**基于决策树的分类算法及其可视化**

摘 要

决策树是一种非参数的有监督学习方法，它能够从一系列有特征和标签的数据中总结出决策规则，并用树状图的结构来呈现这些规则，以解决分类和回归问题。决策树中每个内部节点表示一个属性上的判断，每个分支代表一个判断结果的输出，最后每个叶节点代表一种分类结果。本实验基于 Python语言，运用ID3，C4.5，CART三种不同算法构建决策树进行目标分类，利用前后剪枝对决策树进行优化，并比较分析三种算法的特点及其效率，最后通过Python中的xxx实现可视化。

**关键词：**决策树，目标分类，剪枝，……

**Classification algorithm based on decision tree and its visualization**

**ABSTRACT**

Decision tree is a nonparametric supervised learning method. It can summarize decision rules from a series of data with characteristics and labels, and present these rules with the tree structure to solve the classification and regression problems. Each internal node in the decision tree represents a judgment on an attribute, each branch represents the output of a judgment result, and finally each leaf node represents a classification result. Based on the python language, this experiment uses ID3, C4.5 and CART to build a decision tree for target classification, uses pre-pruning and post-pruning to optimize the decision tree, compares and analyzes the characteristics and efficiency of the three algorithms, and finally realizes visualization through XXX in Python.

**Key words：**decision tree, target classification, pruning，……

目录

[1 实验概述 1](#_Toc100852884)

[1.1 实验目的 1](#_Toc100852885)

[1.2 实验内容 1](#_Toc100852886)

[2 实验方案设计 2](#_Toc100852887)

[2.1 总体设计思路与总体架构 2](#_Toc100852888)

[2.2 核心算法及基本原理 2](#_Toc100852889)

[3 实验过程 5](#_Toc100852890)

[3.1 环境说明 5](#_Toc100852891)

[3.2 源代码文件清单 5](#_Toc100852892)

[3.3 实验结果展示 5](#_Toc100852893)

[4 总结 8](#_Toc100852894)

[4.1 实验中存在的问题及解决方案 8](#_Toc100852895)

[4.2 心得体会 8](#_Toc100852896)

[4.3 后续改进方向 8](#_Toc100852897)

[4.4 总结 8](#_Toc100852898)

[5 附录 9](#_Toc100852899)

[5.1 成员分工与自评 9](#_Toc100852900)

# 1 实验概述

## 1.1 实验目的

熟悉和掌握目标分类问题的定义及其主流求解方法，并利用决策树算法求解目标分类问题，理解求解流程和决策顺序。使用任意语言制作相关图形化界面，可视化决策树算法的决策过程、决策结果与决策树结构。

## 1.2 实验内容

1. 采用 Python语言实现基于决策树的目标分类的程序，设计了三种不同的决策树算法： 第一种：ID3,使用信息增益来选择划分属性；第二种：C4.5，使用信息增益率来选择划分属性；第三种：CART,使用“基尼指数”来选择划分属性。

2. 利用预剪枝和后剪枝，对决策树进行优化，避免决策树学习过程中的过拟合问题。

3.

# 2 实验方案设计

## 2.1 总体设计思路与总体架构

程序整体采用求解与图形化界面分离的策略，分别在 tree.py 与 xxx.py 中实现决策树算法求解目标分类问题与gui图形化界面。

## 2.2 核心算法及基本原理

决策树是一种描述对实例进行分类的树形结构。决策树由点和有向边组成。节点有两种类型：内部节点和叶节点。内部节点表示一种特征或者属性，叶节点表示一个分类。构建决策树时通常采用自上而下的方法，在每一步选择一个最好的属性来分裂。 "最好" 的定义是使得子节点中的训练集尽量的纯。不同的算法使用不同的指标来定义"最好"。决策树构建过程中，首先构建根节点，将所有训练数据放在根节点，选择一个最优特征，按照这一特征的取值将训练数据分割为子集，使各个子集有一个当前条件下最好的分类。如果这些子集能被基本正确分类，那么构造叶节点，将对应子集集中到叶节点。如果有子集不能被正确分类，那么就这些子集选择新的最优特征，继续对其进行分割，构建相应的节点。递归进行上述的操作，直到所有训练数据子集均能被正确分类。决策树每次都找不同的切分点，将样本空间逐渐进行细分，最后把属于同一类的空间进行合并，就形成了决策边界，树的层次越深，决策边界的切分就越细，区分越准确，同时也越有可能产生过拟合。本实验采用三种不同的属性划分标准来构建决策树：

