决策树

黄晟

huangsheng@cqu.edu.cn

办公室：信息大楼B701

程序员的直觉（The Intuition of Programmer）

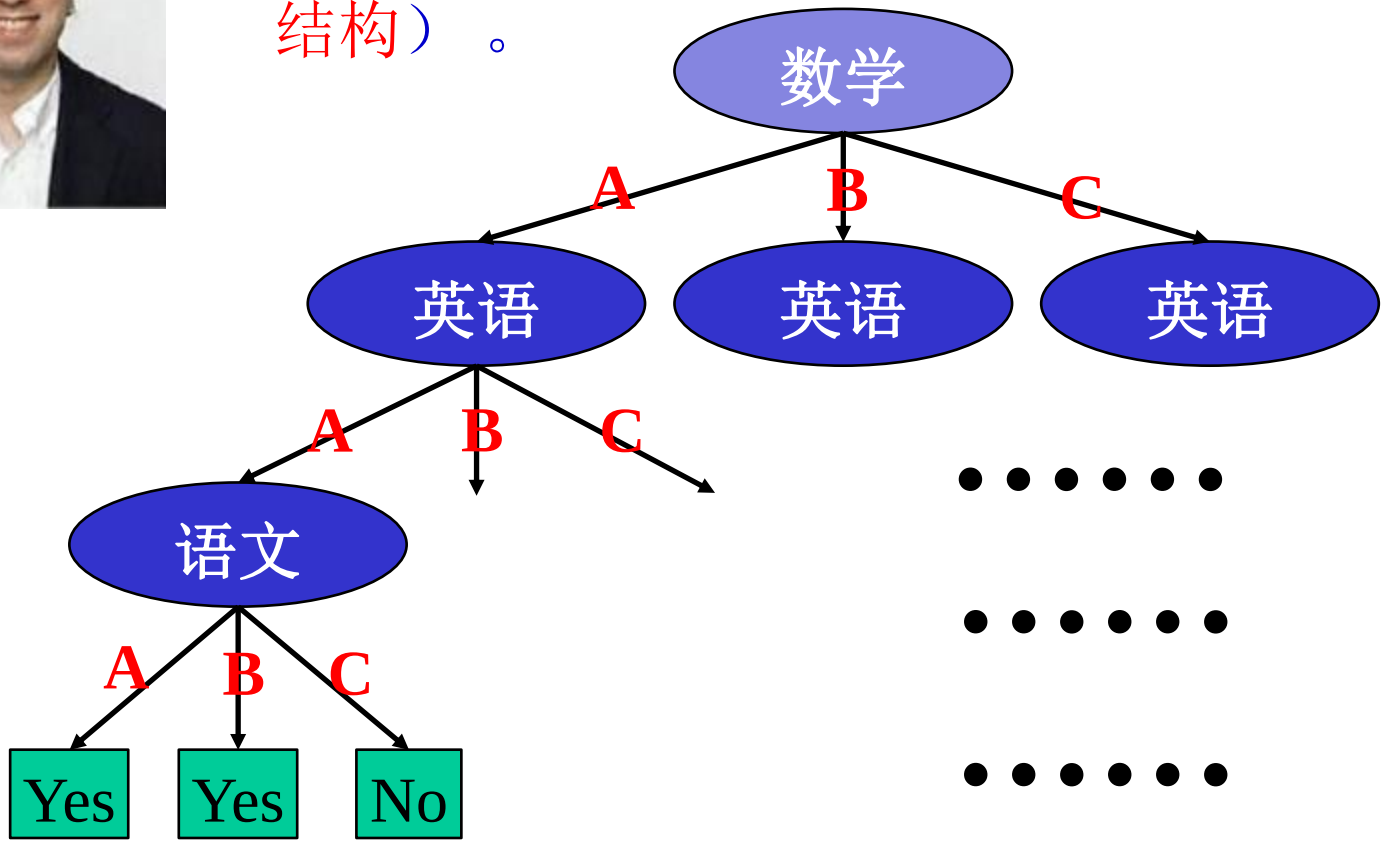
- 实现一个简单程序通过成绩判断他/她能否被录取。

学号	数学	英语	语文	录取
1	A	C	B	Yes
2	A	B	B	Yes
3	A	B	C	No
4	B	B	B	Yes
5	B	C	B	No
6	C	C	B	No
7	C	A	A	Yes

