



|  |  |
| --- | --- |
| 题 目： | 《模式识别》课程报告 |
| 姓 名： | 李建锋 |
| 院 系： | 计算机学院 |
| 班 级： | 191171-09 |
| 学 号： | 20171000547 |

**2020年6月10日**

目录

[一、 报告题目 3](#_Toc42768281)

[二、 分析问题 3](#_Toc42768282)

[三、 解决问题 3](#_Toc42768283)

[1、 数据 3](#_Toc42768284)

[2、计算初始熵 4](#_Toc42768285)

[1. OutLook 4](#_Toc42768286)

[2. Temperature 4](#_Toc42768287)

[3.Humidity 4](#_Toc42768288)

[4.Wind 4](#_Toc42768289)

[2、 第二次计算 4](#_Toc42768290)

[1. sunny： 4](#_Toc42768291)

[2. Rainy 5](#_Toc42768292)

[3、 决策树 6](#_Toc42768293)

[四、 实验设计 6](#_Toc42768294)

[1、 平台 6](#_Toc42768295)

[2、 算法 6](#_Toc42768296)

[3、 input文件 7](#_Toc42768297)

[4、 运行结果 8](#_Toc42768298)

[五、 结论 8](#_Toc42768299)

[六、 参考文献 8](#_Toc42768300)

# 报告题目

提出问题：一种改进的决策树属性选择度量方法

# 分析问题

决策树的建立，在解决实际问题中是为了更快的找到自己想要的结果。如果一棵树建立的十分繁杂，在求解实际问题中也会比较麻烦。理论上一颗完美的决策树应该是只问一个问题就能解决问题，但是现实中这样的情况比较少，比较符合现实情况的一颗比较好的决策树应该是问比较少的问题就能够解决问题。那么什么样的问题应该放到根节点，我认为是一些离散度比较低的问题，和结果相关性越大的问题，越应该放到底层，先判断根结点，再以根节点为准判断子节点。那么就应该有一个值作为度量，在一些算法里面像ID3等算法将其称为熵，计算公式为Σ-x\*logx，我通过自己的理解对计算熵的方法做出了一点改变，Σx^2/n，x为某一个选项中结果为是和否的比例,n为选项个数，再以第一个为结果分别求后面的熵，熵越高说明越适合放到底层。

# 解决问题

## 数据

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OutLook | Temperature | Humidity | Wind | PlayTennis |
| Sunny | Hot | High | Weak | No |
| Sunny | Hot | High | Strong | No |
| Overcast | Hot | High | Weak | Yes |
| Rainy | Mild | High | Weak | Yes |
| Rainy | Cool | Normal | Weak | Yes |
| Rainy | Cool | Normal | Strong | No |
| Overcast | Cool | Normal | Strong | Yes |
| Sunny | Mild | High | Weak | No |
| Sunny | Cool | Normal | Weak | Yes |
| Rainy | Mild | Normal | Weak | Yes |
| Sunny | Mild | Normal | Strong | Yes |
| Overcast | Mild | High | Strong | Yes |
| Overcast | Hot | Normal | Weak | Yes |
| Rainy | Mild | High | Strong | No |

## 2、计算初始熵

### OutLook

Sunny：13/25

Overcast:1

Rainy:13/25

Outlook:0.68

### 2. Temperature

Hot:1/4

Mild:5/9

Cool:8/5

Temperature:0.47

### 3.Humidity

High:25/49

Normal:37/49

Humidty:0.63

### 4.Wind

Weak:5/8

Strong:1/4

Wind:0.4375

结论：观察得Outlook值最大，所以应该放置在根结点，outlook中overcast可以直接得出结论，所以接下来考虑子结点。也可以大胆猜测子结点有Humidity。

