作业 2　决策树

1752931 胡斌

➢ 作业内容

本作业使用 PYTHON 语言完成，实现了根据给定数据集，以信息增益为准则选择划分属性（与 ID3 决策树算法相同），并训练生成一颗决策树的功能。完整的程序由一系列函数和主函数组成，程序中包含的函数及各自的作用如下。

- load_data()：加载数据集。
- calculate_information_entropy(current_data)：计算给定数据集的信息熵。
- find_dividing_point_and_calculate_IG(current_data, feature)：对于给定的数据集和连续属性，寻找离散化（二分）的最佳划分点和对应的最大信息增益。
- split_data_for_discrete_feature(current_data, feature, value)：对于给定的离散属性，将给定数据集中该属性值为某一指定值的样本提取出来，组成子数据集。
- split_data_for_continuous_feature(current_data, feature, dividing_point)：对于给定的连续属性，将给定数据集中该属性值小于等于或大于某一指定划分点的样本分开，分别组成两个子数据集。
- calculate_information_gain(current_data, chosen_feature)：根据给定的数据集和某一指定的属性，计算选择该属性作为当前划分属性的信息增益。
- choose_best_split_feature(current_data)：根据当前的数据集，以信息增益为准则选择最佳的划分属性。
- Vote(current_data)：当需要进行“少数服从多数”操作时，根据当前数据集，找到包含的样本数量最多的标签值（样本类别）。
- build_a_decision_tree(initial_data)：根据原始数据集，生成一颗决策树，并以字典形式存储下来。

在主函数中，使用 load_data 函数加载数据集，然后调用生成决策树的 build_a_decision_tree 函数，就能得到训练生成的决策树。生成决策树的函数调用了编写的其它函数实现其功能，最终的决策树结果存储在一个“字典”数据结构中。加载《机器学习》84 页表 4.3 给出的西瓜数据集 3.0，并以此作为训练集生成决策树，程序输出的决策树结果如图 1 所示。可根据得到的字典，将其绘制成图形的形式，如图 2 所示。

```
DecisionTree
D:\software\anaconda3_2019.07\envs\project\python.exe "C:/Users/Hu Bin/Desktop/studying/模式识别导论/作业/第2次作业-DecisionTree/DecisionTree.py"
{'texture': {'clear': {'density': {'density<=0.3815': 'bad', 'density>0.3815': 'good'}}, 'little_blurry': {'touch': {'soft': 'good', 'hard': 'bad'}}, 'blurry': 'bad'}}

Process finished with exit code 0
```