程序员的直觉

- 一般程序员：

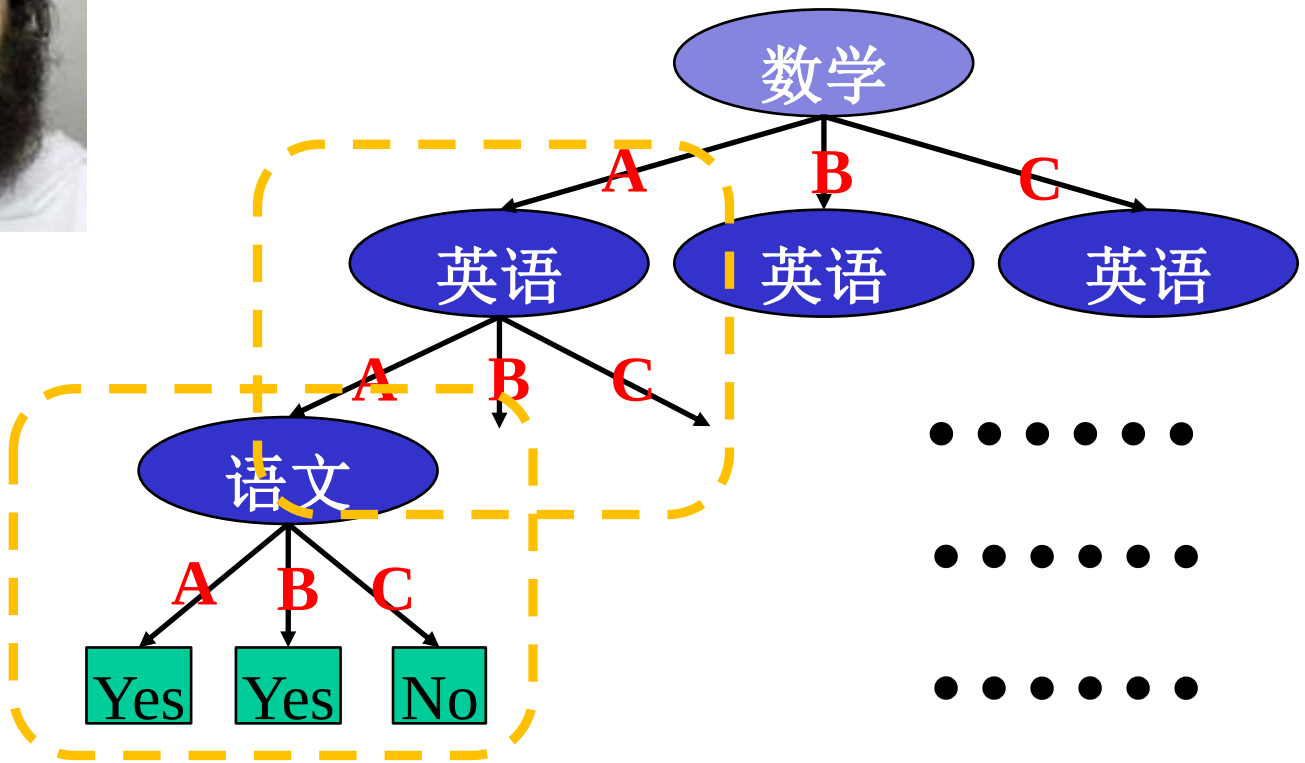
» 绘制程序流程图，然后利用if else或switch case语句进行分支结构的程序实现（树形结构）。



