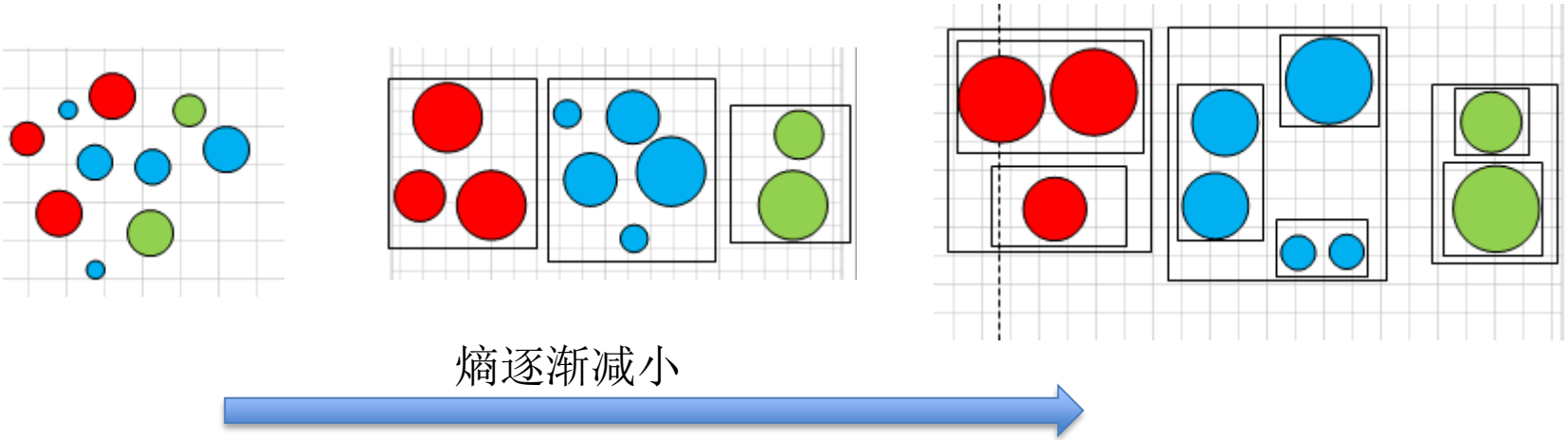
实验五：决策树算法

PPT制作：罗茂权，杨覃娟
出题人　：庞建辉，李祥圣

什么是熵？

- 自然条件之下，事物的混乱程度
- 熵越大，事物的混乱程度越大，
- 熵就会减小，也就是相对的越接近于人工划分（将每一件物品都分好类，同类的放在一起）



决策树

- 预测模型
- 有监督学习模型
- 决策策略：熵（ID3、C4.5）；GINI系数（CART）
- 组成：决策点、枝、结果节点

ID3

- 决策策略：信息增益（Information Gain）
- 步骤：

（1）计算数据集D的经验熵

$$\mathrm{H(D)} = -\sum_{d \in D} p(d) \log p(d)$$

（2）计算特征A对数据集D的条件熵H(D|A)

$$\mathrm{H(D|A)} = \sum_{a \in A} p(a) H(D|A = a)$$

（3）计算信息增益

$$\mathrm{g(D, A)} = \mathrm{H(D)} - \mathrm{H(D|A)}$$

（4）选择信息增益最大的特征作为决策点

ID3计算举例

数据	长鼻子(x)	大耳朵(y)	是否大象(1 or 0)
A1	1	1	1
A2	0	1	0
A3	1	0	0
A4	0	0	0

1. 计算经验熵：

$$H(D)=-1/4*\log(1/4) - 3/4*\log(3/4)$$

2. 计算每个特征下的条件熵：

$$H(D|A=\text{“x”})=(2/4)*(-(1/2)*\log(1/2) - (1/2)*\log(1/2)) + (2/4)*(0 - (2/2)*\log(2/2)))$$

$$H(D|A=\text{“y”})=(2/4)*(-(1/2)*\log(1/2)-(1/2)*\log(1/2)) + (2/4)*0$$

3. 计算：信息增益

$$g(D, A=\text{“x”})=H(D)-H(D|A=\text{“x”})$$

$$g(D, A=\text{“y”})=H(D)-H(D|A=\text{“y”})$$

4. 选择信息增益最大的特征作为决策点

C4.5

- 决策策略：信息增益率（Information Gain Ratio）
- 步骤：

（1）计算特征A对数据集D的信息增益

$$g(D,A) = H(D) - H(D|A)$$

（2）计算数据集D关于特征A的值的熵SplitInfo(D,A)

$$\text{SplitInfo}(D,A) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log\left(\frac{|D_j|}{|D|}\right)$$

