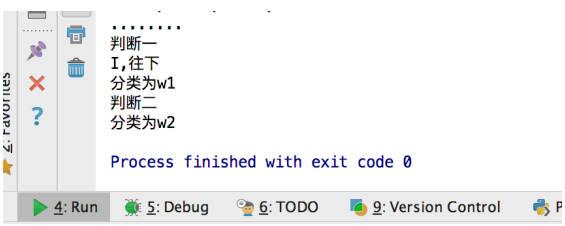
可以看到第一次计算后选择用 H-J 来作为分支，分完后，H，J 的 gain 为 0 说明已经分好类了，其实 H 全为 w1，J 全为 w2,而 I 的 gain>0 说明没有分好类，还要往下判断，I 往下分成 E,G,F,D 四支，并且 gain 都为 0,说明分好了。(感觉表里数据有点问题，但是..就这么分吧，假装 D 在 E-G 里面好了)



用 XMind 画的树图(数字代表样本序号)

【二】使用所有没有使用的属性并计算与之相关的样本熵值

把数据带入到我们生成的决策树当中，并求出结果即可。



{B,G,I,K,N}：

先判断它的第三个属性是 I 所以进入第二层第二个节点，此时没有分类成功，第二的属性是 G，所以到达第三层的第二个节点，对应了 w1 类

{C,D,J,L,M}：

同样的，我们看一下首先它的第三个属性为 J，所以进入等二层第三个节点，发现为 w2 类，结果正确。

【三】写出（2）的分类逻辑表达式，并化简。

1. 第一个表达式：

IF(H-J)=I(E-G)=G THEN属于 w1

2. 第二个表达式：

IF(H-J)=J THEN 属于 $\omega 2$

【四】写出描述类别 w1 和 w2 的逻辑表达式。

W1:

IF[(H-J)=I∧(E-G)=G]∨[(H-J)=I∧(E-G)=E] ∨(H-J)=H THEN 属于$\omega 1$

W2:

IF[(H-J)=I∧(E-G)=F]∨[(H-J)=I∧(E-G)=D] ∨(H-J)=J THEN 属于$\omega 2$

四． 收获与感悟

a) 经过这次实验首先我更加熟悉了 python 的使用

b) 我了解了 ID3 判定树的原理和使用

五． 代码

位于“代码”文件夹下