程序员的直觉

- 资深程序员：

– 发现规律，采用‘分而治之’思想，利用递归进行求解。

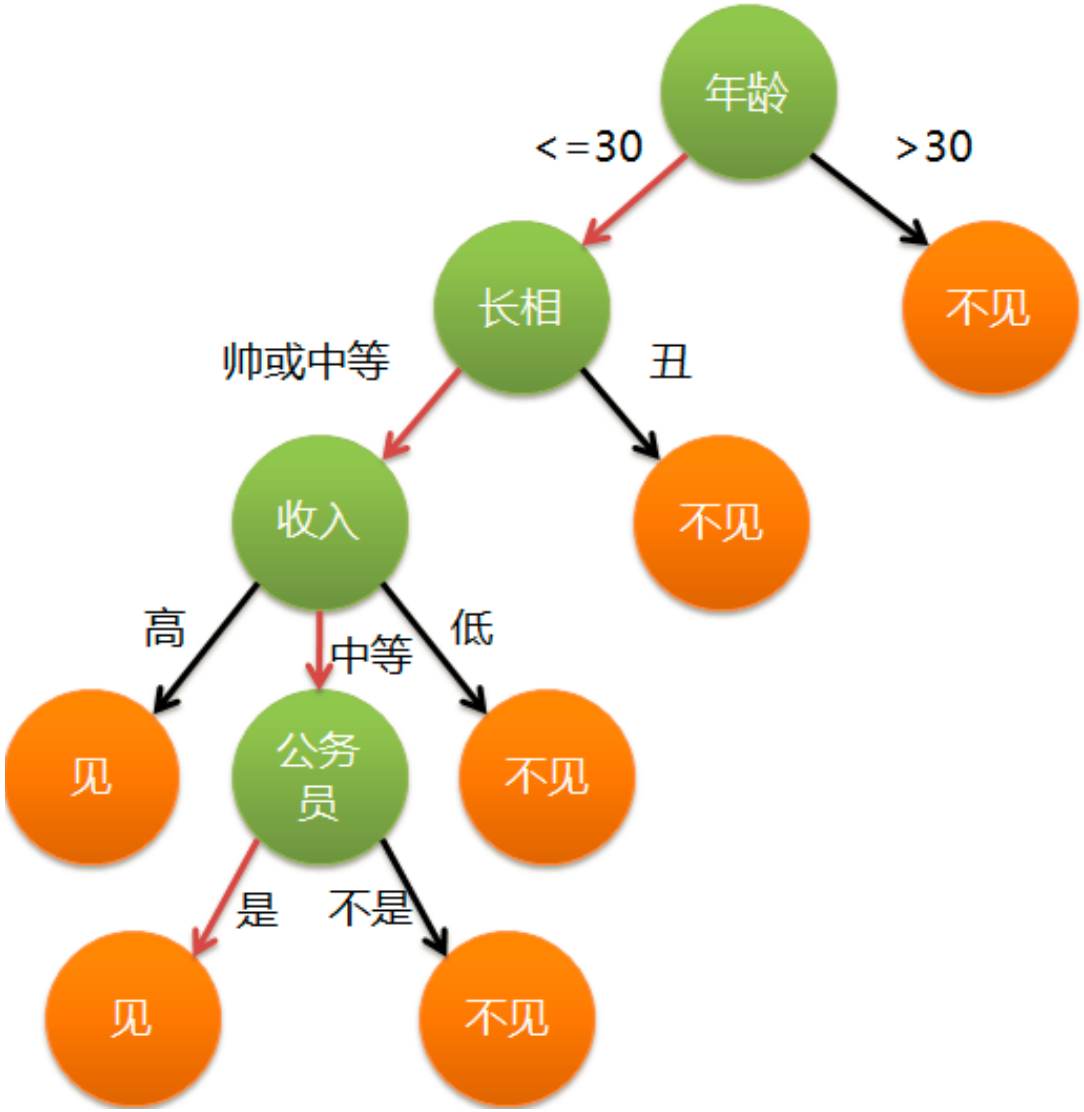


决策树-生活例子

- 相亲——母女对话：
  - 女儿：多大年纪了?
  - 母亲：26。
  - 女儿：长的帅不帅?
  - 母亲：挺帅的。
  - 女儿：收入高不?
  - 母亲：不算很高，中等情况。
  - 女儿：是公务员不?
  - 母亲：是，在税务局上班呢。
  - 女儿：那好，我去见见。

此例子纯属虚构，不代表广大女性同胞的择偶标准。如有雷同纯属巧合。

决策树



决策树（Decision Tree）

- 决策树（decision tree）：构建一个基于属性的树形分类器。
  - 每个非叶节点表示一个特征属性上的测试（分割），
  - 每个分支代表这个特征属性在某个值域上的输出，
  - 每个叶节点存放一个类别。
- 使用决策树进行决策的过程就是从根节点开始，测试待分类项中相应的特征属性，并按照其值选择输出分支，直到到达叶子节点，将叶子节点存放的类别作为决策结果。

决策树

- 决策树构建：分治法思想（递归）
  - 对于当前结点返回递归条件：
    ① 当前结点样本均属于同一类别，无需划分。
    ② 当前属性集为空。
    ③ 所有样本在当前属性集上取值相同，无法划分。
    ④ 当前结点包含的样本集合为空，不能划分。

决策树

- 递归结束条件
  1. 当前结点样本均属于同一类别，无需划分。
     - Example: 下一个要划分的属性为属性1

编号	属性1	类别
1	A	P
2	A	P
3	B	P
4	C	P

决策树

- 递归结束条件
  2. 当前属性集为空。
     - Example: 属性1(B)➔属性2(A)➔属性3(A) 走完该路径已经无属性往下分。

编号1	属性1	属性2	属性3	类别
1	A	C	A	P
2	B	A	A	P
3	B	B	B	N
4	C	C	B	N

决策树

- 递归结束条件

  3.所有样本在当前属性集上取值相同，无法划分。

  - Example: **属性1 B分支下**，样本子集中所有样本属性值完全一样，再往下划分就没有意义了。

编号1	属性1	属性2	属性3	类别
1	A	B	A	P
2	**B**	B	A	P
3	**B**	B	A	N
4	C	C	B	N

决策树

- 递归结束条件

  4.当前结点包含的样本集合为空，不能划分。。

  - Example: **属性1 B分支中 属性2 A分支下**，唯一的属性——属性3，只有在值为A，其余情况样本集合为空。

编号1	属性1	属性2	属性3	类别
1	A	C	A	P
2	B	A	A	P
3	B	B	B	N
4	C	C	B	N

输入： 训练样本集$D = \{(x_1, y_1), (x_2, y_2), \cdots\cdots, (x_n, y_n)\}$;
属性集$A = \{a_1, a_2, \cdots\cdots, a_n\}$

函数 $TreeGenerate(D, A)$:

1. 生成节点node
2. **if** D中样本全属于同一类别C: （1）
3. 将node标记为C类叶节点; **return**;
4. **end if**
5. **if** 属性集A为空或者D的所有属性值均一样: （2）与（3）
6. 将node标记为最多类; **return**;
7. **end if**
8. 从A中选取最佳划分属性a_*;
9. **for** a_*^v in a_*:
10. 为node生成一个分支，令D_v表示D中在a_*属性值为a_*^v的样本子集;
11. **if** D_v为空: （4）
12. 将分支结点标记为叶结点，其类别标签为D中样本最多的类; **return**;
13. **else:**
14. 以$TreeGenerate(D_v, A\backslash\{a_*\})$为分支结点;
15. **end if**
16. **end for**

决策树的核心

- 如何选取最佳划分属性：
  - 极端例子：

编号	属性1	属性2	属性3	标签
1	是	是	是	正
2	否	是	否	负
3	否	是	是	正
4	是	是	否	负
5	是	否	是	正
6	是	否	否	负
7	否	否	是	正

决策树的核心

- 定义最佳划分属性：
  - 经过属性划分后，不同类样本被更好的分离。
  - 理想情况：划分后样本被完美分类。即每个分支的样本都属性同一类。
  - 实际情况：不可能完美划分！尽量使得每个分支某一类样本比例尽量高！即尽量提高划分后子集的纯度（purity）。
- 最佳划分属性目标：
  - 提升划分后子集的纯度
  - 降低划分后子集的不纯度

ID3决策树算法

- 纯度↑=确定性↑=信息量↓
- 度量信息量——信息熵:

$$Ent(D) = -\sum_{k=1}^{|y|} p_k \log_2 p_k$$

  - 信息熵用来度量信息量，信息熵值越小，说明样本集的纯度越高。
- ID3(Iterative Dichotomiser)决策树算法:
  - 利用划分后的**信息增量**来判断属性划分的优劣性。

ID3决策树算法

- 定义关于属性划分后信息熵度量：
  - 假设属性a有V可能取值$\{a^1, a^2, \cdots\cdots, a^V\}$, a^v对应划分后的数据子集为D^v.

