人工智能原理与技术第 11 周作业

2154312 郑博远

打球决策树 数据：14 天的气象数据（属性：outlook，temperature，humidity，windy），并已知这些天气是否打球（play）。

问题：根据决策树算法，构建一棵是否打球的决策树。并给出每个结点选择分裂属性时所作的计算。

解答：

编写 Python 程序，每次选出信息增益最大的标签作为划分，直到没有可供选择的标签或者划分后的是否打球决策均一致。代码如下：

```
import math
import numpy as np
import pandas as pd

# 计算熵
def calEntropy(data):
    p = 0
    n = 0
    for e in data['answer']:
        if(e):
            p = p + 1
        else:
            n = n + 1
    q = p / (p + n)

    if(q == 1 or q == 0):
        return 0
    else:
        return -(q * math.log(q, 2) + (1 - q) * math.log(1 - q, 2))

# 递归进行划分
def getChoice(data, label):
    # 没有标签了 划分结束
    if(len(label) == 0):
       return

    maxgain = 0
    for l in label:
```

```
        newEntropy = 0
        for ref, subdata in data.groupby(l):
            newEntropy += calEntropy(subdata) * subdata.shape[0] / data.shape[0]
        gain = calEntropy(data) - newEntropy
        print('按照', l, '进行划分，信息增益为', gain)
        # 选取信息增益最大的，进行记录
        if(gain > maxgain):
            maxgain = gain
            maxlabel = l

    print('本次选择', maxlabel, '最大信息增益为', maxgain)
    newlabel = label
    newlabel.remove(maxlabel)

    for ref, subdata in data.groupby(maxlabel):
        # answer 统一，划分完全
        if(calEntropy(subdata) == 0):
            print(maxlabel, ref, '已经划分完全')
            continue
        print('下面对', ref, '进一步划分')
        getChoice(subdata, newlabel)


data = pd.DataFrame({'outlook' :
['sunny','sunny','overcast','rainy','rainy','rainy','overcast','sunny','sunny','
rainy','sunny','overcast','overcast','rainy'],
                     'temperature' :
['hot','hot','hot','mild','cool','cool','cool','mild','cool','mild','mild','mild
','hot','mild'],
                     'humidity':['high','high','high','high','normal','normal','
normal','high','normal','normal','normal','high','normal','high'],
                     'windy':[False,True,False,False,False,True,True,False,False
,False,True,True,False,True],
                     'answer':[False,False,True,True,True,False,True,False,True,
True,True,True,True,False]})


label = { 'outlook', 'temperature', 'humidity', 'windy' }

getChoice(data, label)
```

程序的输出如下：

```
按照 outlook 进行划分，信息增益为 0.24674981977443922
按照 humidity 进行划分，信息增益为 0.15183550136234159
```

```
按照 temperature 进行划分，信息增益为 0.02922256565895487
按照 windy 进行划分，信息增益为 0.04812703040826949
本次选择 outlook 最大信息增益为 0.24674981977443922
outlook overcast 已经划分完全
下面对 rainy 进一步划分
按照 humidity 进行划分，信息增益为 0.01997309402197478
按照 temperature 进行划分，信息增益为 0.01997309402197478
按照 windy 进行划分，信息增益为 0.9709505944546686
本次选择 windy 最大信息增益为 0.9709505944546686
windy False 已经划分完全
windy True 已经划分完全
下面对 sunny 进一步划分
按照 humidity 进行划分，信息增益为 0.9709505944546686
按照 temperature 进行划分，信息增益为 0.5709505944546686
本次选择 humidity 最大信息增益为 0.9709505944546686
humidity high 已经划分完全
humidity normal 已经划分完全
```

即，首次划分时选择信息增益最大的“outlook”作为决策依据。“outlook”可将数据分为“overcast”、“sunny”与“rainy”三部分。其中“overcast”中对应决策均为“True”无需继续划分；在“sunny”下继续划分，选出信息增益最大的“humidity”作为决策依据，按照“high”与“ normal”划分后决策均一致，无需继续划分；同理按照“windy”继续对“rainy”进行划分。决策树如下图所示：