西安電子科技大學
XIDIAN UNIVERSITY

人工智能学院

智能数据挖掘课程作业报告

决策树划分

姓名：杨文韬

学号：18020100245

班级：1920012

2022 年 04 月 03 日

目录

决策树划分

1 问题描述

要求: 天气因素有温度、湿度和风况等，通过给出数据，使用决策树算法学习分类，输出一个人是运动和不运动与天气之间的决策树。数据集如表 1 所示。

表 1: 数据集

天气	温度	湿度	风况	运动
晴	85	85	无	不适合
晴	80	90	有	不适合
多云	83	78	无	适合
有雨	70	96	无	适合
有雨	68	80	无	适合
有雨	65	70	有	不适合
多云	64	65	有	适合
晴	72	95	无	不适合
晴	69	70	无	适合
有雨	75	80	无	适合
晴	75	70	有	适合
多云	72	90	有	适合
多云	81	75	无	适合
有雨	71	80	有	不适合

2 原理分析

2.1 划分选择

决策树划分我们希望决策树分支结点包含样本尽可能属于同一类别，即结点纯度 (purity) 越来越高。

2.1.1 信息增益

数据集 D 中第 k 类样本所占的比例为 $p_k(k = 1, 2, \cdots, K)$，则 D 的信息熵:

$$H(D) = -\sum_{k=1}^{K} p_k \log_2 p_k \tag{1}$$

$H(D)$ 的值越小，则 D 的纯度越高。针对某个特征 A，对于数据集 D 的条件熵 $H(D|A)$ 为：

$$H(D|A) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} H(D^v) \tag{2}$$

其中 D^v 表示 D 中特征 A 取第 v 个值的样本子集。属性 A 对样本集 D 进行划分所获得的信息增益 (Information Gain) 为

$$\text{Gain}(D, a) = H(D) - H(D|A) \tag{3}$$

信息增益越大表示使用特征 A 来划分所获得的“纯度提升”越大。ID(Iterative Dichotomiser)3 决策树学习算法以信息增益为准则划分属性，

2.1.2 增益率

信息增益准则对取值数目较多的属性有所偏好，利用信息增益率 (Gain ratio) 可以克服信息增益的缺点，其定义为

$$\text{Gain}_{\text{ratio}}(D, A) = \frac{\text{Gain}(D, A)}{H_A(D)} \tag{4}$$

其中

$$H_A(D) = -\sum_{v=1}^{V} \frac{|D^v|}{|D|} \log_2 \frac{|D^v|}{|D|} \tag{5}$$

称为特征 A 的固有值 (intrinsic value)。C4.5 决策树算法使用增益率来选择最优划分属性。需要注意的是，信息增益率对可取值较少的特征有所偏好（分母越小，整体越大），因此 C4.5 并不是直接用增益率最大的特征进行划分，而是使用一个启发式方法：先从候选划分特征中找到信息增益高于平均值的特征，再从中选择增益率最高的。

2.1.3 基尼指数

熵模型拥有大量耗时的对数运算，基尼指数在简化模型的同时还保留了熵模型的优点。基尼指数代表了模型的不纯度，基尼指数越小，不纯度越低，特征越好。这和信息增益（率）正好相反。

$$\begin{aligned} \text{Gini}(D) &= \sum_{k=1}^{K} p_k(1 - p_k) \\ &= 1 - \sum_{k=1}^{K} p_k^2 \end{aligned} \tag{6}$$

属性 A 的基尼指数定义为

$$\text{Gini}(D|A) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} \text{Gini}(D^v) \tag{7}$$