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

- 信息增益（Information Gain）：

$$Gain(D,a) = Ent(D) - Ent(D,a)$$

信息增益越大，说明当前划分效果越好：

$$a_* = \underset{a \in A}{\operatorname{argmax}}\, Gain(D,a)$$

决策树（Decision Tree）

Computer Sale 实例

No.	age	income	student	credit	Buyer
1	<30	high	no	fair	no
2	<30	high	no	excellent	no
3	30~40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	30~40	low	yes	excellent	yes
8	<30	medium	no	fair	no
9	<30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<30	medium	yes	excellent	yes
12	30~40	medium	no	excellent	yes
13	30~40	high	yes	fair	yes
14	>40	medium	no	excellent	no

信息熵（Information Entropy）

No.	Buyer
1	no
2	no
3	yes
4	yes
5	yes
6	no
7	yes
8	no
9	yes
10	yes
11	yes
12	yes
13	yes
14	no

Class 1: Buyer = "yes" $\Longrightarrow$ $p_1 = \frac{9}{14}$

Class 2: Buyer = "no" $\Longrightarrow$ $p_2 = \frac{5}{14}$

信息熵（Information Entropy）：

$$\mathbf{Ent}(D) = -\sum_{k=1}^{m} p_k \log_2 p_k$$

$$\mathbf{Ent}(D) = -\left(\frac{9}{14}\log_2\frac{9}{14} + \frac{5}{14}\log_2\frac{5}{14}\right) = 0.9403$$

信息熵（Information Entropy）

No.	age	Buyer
1	<30	no
2	<30	no
3	30~40	yes
4	>40	yes
5	>40	yes
6	>40	no
7	30~40	yes
8	<30	no
9	<30	yes
10	>40	yes
11	<30	yes
12	30~40	yes
13	30~40	yes
14	>40	no

■ **Subset 1: < 30.** $p_1 = \frac{2}{5}$ $p_2 = \frac{3}{5}$

$$\mathrm{Ent}(D^1) = -\left(\frac{2}{5}\log_2\frac{2}{5} + \frac{3}{5}\log_2\frac{3}{5}\right) = 0.9710$$

■ **Subset 2: 30~40.** $p_1 = \frac{4}{4}$ $p_2 = \frac{0}{4}$

$$\mathrm{Ent}(D^2) = -\left(\frac{4}{4}\log_2\frac{4}{4} + \frac{0}{4}\log_2\frac{0}{4}\right) = 0$$

■ **Subset 3: > 40.** $p_1 = \frac{2}{5}$ $p_2 = \frac{3}{5}$

$$\mathrm{Ent}(D^3) = -\left(\frac{2}{5}\log_2\frac{2}{5} + \frac{3}{5}\log_2\frac{3}{5}\right) = 0.9710$$

信息熵（Information Entropy）

No.	age	Buyer
1	<30	no
2	<30	no
3	30~40	yes
4	>40	yes
5	>40	yes
6	>40	no
7	30~40	yes
8	<30	no
9	<30	yes
10	>40	yes
11	<30	yes
12	30~40	yes
13	30~40	yes
14	>40	no

- Subset 1: $\mathrm{Ent}(D^1) = 0.9710$
- Subset 2: $\mathrm{Ent}(D^2) = 0$
- Subset 3: $\mathrm{Ent}(D^3) = 0.9710$

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

信息增益(Information Gain)：

$$\begin{aligned}&\mathrm{Gain}(D, \mathrm{age})\\ &= 0.9403 - \left(\frac{5}{14} \times 0.971 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.971\right)\\ &= 0.2467\end{aligned}$$

信息熵（Information Entropy）

No.	income	Buyer
1	high	no
2	high	no
3	high	yes
4	medium	yes
5	low	yes
6	low	no
7	low	yes
8	medium	no
9	low	yes
10	medium	yes
11	medium	yes
12	medium	yes
13	high	yes
14	medium	no

- **Subset 1: high.** $p_1 = \frac{2}{4}$ $p_2 = \frac{2}{4}$

$$\mathrm{Ent}(D^1) = -\left(\frac{2}{4}\log_2\frac{2}{4} + \frac{2}{4}\log_2\frac{2}{4}\right) = 1$$

- **Subset 2: medium.** $p_1 = \frac{4}{6}$ $p_2 = \frac{2}{6}$

$$\mathrm{Ent}(D^2) = -\left(\frac{4}{6}\log_2\frac{4}{6} + \frac{2}{6}\log_2\frac{2}{6}\right) = 0.9183$$

- **Subset 3: low.** $p_1 = \frac{3}{4}$ $p_2 = \frac{1}{4}$

$$\mathrm{Ent}(D^3) = -\left(\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}\right) = 0.8113$$

信息熵（Information Entropy）

No.	income	Buyer
1	high	no
2	high	no
3	high	yes
4	medium	yes
5	low	yes
6	low	no
7	low	yes
8	medium	no
9	low	yes
10	medium	yes
11	medium	yes
12	medium	yes
13	high	yes
14	medium	no

- Subset 1: $\mathrm{Ent}(D^1) = 1$
- Subset 2: $\mathrm{Ent}(D^2) = 0.9183$
- Subset 3: $\mathrm{Ent}(D^3) = 0.8113$