·ID3算法是在决策树的各个结点上应用信息增益准则进行特征选择。具体做法是：

a. 从根节点开始，对结点计算所有可能特征的信息增益，选择信息增益最大的特征作为结点的特征，并由该特征的不同取值构建子节点；

b. 对子节点递归地调用以上方法，构建决策树；

c. 直到所有特征的信息增益均很小或者没有特征可选时为止。

·相比 ID3 算法，C4.5 算法更换了特征选择的标准，使用信息增益比进行特征选择。不直接选择增益率最大的候选划分属性，候选划分属性中找出信息增益高于平均水平的属性，这样保证了大部分好的的特征。再从中选择增益率最高的，这又保证了不会出现编号特征这种极端的情况。

·……

前面的算法生成的决策树非常的详细而庞大，每个属性都被详细地加以考虑，决策树的树叶节点所覆盖的训练样本都是“纯”的。因此用这个决策树来对训练样本进行分类的话，它可以100%完美正确地对训练样本集中的样本进行分类，因为决策树本身就是100%完美拟合训练样本的产物。但是，这会带来一个问题，如果训练样本中包含了一些错误，按照前面的算法，这些错误也会完全被决策树学习，导致“过拟合”。 为了避免决策树“过拟合”样本，可以采用剪枝来对决策树进行优化。

·预剪枝是在树的构建过程中，设置一个阈值，使得当在当前分裂节点中分裂前和分裂后的误差超过这个阈值则分裂，否则不进行分裂操作。所有决策树的构建方法，都是在无法进一步降低熵的情况下才会停止创建分支的过程，为了避免过拟合，可以设定一个阈值，熵减小的数量小于这个阈值，即使还可以继续降低熵，也停止继续创建分支。

·后剪枝是在决策树构造完成后进行剪枝。剪枝的过程是对拥有同样父节点的一组节点进行检查，判断如果将其合并，熵的增加量是否小于某一阈值。如果确实小，则这一组节点可以合并成一个节点，其中包含了所有可能的结果。后剪枝删除了一些多余的子树，并用其叶子节点代替。

2.3 图形化界面设计

# 3 实验过程

## 3.1 环境说明

操作系统 windows10

开发语言：Python

开发环境：Python >= 3.0.0

## 3.2 源代码文件清单

## 3.3 实验结果展示

# 4 总结

## 4.1 实验中存在的问题及解决方案

（1）如何有机统一不同特征的不同的取值可能和在推理时的便利实现相结合：Python内建的dict字典类型是一个很好的选择，通过设置特征的不同的可能取值为key值，对应的value为递归返回的结果，若连接的为叶节点，则value为 '0' 或 '1' 表示正负例，若为决策树的内部节点，则储存对应的对象，将查询请求递归处理直到叶节点。

（2）

## 4.2 心得体会

在本次实验中，我们小组成员基于分工协作，完成了基于决策树对目标进行分类的程序和其相应的图形化界面显示。通过本次实验，我们动手实践了决策树算法的编写，加深了对决策树算法的知识，了解到了不同决策树算法之间的差异以及决策树的优化方式。

在本次实验中，我们使用python编写图形化界面，第一次了解如何从用户的角度完善页面，从使用者而非是程序编写者的角度看待功能。为此，我们设计了画布、多种选择栏和输入栏，以及信息显示栏，为了达成更好的交互体验。我们还使用ttkBootstrap 进一步提升页面美观程度，让用户体验更加。

在本次试验中，我们更是了解到沟通协作在团队工作中的重要性，通过组员的积极配合和及时沟通，我们顺利完成本次作业。

## 4.3 后续改进方向

（1）使决策树能学习连续属性，目前只实现了基于离散属性来生成决策树

（2）现实任务中常会遇到不完整样本，需要使决策树能利用有缺失属性值的训练样例进行学习

## 4.4 总结

# 5 附录

## 5.1 成员分工与自评

朱昀玮：

林日中：

陆明奇：决策树算法编写及优化，10/10