2.1.4 总结对比

下面总结对比 ID3、C4.5 和 CART 三者之间的差异。

- 划分标准的差异：ID3 使用信息增益偏向特征值多的特征，C4.5 使用信息增益率克服信息增益的缺点，偏向于特征值小的特征，CART 使用基尼指数克服 C4.5 需要求 log 的巨大计算量，偏向于特征值较多的特征。
- 使用场景的差异：ID3 和 C4.5 都只能用于分类问题，CART 可以用于分类和回归问题；ID3 和 C4.5 是多叉树，速度较慢，CART 是二叉树，计算速度很快；
- 样本数据的差异：ID3 只能处理离散数据且缺失值敏感，C4.5 和 CART 可以处理连续性数据且有多种方式处理缺失值；从样本量考虑的话，小样本建议 C4.5、大样本建议 CART。C4.5 处理过程中需对数据集进行多次扫描排序，处理成本耗时较高，而 CART 本身是一种大样本的统计方法，小样本处理下泛化误差较大；
- 样本特征的差异：ID3 和 C4.5 层级之间只使用一次特征，CART 可多次重复使用特征；
- 剪枝策略的差异：ID3 没有剪枝策略，C4.5 是通过悲观剪枝策略来修正树的准确性，而 CART 是通过代价复杂度剪枝。

3 实验过程

3.1 理论计算

适合占 $p_1 = \frac{9}{14}$，不适合占 $p_2 = \frac{5}{14}$，根结点信息熵为

$$H(D) = -\sum_{k=1}^{2} p_k \log_2 p_k = 0.940 \tag{8}$$

以属性'weather' 为例对 D 进行划分，得到 3 个子集 D^1(weather= 晴)，D^2(weather= 多云)，D^3(weather= 有雨)，计算 3 个分支结点信息熵

$$\begin{aligned} H(D^1) &= -\left(\frac{3}{5}\log_2\frac{3}{5} + \frac{2}{5}\log_2\frac{2}{5}\right) = 0.971 \\ H(D^2) &= 0 \\ H(D^3) &= -\left(\frac{3}{5}\log_2\frac{3}{5} + \frac{2}{5}\log_2\frac{2}{5}\right) = 0.971 \end{aligned} \tag{9}$$

计算'weather' 属性信息增益如下

$$\begin{aligned}\text{Gain}(D, weather) &= H(D) - \sum_{v=1}^{3} \frac{|D^v|}{|D|} H(D^v) \\ &= 0.940 - \left(\frac{5}{14} \times 0.971 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.971 \right) \\ &= 0.246\end{aligned} \tag{10}$$

类似可计算其他属性信息增益，属性'temperature' 信息增益最大，它被选为划分属性。具体结果见 3.2。

3.2 Python 创建决策树

导入数据集后进行预处理，将值进行转换，'weather' 属性中晴->0、多云->1、有雨->2，'wind' 属性中无->0、有->1，'sports' 属性中不适合->no、适合->yes。然后设置'sports' 属性为因变量，其他属性为自变量。

然后采用库函数 `sklearn.tree.DecisionTreeClassifier` 中 ID3 算法划分决策树，设置树最大深度为 4，采用库函数 `sklearn.tree.plot_tree` 绘制树图，划分结果如图 1 所示。具体代码见附录。