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

信息增益(Information Gain)：

$$\begin{aligned}&\mathrm{Gain}(D, \mathrm{income}) \\ &= 0.9403 - \left(\frac{4}{14} \times 1 + \frac{6}{14} \times 0.9183 + \frac{4}{14} \times 0.8113\right) \\ &= 0.0291\end{aligned}$$

信息熵（Information Entropy）

No.	student	Buyer
1	no	no
2	no	no
3	no	yes
4	no	yes
5	yes	yes
6	yes	no
7	yes	yes
8	no	no
9	yes	yes
10	yes	yes
11	yes	yes
12	no	yes
13	yes	yes
14	no	no

- **Subset 1: yes.** $p_1 = \frac{6}{7}$ $p_2 = \frac{1}{7}$

$$\mathrm{Ent}(D^1) = -\left(\frac{1}{7}\log_2\frac{1}{7} + \frac{6}{7}\log_2\frac{6}{7}\right) = 0.5917$$

- **Subset 2: no.** $p_1 = \frac{3}{7}$ $p_2 = \frac{4}{7}$

$$\mathrm{Ent}(D^2) = -\left(\frac{3}{7}\log_2\frac{3}{7} + \frac{4}{7}\log_2\frac{4}{7}\right) = 0.9852$$

信息增益(Information Gain)：

$$\mathrm{Gain}(D, \mathrm{student}) = 0.9403 - \left(\frac{7}{14}\times 0.5917 + \frac{7}{14}\times 0.9852\right) = 0.1519$$

信息熵（Information Entropy）

No.	credit	Buyer
1	fair	no
2	excellent	no
3	fair	yes
4	fair	yes
5	fair	yes
6	excellent	no
7	excellent	yes
8	fair	no
9	fair	yes
10	fair	yes
11	excellent	yes
12	excellent	yes
13	fair	yes
14	excellent	no

- **Subset 1: fair.** $p_1 = \frac{6}{8}$ $p_2 = \frac{2}{8}$

$$\mathrm{Ent}(D^1) = -\left(\frac{6}{8}\log_2\frac{6}{8} + \frac{2}{8}\log_2\frac{2}{8}\right) = 0.8113$$

- **Subset 2: excellent.** $p_1 = \frac{3}{6}$ $p_2 = \frac{3}{6}$

$$\mathrm{Ent}(D^2) = -\left(\frac{3}{6}\log_2\frac{3}{6} + \frac{3}{6}\log_2\frac{3}{6}\right) = 1$$

信息增益(Information Gain)：

$$\begin{aligned}&\mathrm{Gain}(D, \mathrm{credit})\\ &= 0.9403 - \left(\frac{8}{14}\times 0.8113 + \frac{6}{14}\times 1\right)\\ &= 0.0481\end{aligned}$$

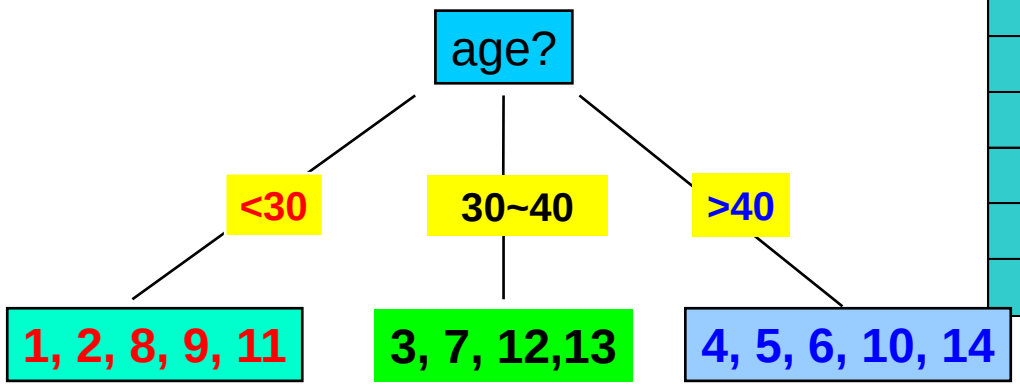
信息熵（Information Entropy）

$\text{Gain}(D, \text{age}) = 0.2467$

$\text{Gain}(D, \text{income}) = 0.0291$

$\text{Gain}(D, \text{student}) = 0.1519$

$\text{Gain}(D, \text{credit}) = 0.0481$



No.	age	Buyer
1	<30	no
2	<30	no
3	30~40	yes
4	>40	yes
5	>40	yes
6	>40	no
7	30~40	yes
8	<30	no
9	<30	yes
10	>40	yes
11	<30	yes
12	30~40	yes
13	30~40	yes
14	>40	no

信息熵（Information Entropy）

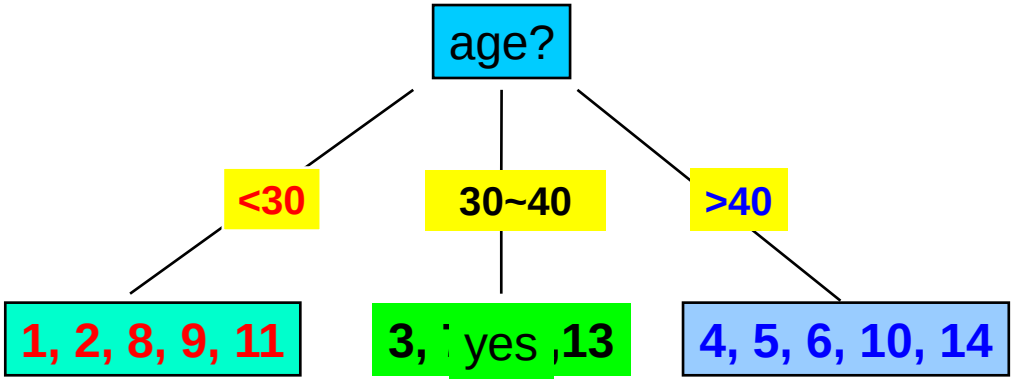
age	Buyer
<30	no
<30	no
30~40	yes
>40	yes
>40	yes
>40	no
30~40	yes
<30	no
<30	yes
>40	yes
<30	yes
30~40	yes
30~40	yes
>40	no

