人工智能-第 3 次课程作业报告

授课教师：杨旭 作者：<徐子航>-<61520711>

1 问题描述

1.1 题目介绍

利用决策树判断是否需要在餐馆等座位，本问题主要基于如下属性列表基础上决策的。

1. ***Alternate***（改变）：附近是否有另一家合适的餐馆。
2. ***Bar***（酒吧）：该餐馆中供顾客等候的酒吧区是否舒适。
3. ***Fri/Sat***（周五/周六）：若是周五或周六，则为真。
4. ***Hungry***（饥饿）：我们是否饥饿。
5. ***Patrons***（顾客）：该餐馆中有多少顾客（值为 ***None***（没人）、***Some***（一些）或 ***Full***（满座））。
6. ***Price***（价格）：餐馆的价格范围（$，$$，$$$）。
7. ***Raining***（下雨）：外面是否在下雨。
8. ***Reservation***（预约）：我们是否预约过。
9. ***Type***（类型）：餐馆的种类（法式、意大利式、泰式或汉堡店）。
10. ***WaitEstimate***（等候时间估计）：餐馆主人估计的等候时间（0～10 分钟，10～30，30～60，>60）。

1.2 任务说明

利用 restaurant_willwait 数据集中的 75 条数据训练决策树模型。数据的属性值如下表。

表 1 属性值离散表示

属性	离散取值
Alternate	1 0
Bar	1 0
Fri/sat	1 0
Hungry	1 0
Patrons	0 0.5 1
Price	0 0.5 1
Raining	1 0
Reservation	1 0
Type	1 0.67 0.33 0
Waitestimate	0 0.17 0.5 1

表 2 是否等待离散表示

属性	离散取值
willwait	1 0

1.3 实验环境

Visual Stdio 2022

1.4 评价标准

$$\text{正确率A} = \frac{\text{判断正确数量}}{\text{测试集数据数量}}$$

2 实验方案

这次实验需要完成的是决策树中的信息熵计算函数、importance 函数，学习函数和分类预测函数。其中，分类预测函数 classify_rec 函数的注释给的比较完善，其它三个函数是完成的重点。

2.1 熵与 Importance 函数

熵衡量的其实是随机变量的不确定性，不确定性越高，熵越大，而随着信息的获取，确定性增加，熵也就减小。而决策树就是希望让熵值最快地下降。

下面是相关的计算公式，而 Importance 就是要选取信息收益最大的属性 A。这样能提升分类的效果。

$$\text{熵：} H(V) = \sum_k P(v_k)\log_2 \frac{1}{P(v_k)} = -\sum_k P(v_k)\log_2 P(v_k)$$

$$Bool\text{随机变量的熵：} B(q) = -(q\log_2 q + (1-q)\log_2(1-q))$$

$$\text{剩余的期望熵：} \text{Remainder}(A) = \sum_{k=1}^{d} \frac{p_k + n_k}{p+n} B\left(\frac{p_k}{p_k+n_k}\right)$$

$$\text{信息收益：} Gain(A) = B\left(\frac{p}{p+n}\right) - \text{Remainder}(A)$$

2.2 决策树学习算法

```
function DECISION-TREE-LEARNING(examples, attributes, parent_examples) returns
a tree

  if examples is empty then return PLURALITY-VALUE(parent_examples)
  else if all examples have the same classification then return the classification
  else if attributes is empty then return PLURALITY-VALUE(examples)
  else
      A ← argmax_{a ∈ attributes} IMPORTANCE(a, examples)
      tree ← a new decision tree with root test A
      for each value v_k of A do
          exs ← {e : e ∈ examples and e.A = v_k}
          subtree ← DECISION-TREE-LEARNING(exs, attributes − A, examples)
          add a branch to tree with label (A = v_k) and subtree subtree
      return tree
```

