智能数据挖掘作业 4

19200300004 黄铭瑞

实验目的

掌握使用决策树算法进行分类的方法。

实验原理

决策树

决策树（Decision Tree）是有监督学习中的一种算法，并且是一种基本的分类与回归的方法。也就是说，决策树有两种：分类树和回归树。本次实验使用的是分类树。

决策树包含有根节点，中间节点，叶子节点。根节点只出不进，中间节点可进可出，但是进边只有一条，出边可多条，叶子节点只进不出。两个相连的节点中，靠近根节点的是父节点，远离根节点的是子节点。

香农熵

随着划分的进行，我们希望属于同一类的样本尽量分到一块，即希望节点纯度要尽量的高。我们用香农熵来度量节点的纯度。

$$Entropy(D) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i)$$

$Entropy(D)$:香农熵，$p(x_i)$:属于第 i 类样本占的比例，$Entropy(D)$越小，说明信息越纯，不确定性越小。

信息增益

信息增益计算父节点的信息熵和他的所有子节点的信息熵求和后的差。

$$Gain(D,a) = Entropy(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Entropy(D^v)$$

$Gain(D,a)$:信息增益，$Entropy(D)$:父节点的信息熵，D^v: 第 v 个分支节点包含 D 中所有属性 a 上取值为a^v的样本集，D：父节点包含的样本集，$Entropy(D^v)$:子节点的信息熵。

举个例子：

	天气	温度	湿度	风况	运动
0	晴	热	湿	无	不适合
1	晴	热	湿	有	不适合
2	多云	热	干	无	适合
3	有雨	凉	湿	无	适合
4	有雨	凉	湿	无	适合
5	有雨	凉	干	有	不适合
6	多云	凉	干	有	适合
7	晴	热	湿	无	不适合
8	晴	凉	干	无	适合
9	有雨	热	湿	无	适合
10	晴	热	干	有	适合
11	多云	热	湿	有	适合
12	多云	热	干	无	适合
13	有雨	凉	湿	有	不适合

（天气与运动_已处理）

运动这一项的信息熵为：$playEnt = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940$

	天气	温度	湿度	风况	运动
0	晴	热	湿	无	不适合
1	晴	热	湿	有	不适合
2	多云	热	干	无	适合
7	晴	热	湿	无	不适合
9	有雨	热	湿	无	适合
10	晴	热	干	有	适合
11	多云	热	湿	有	适合
12	多云	热	干	无	适合

（天气与运动_已处理，df[‘温度’] = ‘热’）

	天气	温度	湿度	风况	运动
3	有雨	凉	湿	无	适合
4	有雨	凉	湿	无	适合
5	有雨	凉	干	有	不适合
6	多云	凉	干	有	适合
8	晴	凉	干	无	适合
13	有雨	凉	湿	有	不适合

（天气与运动_已处理，df[‘温度’] = ‘凉’）

温度为“热”的有 8 个，其中“适宜”有 5 个，温度为“凉”的有 6 个，其中“适宜”有 4 个。

温度的信息熵为：

$$temEnt = -\frac{8}{14}\left[\frac{5}{8}\log_2\frac{5}{8}+\frac{3}{8}\log_2\frac{3}{8}\right]-\frac{6}{14}\left[\frac{4}{6}\log_2\frac{4}{6}+\frac{2}{6}\log_2\frac{2}{6}\right]=0.939$$

“温度”的信息增益为：

$$Gain = playEnt - temEnt = 0.001$$

ID3 算法

ID3 算法，即 Iterative Dichotomiser3，迭代二叉树 3 代，是 Ross Quinlan 发明的一种决策树算法，这个算法的基础就是奥卡姆剃刀原理，越是小型的决策树越优于大的决策树，尽管如此，也不总是生成最小的树型结构，而是一个启发式算法。

在决策树的每一个非叶子结点划分之前，先计算每一个属性所带来的信息增益，选择最大信息增益的属性来划分，因为信息增益越大，区分样本的能力就越强，越具有代表性，很显然这是一种自顶向下的贪心策略。以上就是 ID3 算法的核心思想。

实验过程

数据集部分

数据集获取

根据题目要求整理数据，并制成 csv 文件“天气与运动.csv”。

```
•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
```