income	Buyer
high	no
high	no
high	yes
medium	yes
low	yes
low	no
low	yes
medium	no
low	yes
medium	yes
medium	yes
medium	yes
high	yes
medium	no

student	Buyer
no	no
no	no
no	yes
no	yes
yes	yes
yes	no
yes	yes
no	no
yes	yes
yes	yes
yes	yes
no	yes
yes	yes
no	no

credit	Buyer
fair	no
excellent	no
fair	yes
fair	yes
fair	yes
excellent	no
excellent	yes
fair	no
fair	yes
fair	yes
excellent	yes
excellent	yes
fair	yes
excellent	no

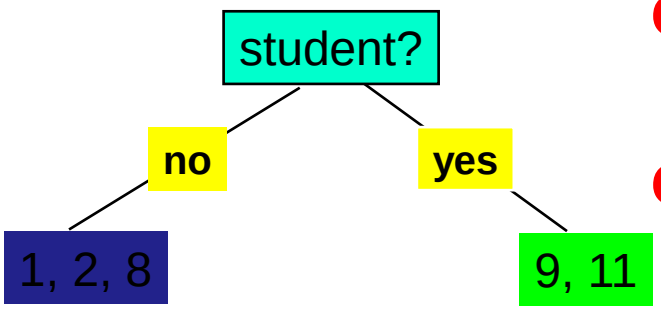
信息熵



- <30,样本仍然有两类，不符合所有递归返回条件，仍然可分，递归继续。
- 30~40， 样本类别均为Yes，满足递归返回条件1， 设为标签为yes的叶节点。
- >40, 样本仍然有两类，不符合所有递归返回条件, 仍然可分，递归继续。

信息熵（Information Entropy）

No.	income	student	credit	Buyer
1	high	no	fair	no
2	high	no	excellent	no
8	medium	no	fair	no
9	low	yes	fair	yes
11	medium	yes	excellent	yes



Gain(D, income) = 0.9710 -0.4=0.5710

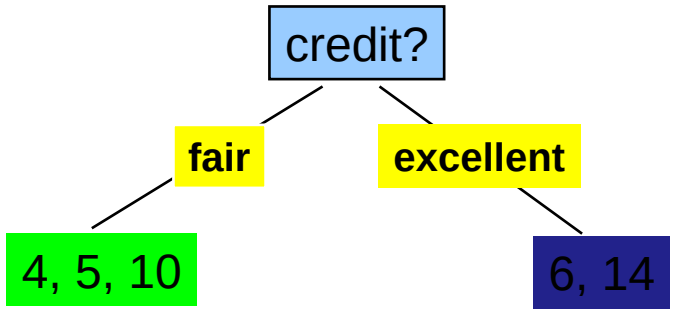
Gain(D, student) = 0.9710-0= 0.9710

Gain(D, credit) = 0.9710 -0.9510=0.02

信息熵（Information Entropy）

No.	income	student	credit	Buyer
4	medium	no	fair	yes
5	low	yes	fair	yes
6	low	yes	excellent	no
10	medium	yes	fair	yes
14	medium	no	excellent	no