上图是教材中给出的基本学习算法，在排除了三种特殊状况(example 集为空，attribute 集为空，所有样本均为同一类)后，根据 Importance 函数选择最优的属性进行构建。当然，对于算法中的 *exs*(倒数第四行)，我认为也有必要进行是否为空的判定。

3 实验结果

根据代码中的提示，正确的决策树应该由 PATRONS 作为第一个分类属性，观察这里的树的结构，确实符合这一要求。

```
Split on PATRONS
If PATRONS == 0
        WILLWAIT: 0
If PATRONS == 0.5
        WILLWAIT: 1
If PATRONS == 1
        Split on WAITESTIMATE
        If WAITESTIMATE == 0
                WILLWAIT: 1
        If WAITESTIMATE == 0.17
                Split on ALTERNATE
                If ALTERNATE == 0
                        WILLWAIT: 1
                If ALTERNATE == 1
                        WILLWAIT: 0
        If WAITESTIMATE == 0.5
                Split on PRICE
                If PRICE == 0
                        Split on BAR
                        If BAR == 0
                                WILLWAIT: 0
                        If BAR == 1
                                WILLWAIT: 1
                If PRICE == 0.5
                        WILLWAIT: 1
                If PRICE == 1
                        WILLWAIT: 1
        If WAITESTIMATE == 1
                WILLWAIT: 0
```

实验中，我也把给的数据进行了划分，分成了训练集和测试集。最终的测试结果可以达到 93%的准确率。

```
Precision: 0.933333
```

4 实验分析

数据占比	准确率
2:1 (train-50,test-25)	92%
4:1 (train-60,test-15)	93.3%

上面的实验数据都是我把 75 条数据的前 50 条，或前 60 条拿来做训练，剩下的做测试而得到的实验结果。但随后发现了这样得到的数据的偶然性，于是又

进行了交叉验证,按照 4:1 进行数量划分,每次把分好的 5 份数据中拿一份作为测试集，剩下的为训练集。

验证序号	1	2	3	4	5
准确率/%	93.3	86.7	86.7	93.3	93.3

把这 5 次的准确率取均值，平均准确率为 90.66%。

从这个数据波动中，可以看到决策树还是很容易受到数据的影响的，稍有不慎，就会过拟合。从结果反推，数据中下面几点很可能会影响决策树的效果

- 噪声数据：其实噪声一直是影响机器学习模型的一大因素，具体到决策树中，一旦以噪声为分割标准，就难以代表真实数据
- 代表性数据缺失：这是从数据的分布方面来进行分析。数据集很可能是符合“长尾分布”的，这样，一些类的数据较少，训练出的模型也就难以很好地对这些模型进行分类。

要在一定程度上解决这些问题，除了在数据收集的时候想办法，也可以考虑采取一些模型上的优化。一些技术虽然作业框架里没有使用，但确实能够提升模型的鲁棒性。对树的剪枝操作就非常重要。最简单的方法就像深度优先算法那样，设一个预定深度，当然这里还要设一个预定宽度。也可以使用一些更“高级”的算法，如 CART 等。

5 结论

在这一次实验中，我依照提示，实现了一个简单的决策树。在进一步理解了相关理论的同时，对于决策树的优点也有了认识。首先决策树这种模型的可视化非常容易，相较于现在深度学习中的各种模型要直观地多。 而且决策树对于数据量的需求也不大。这里给的数据只有 75 条，我自己进行测试时只用了 50~60 条数据，就获得了一个似乎还不错的模型。当然，一些缺点也很明显，对于一些更复杂的任务难以胜任，而且很容易过拟合，需要配合一些复杂的剪枝技术才能更好的运作。不过，联系到我正在学习的计算机视觉、自然语言处理等课程，我可以感受到这些“传统算法”的魅力与威力。神经网络、深度学习等等确实很有用，但这些传统算法的优势丝毫不弱。最近也有学者在把两者结合组建模型，或许这才是正确的方向。不论是传统算法还是深度学习算法，都应该基于实际问题去选择和设计。