（3）计算信息增益率

$$\text{gRatio}(D,A) = (H(D) - H(D|A))/\text{SplitInfo}(D,A)$$

（4）选择信息增益率最大的特征作为决策点

C4.5计算举例

数据	长鼻子(x)	大耳朵(y)	是否大象(1 or 0)
A1	1	1	1
A2	0	1	0
A3	1	0	0
A4	0	0	0

1. 计算在每个特征条件下的信息增益

g(D, A="x")=H(D)-H(D|A="x")； g(D, A="y")=H(D)-H(D|A="y")

2. 计算每个特征的熵

SplitInfo(D,A="x")=-(2/4)*log(2/4)-(2/4)*log(2/4)

SplitInfo(D,A="y")=-(2/4)*log(2/4)-(2/4)*log(2/4)

3. 计算信息增益率

gainRatio(D, A="x")=g(D, A="x")/SplitInfo(D,A="x")

gainRatio(D, A="y")=g(D, A="y")/SplitInfo(D,A="y")

4. 选择信息增益率最大的特征作为决策点

CART

- 决策策略：GINI系数（Gini Index，值越小表示不确定性越小）
- 步骤：

（1）计算特征A的条件下，数据集D的GINI系数

$$\text{gini}(\text{D}, \text{A}) = \sum_{j=1}^{v} p(A_j) \times gini(D_j | A = A_j)$$

其中：

$$gini(D_j | A = A_j) = \sum_{i=1}^{n} p_i(1 - p_i) = 1 - \sum_{i=1}^{n} p_i^2$$

（2）选择GINI系数最小的特征作为决策点

CART计算举例

数据集

数据	长鼻子(x)	大耳朵(y)	是否大象(1 or 0)
A1	1	1	1
A2	0	1	0
A3	1	0	0
A4	0	0	0

1. 计算在每个特征的条件下，数据集的GINI系数

$$gini(D, A=\text{“x”}) = |D_{x=1}|/|D| * gini(D_{x=1}) + |D_{x=0}|/|D| * gini(D_{x=0})$$

$$= (2/4) * [1-(1/2)\ ^2-(1/2)\ ^2] + (2/4) * [1-(2/2)^2-0^2]$$

$$gini(D, A=\text{“y”}) = |D_{y=1}|/|D| * gini(D_{y=1}) + |D_{y=0}|/|D| * gini(D_{y=0})$$

$$= (2/4) * [1-(1/2)\ ^2-(1/2)\ ^2] + (2/4) * [1-1^2-0^2]$$

2. 选择GINI系数最小的特征作为决策点

建树举例（ID3）

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

建树举例（ID3）

- 计算信息增益

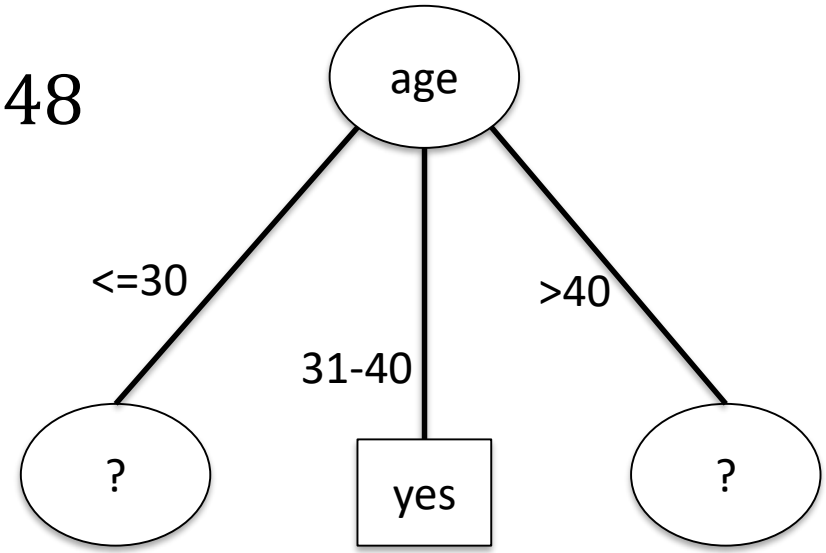
g(D,A="age")=0.246

g(D,A="income")=0.029

g(D,A="student")=0.151

g(D,A="credit_rating")=0.048

- 得到决策点：age
- 下一步？



建树举例（ID3）

- age有三个属性值，三个分枝，分成三个子数据集
- age='<=30'