图 1 根据西瓜数据集 3.0 生成的决策树结果

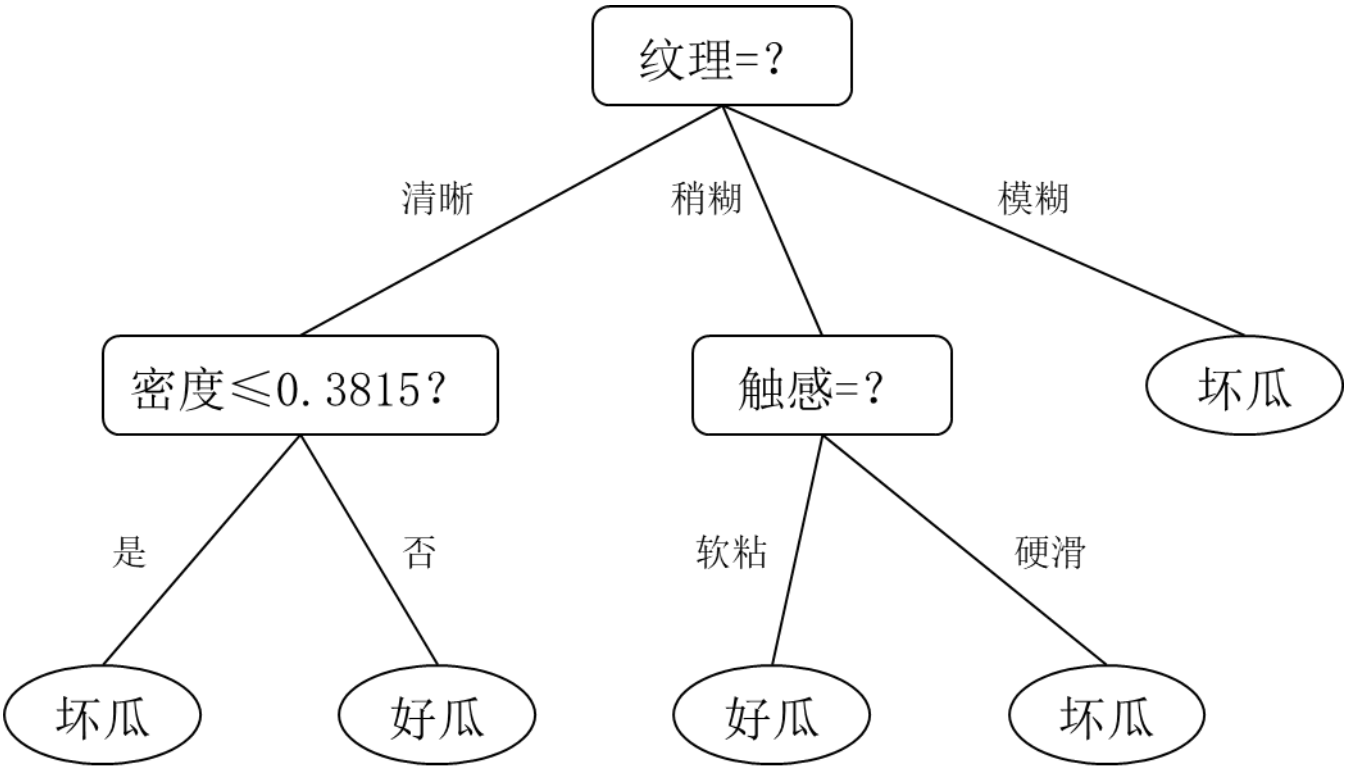

图 2 用图形表达的决策树

➢ 作业说明

本作业的程序逻辑相对清晰，在代码中有充分注释，能够比较容易地看懂。下面列出比较关键的点或在完成过程中遇到的一些问题进行说明。

- 数据集的数据结构

加载数据集使用的数据结构将影响算法的阅读性和编写难度。本作业使用一个字典结构（dictionary）存储每一个样本的信息，字典中每个键（key）是不同的属性，每个键对应对的值（value）是该样

本对应属性的值。然后，将所用样本对应的所有字典分别作为一个元素，存放在一个列表（list）中。这样可以利用字典的特性，方便的得到某一样本某一属性的值，无需对列表的下标进行过多的操作。

- 字典数据结构存储决策树

生成决策树的过程是一个递归的过程，需要使用适当的数据结构存储结果，方便之后的决策树的绘制。在 PYTHON 中，字典是用来存储树的一种常用数据结构。在递归的过程中，使用字典嵌套的方式，能够将决策树方便地存储下来。

- 深拷贝（deepcopy）

根据决策树的生成算法，对于离散属性，一旦被选用作了划分属性，在之后的递归过程中，该属性就不能再作为划分属性。在程序中，在选定了最佳划分属性并判断为离散属性后，将所有样本的该属性对应键和值删除，并进行样本的划分，做好递归的准备。但在调试过程中发现，如果在样本划分函数中简单的使用 append 方法，将会导致之后的所有样本都失去该划分属性，导致错误。这是因为 append 方法添加的内容与原内容实际是共享的，一旦删除了该内容，原内容也会被删除。这里需要使用 deepcopy 方法，问题得以解决。具体内容参见代码。

- **效果分析**

参考《机器学习》76 页表 4.1 和 80 页表 4.2 的内容，将编号为{1，2，3，6，7，10，14，15，16，17}的样例组成训练集，编号为{4，5，8，9，11，12，13}的样例组成验证集，分析决策树算法的效果。