## 第二次计算

### sunny：

1. Temperature:

Hot:1

Mild:1/4

Cool:1

Temperature:0.75

1. Humidity

High:1

Normal:1

Humidity:1

1. Wind:

Weak:5/9

Strong:1/4

Wind:0.40

结论：通过对比可得Humidity最大所以sunny应该放humidity，而且humidity下都是1，所以不用再放置子结点。

### Rainy

1. Temperature

Mild:5/9

Cool:1/4

Temperature:0.4

1. Wind

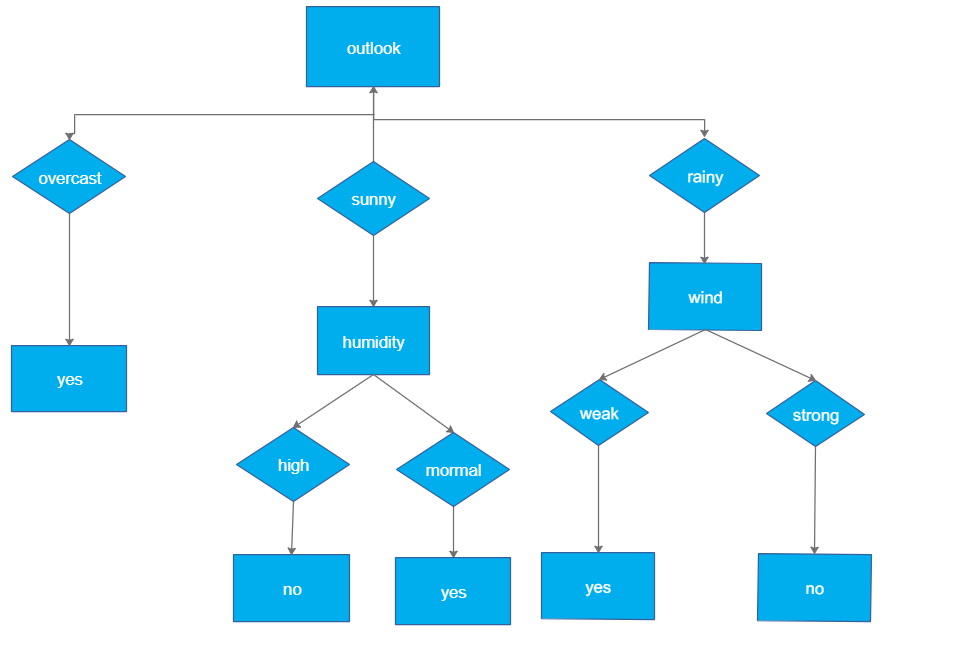
Weak:1

Strong:1

Wind:1

结论：rainy下应该放置wind结点，通过观察可知temperature是一个无关结点，不影响结果。

## 决策树



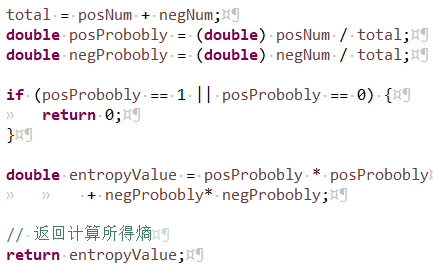
# 实验设计

## 平台

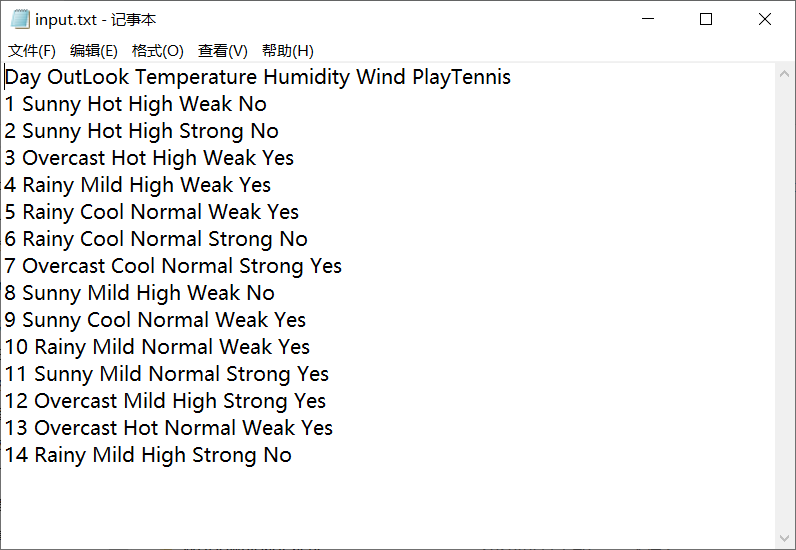