Ent(D)=0.9710

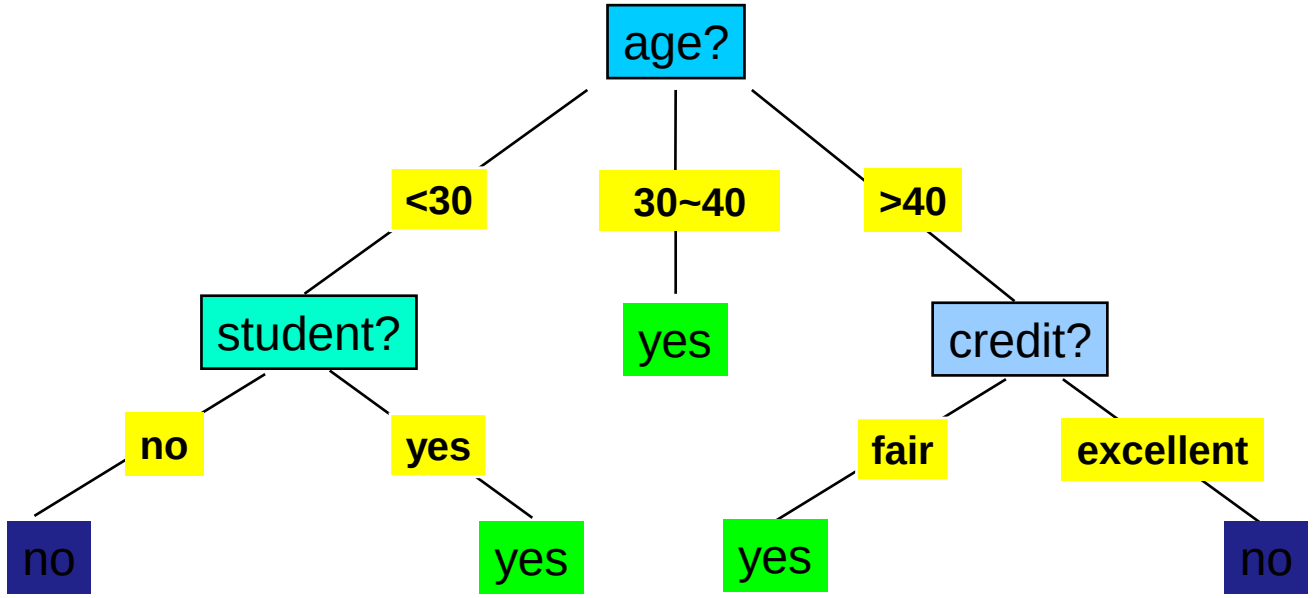


Gain(D, income) = 0.9710 – 0.9510 =0.02

Gain(D, student) = 0.9710 – 0.9510 =0.02

Gain(D, credit) = 0.9710-0=0.9710

信息熵（Information Entropy）



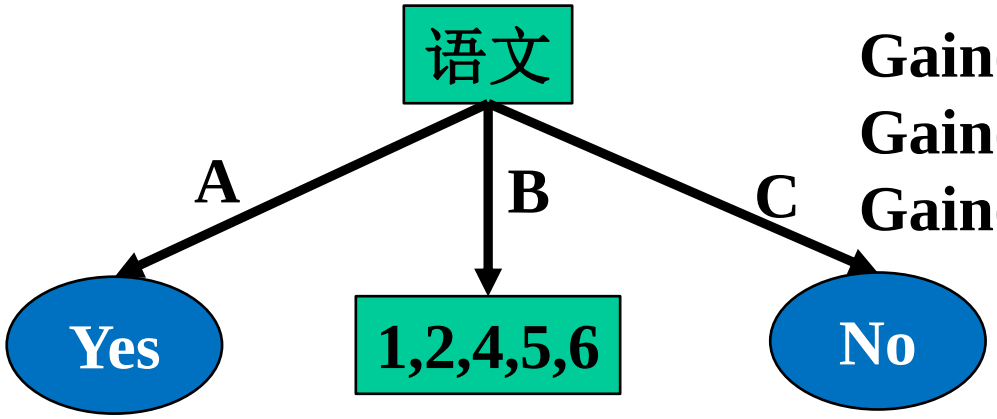
练练手

- 回归开头的例子，动手绘制它的决策树。

学号	数学	英语	语文	录取
1	A	C	B	Yes
2	A	B	B	Yes
3	A	B	C	No
4	B	B	B	Yes
5	B	C	B	No
6	C	C	B	No
7	C	A	A	Yes

Log2(3)=1.5850; log2(5)=2.3219; log2(7)=2.8074;