运行对应的主函数，得到生成的决策树。这里不再给出字典形式的决策树，直接给出图形表示的决策树，如图 3 所示。字典形式的决策树可运行程序得到。

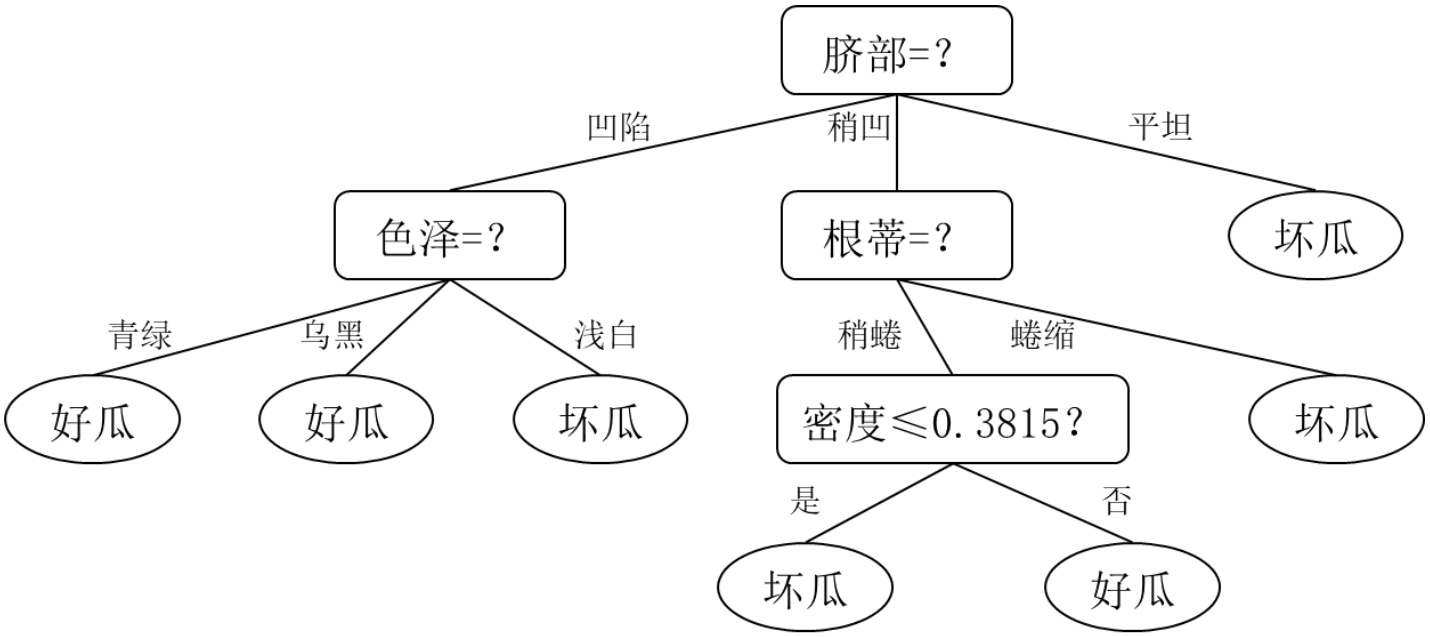

图 3 使用训练集生成的决策树

根据图 3 的决策树，对验证集中的样例进行预测，并与实际类别比对，如表 1 所示。

编号	实际类别	预测类别	是否预测正确
4	好瓜	好瓜	是
5	好瓜	坏瓜	否
8	好瓜	好瓜	是
9	坏瓜	好瓜	否
11	坏瓜	坏瓜	是
12	坏瓜	坏瓜	是
13	坏瓜	好瓜	否

表 1 验证集的预测类别与实际类别比对

由表 1 可得，根据训练集生成的决策树，其在验证集上的精度为：

$$\frac{4}{7} \approx 57.14\%$$

- **改进方向**

- 编写后续程序，根据字典存储的决策树，直接生成图形。
- 编写后续程序，根据决策树结果，生成可直接进行判断的决策树结构，使得给定新样本的数据，能够自动进行类别的判断。
- 考虑剪枝操作。
- 考虑缺失数据的情况。
- 考虑增益率，或信息增益与增益率结合的划分属性选择原则。
- ……