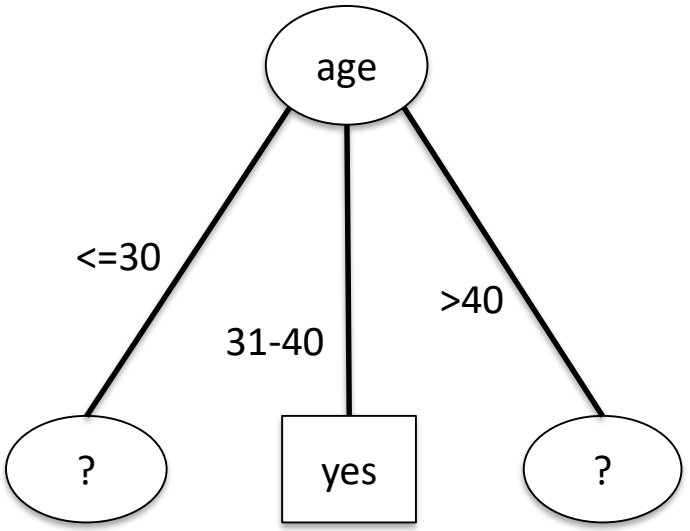
age	income	student	credit_rating	buy_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
<=30	medium	yes	excellent	yes

- age='31-40'

age	income	student	credit_rating	buy_computer
31-40	high	no	fair	yes
31-40	low	yes	excellent	yes
31-40	medium	no	excellent	yes
31-40	high	yes	fair	yes

- age='>40'

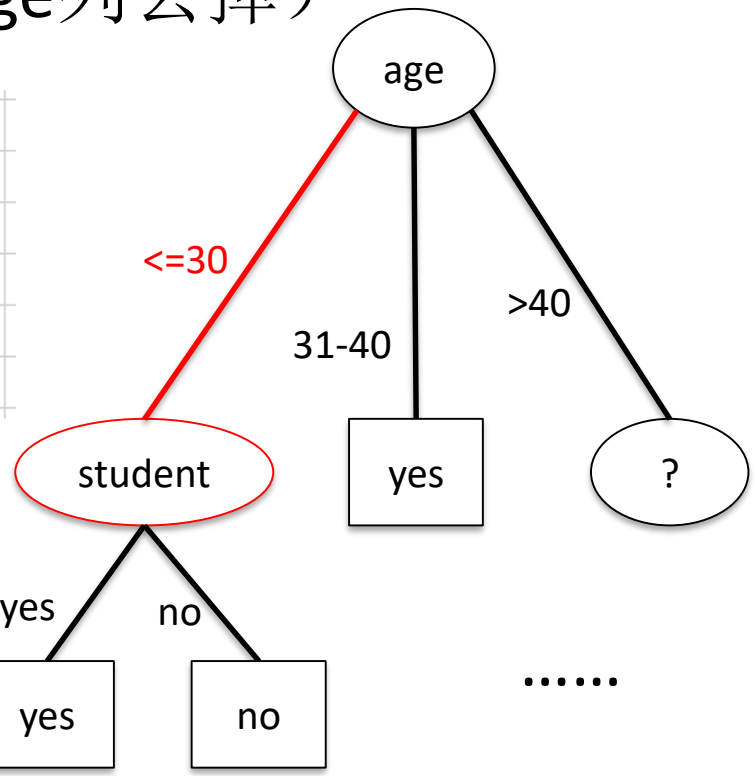
age	income	student	credit_rating	buy_computer
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
>40	medium	yes	fair	yes
>40	medium	no	excellent	no



建树举例（ID3）

- 子数据集 age=“<=30” （要把age列去掉）

income	student	credit_rating	buy_computer
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes



- 计算信息增益

$H(D_{age=“<=30”})=0.971$

$g(D_{age=“<=30”}, A=“income”)=0.571$

$g(D_{age=“<=30”}, A=“student”)=0.971$

$g(D_{age=“<=30”}, A=“credit\_rating”)=0.020$

- 选择特征 “student”

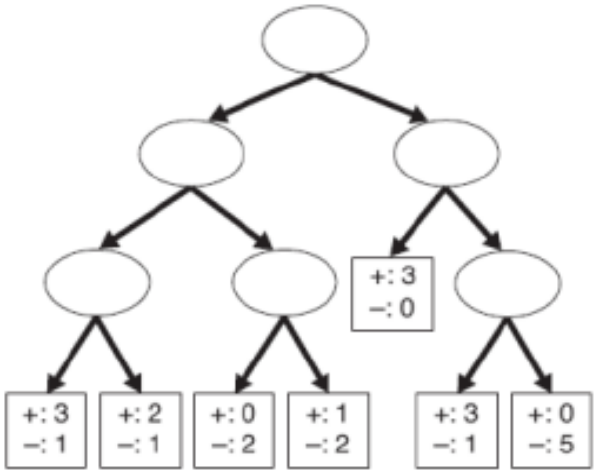
连续数据的划分

- 决策树适合常规性属性值（如晴天、雨天）
- 对于连续数据，需要划分为离散的数据
- 划分多类：每两个相邻值的中位数$(a_i+a_{i+1})/2$作为一个阈值
- 划分两类：取连续属性A中的最大值和最小值的中位数作为一个阈值，分成两类。
- 推荐：根据预训练效果确定划分类数