（天气与运动.csv）

数据集处理

“温度”中，把大于其均值的数替换为“热”，把小于其均值的数替换为“凉”。
“湿度”中，把大于其均值的数替换为“湿”，把小于其均值的数替换为“干”。

```
天气,温度,湿度,风况,运动
晴,热,湿,无,不适合
晴,热,湿,有,不适合
多云,热,干,无,适合
有雨,凉,湿,无,适合
有雨,凉,干,无,适合
有雨,凉,干,有,不适合
多云,凉,干,有,适合
晴,凉,湿,无,不适合
晴,凉,干,无,适合
有雨,热,干,无,适合
晴,热,干,有,适合
多云,凉,湿,有,适合
多云,热,干,无,适合
有雨,凉,干,有,不适合
```

（天气与运动_已处理.csv）

并保存为“天气与运动_已处理.csv”

生成决策树部分

导入数据集

定义 loadDataSet()，用于数据集的导入。导入后得到 dataSet 和 dataLabels。

```
[['晴', '热', '湿', '无', '不适合'],
 ['晴', '热', '湿', '有', '不适合'],
 ['多云', '热', '干', '无', '适合'],
 ['有雨', '凉', '湿', '无', '适合'],
 ['有雨', '凉', '干', '无', '适合'],
 ['有雨', '凉', '干', '有', '不适合'],
 ['多云', '凉', '干', '有', '适合'],
 ['晴', '凉', '湿', '无', '不适合'],
 ['晴', '凉', '干', '无', '适合'],
 ['有雨', '热', '干', '无', '适合'],
 ['晴', '热', '干', '有', '适合'],
 ['多云', '凉', '湿', '有', '适合'],
 ['多云', '热', '干', '无', '适合'],
 ['有雨', '凉', '干', '有', '不适合']]
```

（dataSet）

```
['天气', '温度', '湿度', '风况', '运动']
```

（dataLabels）

香农熵函数

定义 calcShannonEnt（），计算每个类别的香浓熵

划分数据集

定义 splitDataSet（），按照给定的特征划分数据集，返回第 featNum 个特征其值为 value 的样本集合，且返回的样本数据中已经去除该特征。

特征选择

定义 chooseBestFeatToSplit（），选出最好的特征以划分数据集。计算样本的熵，以每个特征进行分类，找出让信息增益最大的特征。

确定叶子节点类型

定义 majorityCnt（），如果决策树递归生成完毕，且叶子节点中样本不是属于同一类，则以少数服从多数原则确定该叶子节点类别

创建决策树

定义 createDecideTree（），所有样本属于同一类时，停止划分，返回该类别。所有特征已经遍历完，停止划分，返回样本数最多的类别。选择最好的特征来划分。以字典的形式表示树，根据选择的特征，遍历该特征的所有属性值，

在每个划分子集上递归调用 createDecideTree（）来生成子树。

绘制决策树部分

获取叶子节点数目

定义 getNumLeafs（）。统计节点个数，以字典的方式获取根节点的特征，递归遍历叶子节点，如果下一个节点仍为仍为字典，说明下一个节点仍然是树，调用递归，否者为叶子节点。

获取树的深度

定义 getTreeDepth（）。遍历所有子节点，如果节点类型为字典，说明下一个节点仍为一棵树，调用递归，子节点的深度加 1。遍历完成后，找出子节点的最大深度作为树的深度。