Windows下的eclipse，语言为Java

## 算法

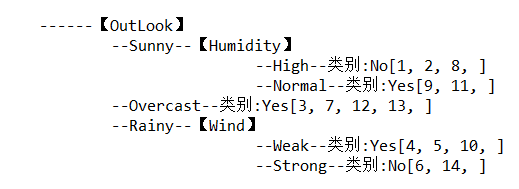
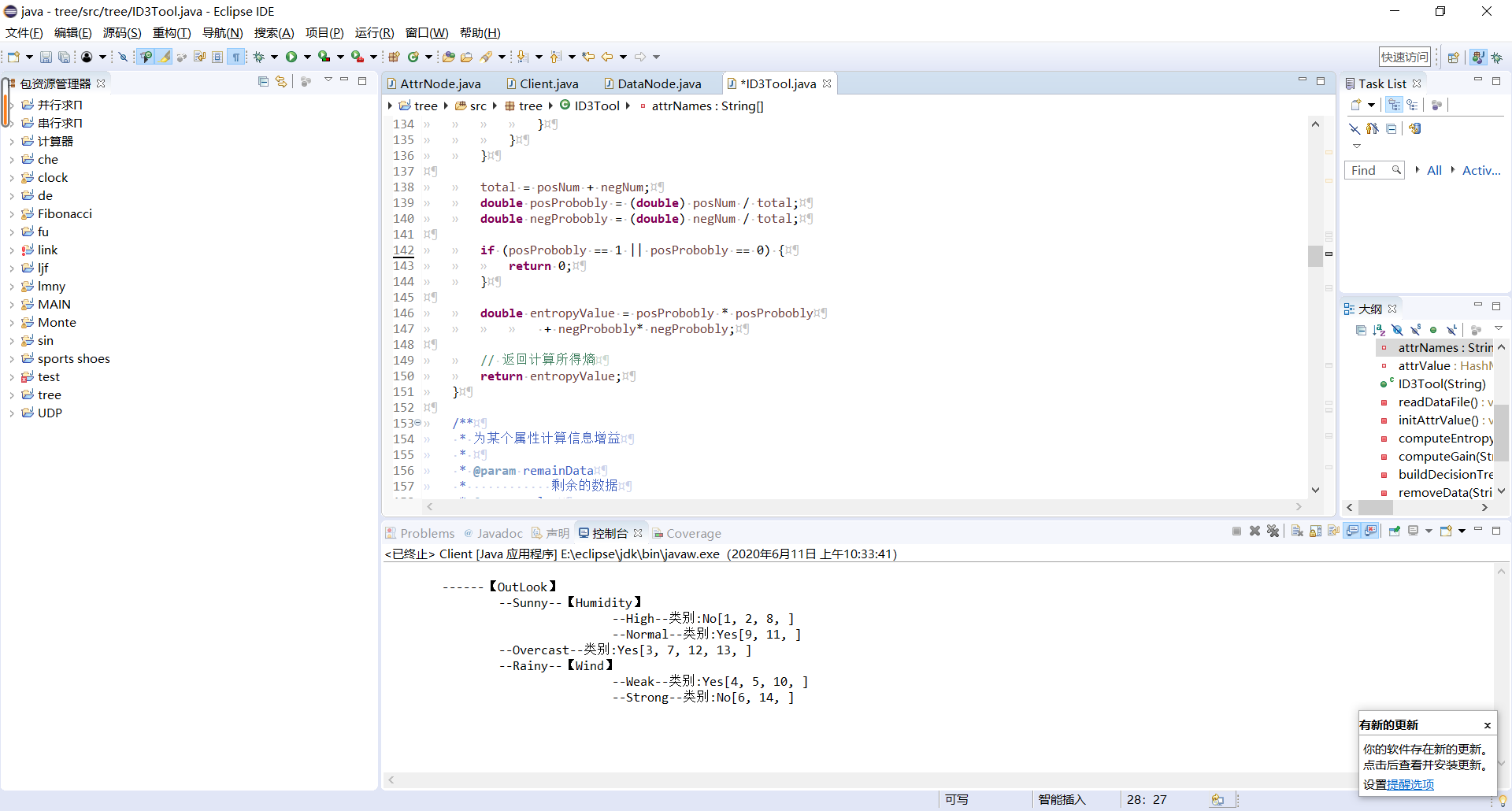
以ID3算法为基础，将计算熵修改为Σx^2/n



## input文件



## 运行结果



# 结论

修改后的结果和原ID3算法一样，也和我自己预测的结果一样。但是是不是对于ID3算法是一种改进，只凭借实验结果是没有办法看出来的，因为结果相同，但是理论上，应该是有所优化。ID3算法是对熵进行差运算来作为标准，再计算的过程中发现，如果一个结点类别比较多的话会对结果造成影响，运用和差消除不了这种影响，如果有无关项也会对ID3算法造成影响。而我的算法在求和后进行了平均，就消除了多项所造成的后果。结点的比较是建立在根结点上也是消除了无关项的影响。

# 参考文献

知网：

1. 基于改进ID3决策树的停电敏感用户辨识方法
2. 一种改进的ID3决策算法及其应用
3. 基于C4.5决策树分类算法的改进与应用
4. 面向大数据分析的决策树算法