决策树实验报告

任俊彦　520021910546
电子信息与电气工程学院 自动化

目录

1 问题描述

1.1 问题设定

本次作业选择在 UCI 机器学习数据库中的鲍鱼数据集 (Abalone Dataset), 数据集链接 https://archive.ics.uci.edu/ml/datasets/Abalone。该数据集共 4177 例样本，其含有离散和连续的两类特征，便于检验编写的决策树模型的正确性。设定任务为：依据鲍鱼样本的各易测量的形态特征（长度特征、质量特征等），代替传统的显微镜观察法，预测鲍鱼的实际年龄，实际年龄被分为年幼、成熟及年长三类。本次作业编写决策树模型对鲍鱼年龄进行三分类预测。

1.2 问题分析

给定数据集的鲍鱼年龄为分布在 $1-29$ 间的整数，其分布如下表。

年龄	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
数量	1	1	15	57	115	259	391	568	689	634	487	267	203	126
年龄	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**29**
数量	103	67	58	42	32	26	14	6	9	2	1	1	2	1

在数据集说明中指出已有基于此数据集的一系列研究，其均将鲍鱼分成年幼（1-8 岁），成熟（9-10 岁）及年长（11 岁及以上）并构建相应的分类模型。依据此分类方法。各类样本数量均充足且差距不显著，其具有合理性，故沿用此三分类方法。

2 数据清洗

在对数据代入模型求解前，需要处理空数据并剔除不合理数据。经分析，鲍鱼数据集中不存在空数据且数据数值区间均合理。值得注意的是，决策树可以采取特定策略训练具有缺失值的数据。

3 算法策略

3.1 连续型特征的处理

鲍鱼数据集中有若干连续型特征。由于连续型特征可取数目不再有限，故不能依据其取值来进行属性划分，需采取特定离散化方法。本次作业采用 C4.5 中采用的二分法，即对连续特征值 $a_1, a_2, ..., a_n$ 先进行排序，确定划分候选点集合

$$T_a = \{\frac{a_i + a_{i+1}}{2} | 1 \leq i \leq n-1\} \tag{1}$$

并在集合中选取信息增益最大的划分点作为二分划分点，即

$$t_{best} = \underset{t \in T_a}{\operatorname{argmax}}\, Entropy(D) - \sum_{\lambda \in \{-,+\}} \frac{|D_t^\lambda|}{|D|} Entropy(D_t^\lambda). \tag{2}$$

将数值大小在划分点两侧的数据样本归为两类。

3.2 属性划分指标选取

三大决策树算法 ID3，C4.5，CART 中，ID3 与 C4.5 均采用了信息熵及信息增益（信息增益比）作为属性划分指标，而 CART 中采用了 Gini 指数。Gini 指数可以看作信息熵计算的一阶近似，其统计学意义为从数据集中随机抽取两个样本，其类别标记不一致的概率。相较于信息熵，其计算式中没有对运算部分，故效率稍有提高。本次作业对两种指标均有代码实现，代码中可选，详见代码实现部分。

3.3 递归终止条件

本次作业实现决策树采用递归方法。依据决策树算法逻辑并保证算法开销不过大，递归终止条件设定为：

1. 分裂至某一分枝时数据样本的类别一致
2. 数据样本的取值一致，或数据特征集为空
3. 决策树达到设定的最大深度

3.4 剪枝策略

采取剪枝策略可以降低过拟合的风险。常用的剪枝策略有预剪枝和后剪枝。预剪枝在决策树生成的过程中抑制其展开而达到剪枝目的，可以减少决策树生成的开销，但亦可能导致欠拟合。后剪枝在决策树生成后再对非叶节点逐一考察并剪枝，欠拟合风险较小，且泛化性能往往优于预剪枝，但其开销极大。本次作业采取 C4.5 中的悲观剪枝法，其为后剪枝方法之一；本次作业中设定的最大深度终止条件也可看作是预剪枝的一种。

对于悲观剪枝法的代码实现，使用递归完成自下而上的节点遍历检查，对剪枝前和剪枝后的子树在验证集上验证其准确度，若剪枝后的准确度高，则剪去该枝，详见 $DecisionTree.\_CheckNode()$. 对于验证集和训练集的划分，采取从每类中分别选取 20% 作验证集的策略。

3.5 五倍交叉验证

评估模型表现使用 K 倍交叉验证 (K-Fold Cross Validation)，其会得到 K 个模型，K 个模型分别在验证集中评估结果，最后的误差取平均即得到交叉验证误差。交叉验证有效利用了有限的数据，并且评估结果能够尽可能接近模型在测试集上的表现，可以做为模型优化的指标使用。本次作业将交叉验证写在了决策树的类里，后续可考虑写一个线性模型的父类并将验证写在父类中。代码中 K 默认选 5 且可调。