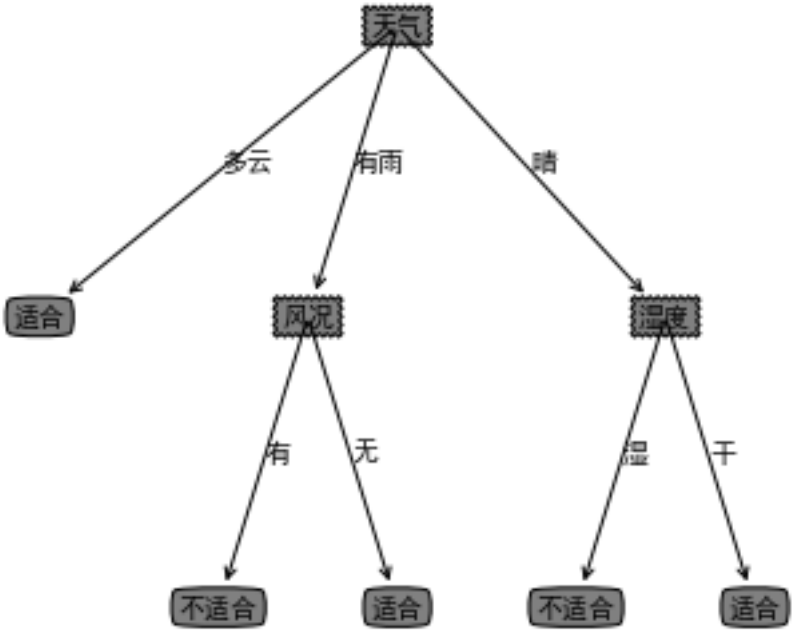
绘制节点以及边信息

定义 createPlot（）。作为画板来绘制图像。
定义 plotNode（）。用来绘制节点。根节点和中间节点用波浪线框，叶子节点用光滑线框。
定义 plotMidText（）。计算出父子节点的位置，并填充他们的边。
定义 plotTree（）。完成整个绘制过程。

实验结果与分析

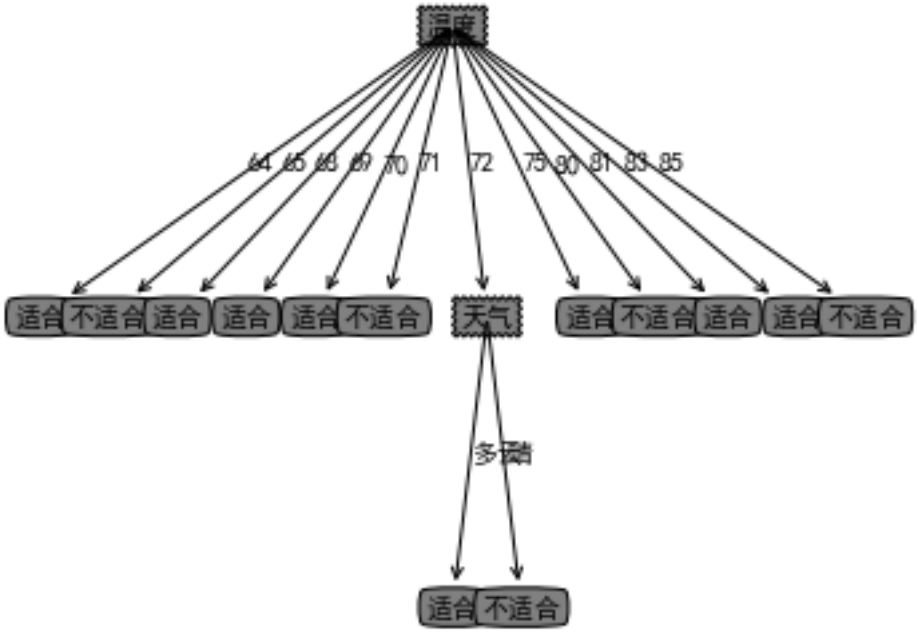
实验结果

决策树模型：
{'天气': {'多云': '适合', '有雨': {'风况': {'有': '不适合', '无': '适合'}}, '晴': {'湿度': {'湿': '不适合', '干': '适合'}}}}



（天气与运动_已处理的决策树）

以下是原数据所绘制的决策树：
决策树模型：
{'温度': {64: '适合', 65: '不适合', 68: '适合', 69: '适合', 70: '适合', 71: '不适合', 72: {'天气': {'多云': '适合', '晴': '不适合'}}, 75: '适合', 80: '不适合', 81: '适合', 83: '适合', 85: '不适合'}}



（天气与运动的决策树）

比较两者，不难看出原始数据作出的决策树过于复杂化。由于使用的 ID3，采用信息增益进行数据分裂容易偏向取值较多的特征，没有采用剪枝，决策树

的结构过于复杂，出现过拟合。当离散数据足够大，可视为连续时，ID3 就不能有效处理了。

附录

天气运动决策树.html