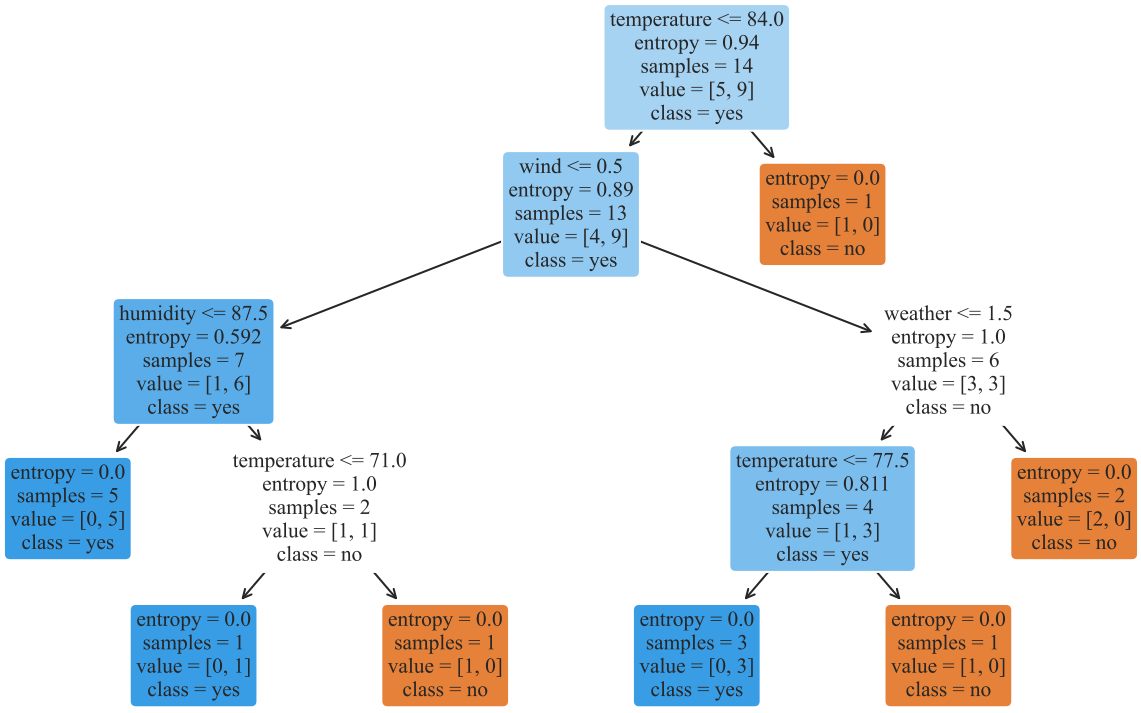

图 1: 决策树划分结果

4 附录

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.tree import DecisionTreeClassifier as dtc # 树算法
from sklearn.tree import plot_tree # 树图
%matplotlib inline
```

In [2]:
```
sns.set_style("darkgrid", {"grid.color": ".6", "grid.linestyle": ":"})
sns.set_theme(font='Times New Roman', font_scale=1.2)
plt.rc("figure", autolayout=True)
# Chinese support
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False
```

In [3]:
```
df = pd.read_csv('dataset.csv')
df
```

Out[3]:

	weather	temperature	humidity	wind	sports
0	晴	85	85	无	不适合
1	晴	80	90	有	不适合
2	多云	83	78	无	适合
3	有雨	70	96	无	适合
4	有雨	68	80	无	适合
5	有雨	65	70	有	不适合
6	多云	64	65	有	适合
7	晴	72	95	无	不适合
8	晴	69	70	无	适合
9	有雨	75	80	无	适合
10	晴	75	70	有	适合
11	多云	72	90	有	适合
12	多云	81	75	无	适合
13	有雨	71	80	有	不适合

In [4]:
```
for i in df['weather'].values:
    if i == '晴':
        df['weather'].replace(i, 0, inplace=True)
```

```
        elif i == '多云':
            df['weather'].replace(i, 1, inplace=True)
        elif i == '有雨':
            df['weather'].replace(i, 2, inplace=True)

    for i in df['wind'].values:
        if i  == '无':
            df['wind'].replace(i, 0, inplace=True)
        elif i == '有':
            df['wind'].replace(i, 1, inplace=True)

    for i in df['sports'].values:
        if i == '不适合':
            df['sports'].replace(i, 'no', inplace=True)
        elif i == '适合':
            df['sports'].replace(i, 'yes', inplace=True)

    df
```

Out[4]:

	weather	temperature	humidity	wind	sports
0	0	85	85	0	no
1	0	80	90	1	no
2	1	83	78	0	yes
3	2	70	96	0	yes
4	2	68	80	0	yes
5	2	65	70	1	no
6	1	64	65	1	yes
7	0	72	95	0	no
8	0	69	70	0	yes
9	2	75	80	0	yes
10	0	75	70	1	yes
11	1	72	90	1	yes
12	1	81	75	0	yes
13	2	71	80	1	no

```
In [5]: X_var = df[['weather', 'temperature', 'humidity', 'wind']].values # 自变量
        y_var = df['sports'].values # 因变量

In [6]: model = dtc(criterion = 'entropy', max_depth = 4)
```

```
        model.fit(X_var, y_var)
```

Out[6]:
```
DecisionTreeClassifier(criterion='entropy', max_depth=4)
```

In [7]:
```
        feature_names = df.columns[:4].tolist()
        target_names = df['sports'].unique().tolist()

        plt.figure(figsize=(10, 6), dpi=80)
        plot_tree(model,
                  feature_names=feature_names,
                  class_names=target_names,
                  filled=True,
                  rounded=True)

        #plt.savefig('tree_result.pdf')
        plt.show()
```