4 代码实现

4.1 代码结构

决策树的类实现在 *algorithm.py* 中，*main.py* 给出示例程序，可以直接运行得到本文中所有展示的结果，欢迎尝试。

4.2 类 API

决策树类接口定义如下：

1. 创建实例：mytree = DecisionTree $(data, label)$
2. 修改配置：mytree.Config $(max\_depth = 15, gini = False, prune = False)$

   若不修改配置，配置默认为最大深度 15，使用信息熵作为划分指标，不采用后剪枝。
3. 五倍交叉验证模型表现：mytree.Model_test ()

欲验证决策树在某数据集上的表现，在创建实例、修改配置后，直接使用 Model_test 函数即可，若欲单独在全数据集上训练决策树并进行预测，代码亦提供两个接口函数：

1. 构建决策树：mytree.BuildTree () -> dict: Dicision Tree
2. 决策树预测：mytree.Inference $(data, label, tree)$ -> int: error

   其中 tree 为要用于预测的决策树。

关于类的具体实现，请参见 *algorithm.py*，文件里写有详细的注释，此处不赘述。

5 模型表现

5.1 剪枝与准确率

设置不同的最大深度，选取信息熵为属性划分指标，得到在训练集与测试集上的准确率与决策树的最大深度关系如下图。其中十字数据点的两条图线为采用悲观后剪枝策略后的准确率曲线。

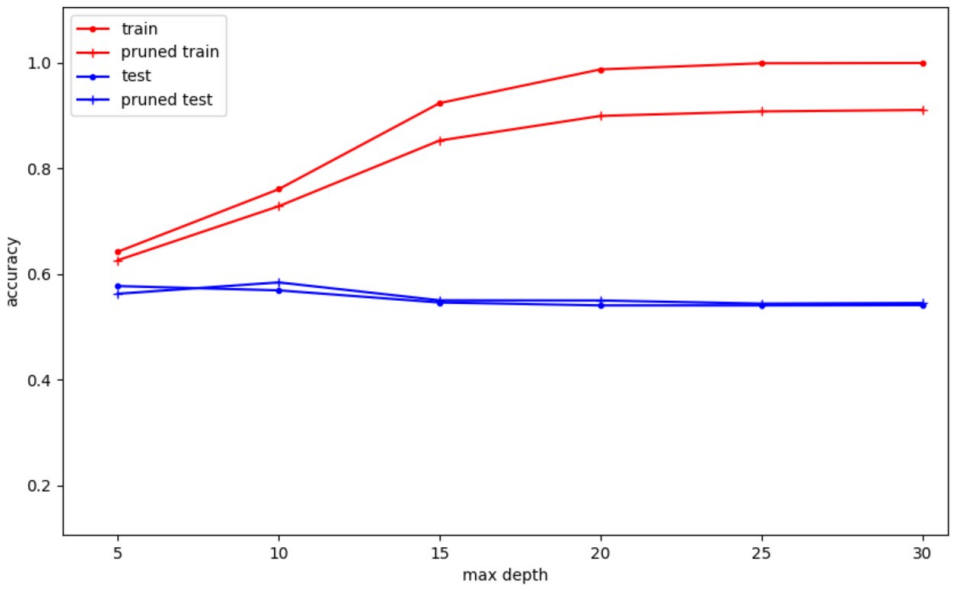

图 1: 剪枝与准确率关系

可以发现，当最大深度较小时，决策树展开不充分，训练集与测试集误差均较大，随着最大深度的增大，决策树在训练集上趋近于展开完全，准确率提升，且在最大深度为 25 时，决策树完全展开，准确率提升至 1，而在测试集上准确率未见提升，即其泛化性能未得到提升。

对于后剪枝，可以观察到在进行悲观后剪枝后训练集准确率下降，与理论预期一致。而测试集准确率仅有微小提升，即泛化性能未见明显提升。此结果应当是本决策树模型在该数据集上的局限性所导致，可以考虑采用更合适的连续值的离散化标准而非简单的二分法。

5.2 信息熵与基尼系数

分别采用信息熵与基尼系数作为划分指标，最大深度设置为 15，观察两模型决策树构建结果及构建时间，发现使用两种划分指标生成的决策树相同，而基尼系数用时更短，即基尼系数作为划分指标效率更高。事实上，观察结果可知该数据集上构建的决策树没有使用离散特征节点作为分枝决策变量，对于二分法划分的连续特征而言，采用基尼系数和信息熵作为划分指标不会改变其分支的构建结果。

划分指标	信息熵	基尼系数
运行时间/秒	50.1	48.2

5.3 模型结果

由于鲍鱼数据集有 4177 个样本，其构建的决策树规模较大，可视化树规模较大，故此处仅给出在最大深度为 10，未进行剪枝时的模型在测试集上的预测准确率表，并与已有研究 (Waugh, 1995) 的准确率进行比较，结果如下表。结果表明，决策树在该数据集上的表现不理想，若继续考虑使用决策树解决该问题，可能需要考虑采用更合适的连续值的离散化标准而非简单的二分法。

决策树模型	本次作业	Waugh, 1995
预测准确率	0.57	0.59