错误率

- 决策树分类错误的数据个数为a，叶子结点个数为b，一共有n个分类数据，假设一个惩罚值β。那么错误率的计算如下：

$$\varepsilon = \frac{a + b * \beta}{n}$$



若β=0.5，则错误率=(4+7*0.5)/24

剪枝

- **预剪枝(Pre-pruning)**

(1) 错误率剪枝

设定一个决策树的错误率阈值s，根据s决定当前的决策树要不要继续往下分，及早的停止树增长。

(2) 深度剪枝

给定一个决策树深度的阈值d，避免决策树过深。

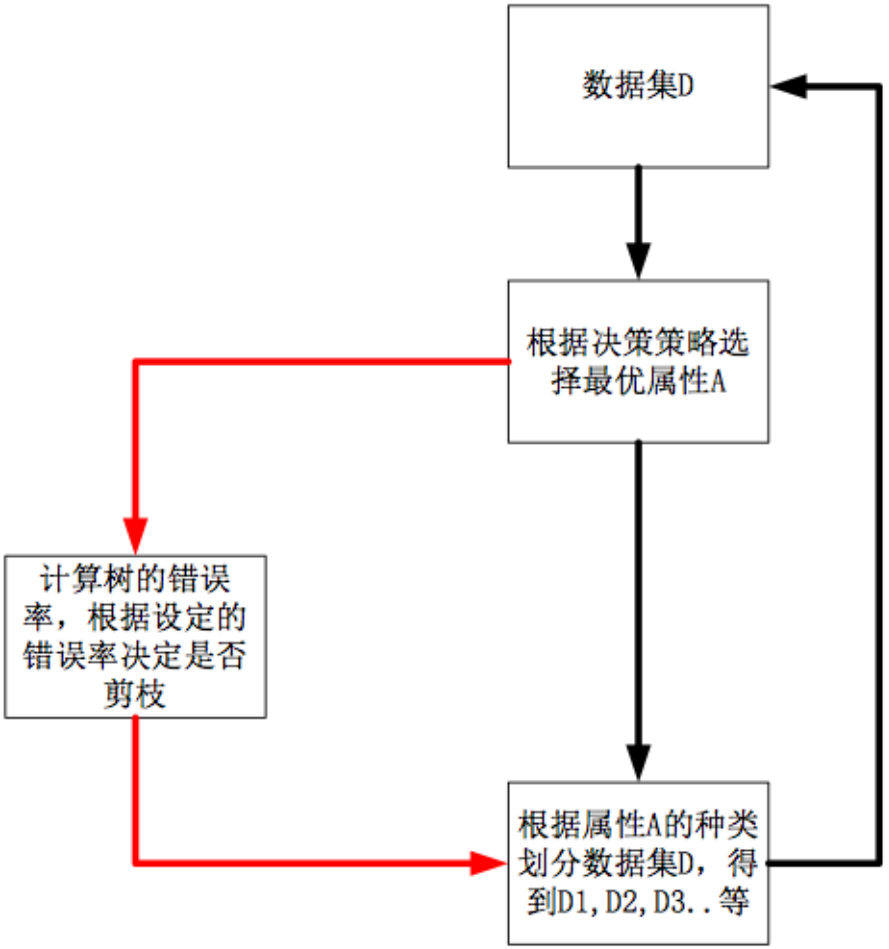
- **后剪枝(Post-pruning)**

(1)错误率降低剪枝

若发现当修剪后的树对于验证集的性能不会比原来的树差时，删除该结点。

代码思路

决策树的实现可以用递归的编程技巧实现。



Python实现举例

- **思路：**使用dict存储整个决策树，用递归构建决策树
- **关键函数：**

(1) chooseBestFeatureToSplit() ,根据策略选择一个最优的属性/特征来划分数据集得到多个数据子集并将子集的该特征维度删去，根据递归的原则，对子集数据集进行同样的处理。

(2) splitDataSet() 根据给定的特征划分数据集。

- **递归结束条件：**

(1) 若分支下所有的实例都具有相同的分类，则得到一个叶节点（结果节点）。

(2) 程序遍历完所有划分数据集的特征，但是类标签依然不是唯一的，采用多数投票的方法决定该叶节点的分类。

参考链接：http://blog.csdn.net/alvine008/article/details/37760639

C++实现举例

- 递归构建决策树

```
node* buildDecisionTree(vvs &table, node* nodePtr, vvs &tableInfo)
{
    //没有待分割的值
    if (tableIsEmpty(table)) {
        return NULL;
    }
    //属性中所有值拥有相同标签
    if (isHomogeneous(table)) {
        nodePtr->isLeaf = true;
        nodePtr->label = table[1][table[1].size()-1];
        return nodePtr;
    } else {
        //得到待分割的属性 名字
        string splittingCol = decideSplittingColumn(table);
        nodePtr->splitOn = splittingCol;
        //返回这个属性的列号
        int colIndex = returnColumnIndex(splittingCol, tableInfo);
        int iii;
        for (iii = 1; iii < tableInfo[colIndex].size(); iii++) {
            node* newNode = (node*) new node;
            newNode->label = tableInfo[colIndex][iii];
            nodePtr->childrenValues.push_back(tableInfo[colIndex][iii]);
            newNode->isLeaf = false;
            newNode->splitOn = splittingCol;
            //基于要分割的属性，修建这个数据table，去除所有数据中的那一列属性元素
            vvs auxTable = pruneTable(table, splittingCol, tableInfo[colIndex][iii]);
            //递归建树
            nodePtr->children.push_back(buildDecisionTree(auxTable, newNode, tableInfo));
        }
    }
    return nodePtr;
}
```

参考链接：http://blog.csdn.net/fy2462/article/details/31762429

数据集

- 实验数据集文件为csv格式，每行数据集是有9个特征和1个标签结果组成。
- train.csv（583个训练数据，前9列是数据特征，最后一列预测结果）；
- test.csv（100个测试数据，同上）；
- 注意特征数据需要离散化

评测指标

1、准确度（Accuracy）

通过训练集训练后，在测试集实现分类预测，求出分类准确度。

2、决策树的错误率（ε）

计算训练出的最终决策树的错误率，惩罚值β默认设置为0.5和1两种，并解释β的意义。

实验任务与提交要求

选做DT分类的需要完成：

1. 参照lab5实验要求
2. 评测指标的实现和分析
3. 实验思路，关键代码截图，并解释
4. 决策树的改进方法（如果有）
5. 实验报告命名：学号_拼音名字.pdf

实验任务与提交要求

提交代码要求：

1. 代码文件命名格式：DT_决策树种类名字.xxx，再对应写一个说明文件：DT_决策树种类名字_readme，说明有没有优化，怎么跑这个代码。
2. 将所有代码文件打包压缩，命名：学号_拼音姓名.zip

实验任务与提交要求

注意事项：

1. 截止日期：2016年11月20日23点59分59秒
   超过视为迟交
2. FTP地址：ftp://my.ss.sysu.edu.cn/~ryh
3. 抄袭，双方均为0分