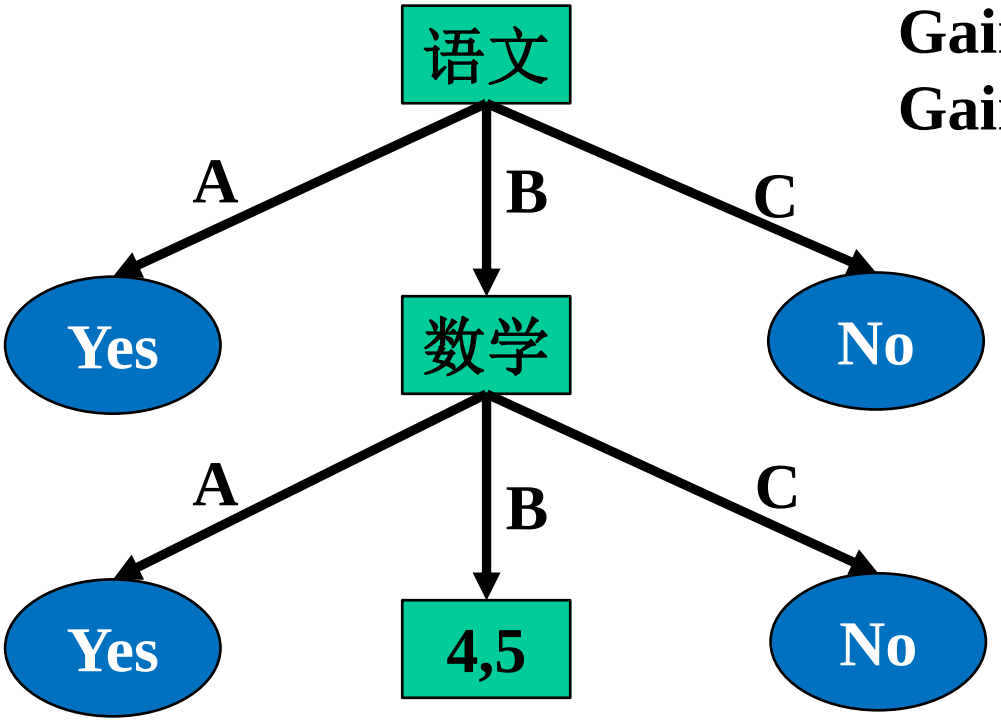
答案



Gain(数学)=0.0202
Gain(英语)=0.1981
Gain(语文)=0.2917

学号	数学	英语	语文	录取
1	A	C	B	Yes
2	A	B	B	Yes
4	B	B	B	Yes
5	B	C	B	No
6	C	C	B	No

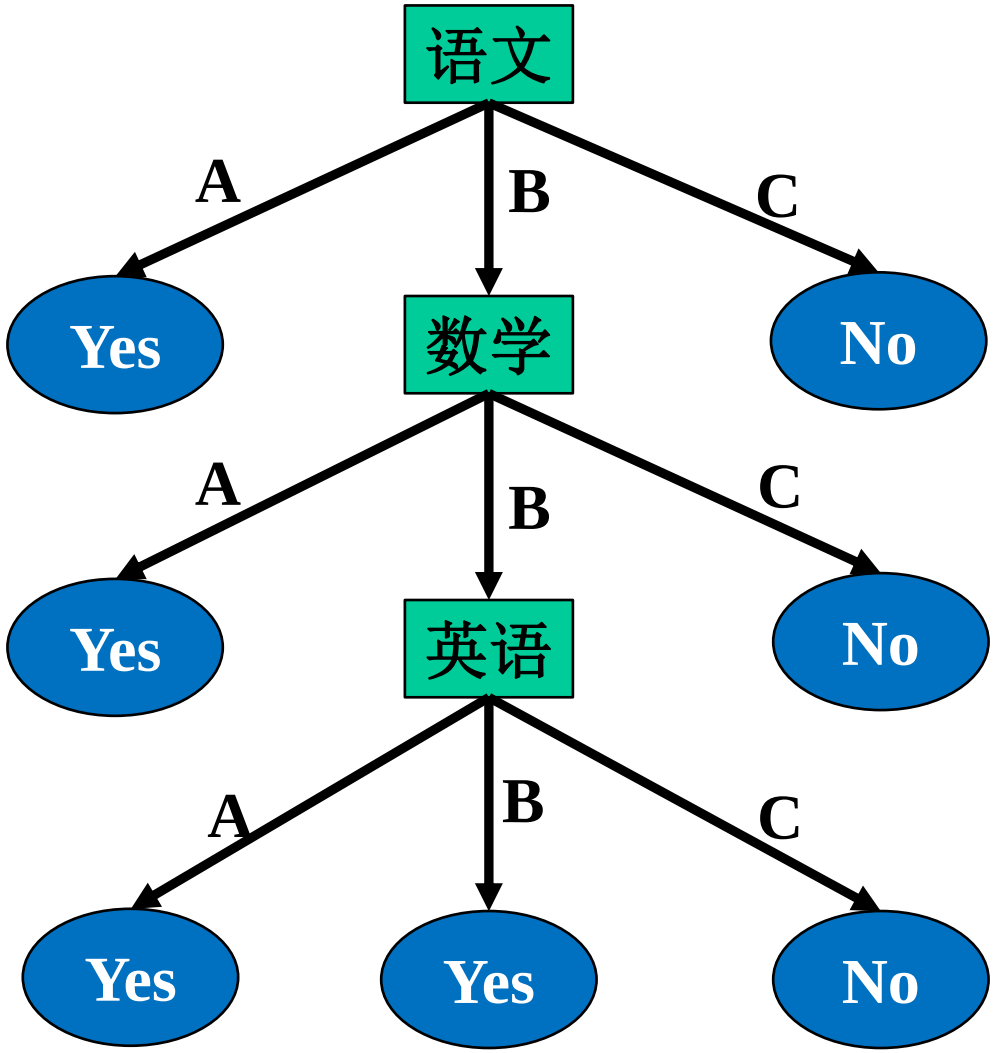
答案



Gain(数学)=0.5710;
Gain(英语)=0.4200;

学号	数学	英语	语文	录取
4	B	B	B	Yes
5	B	C	B	No

答案



信息增量的偏置

- 信息增量准则对可取值数目较多的属性有所偏好。
  - 考虑学号为一个属性
    - ① **Gain(数学)=0.0202**
    - ② **Gain(英语)=0.1981**
    - ③ **Gain(语文)=0.2917**
    - ④ **Gain(学号)=0.9852**
  - **每个学号因为只有一个样本，纯度都很高！**

C4.5决策树算法

- 新准则——增益率（Gain Ratio）

$$\text{Gain\_ratio}(D,a)=\frac{Gain(D,a)}{IV(a)}$$

$IV(a)$称为属性a的“固有值”（Intrinsic Value）

$$IV(a)=-\sum_{v=1}^{V}\frac{|D^v|}{|D|}\log_2\frac{|D^v|}{|D|}$$

- 采用此新准则的方法称为C4.5决策树算法。

① **Gain_ratio(数学)=0.0130**

② **Gain_ratio(英语)=0.1367**

③ **Gain_ratio(语文)=0.2539**

④ **Gain_ratio(学号)= 0.3509**

C4.5算法

age	income	student	credit
<30	high	no	fair
<30	high	no	excellent
30~40	high	no	fair
>40	medium	no	fair
>40	low	yes	fair
>40	low	yes	excellent
30~40	low	yes	excellent
<30	medium	no	fair
<30	low	yes	fair
>40	medium	yes	fair
<30	medium	yes	excellent
30~40	medium	no	excellent
30~40	high	yes	fair
>40	medium	no	excellent

$$\text{Gain\_ratio}(D,a)=\frac{\text{Gain}(D,a)}{IV(a)}$$

$$IV(a)=-\sum_{v=1}^{V}\frac{|D^v|}{|D|}log_2\frac{|D^v|}{|D|}$$

- **Age:** $D^1=5, D^2=4, D^3=5$

$$IV(a)=-\sum_{v=1}^{V}\frac{|D^v|}{|D|}log_2\frac{|D^v|}{|D|}=1.5774$$

$$\text{Gain}_{\text{ratio}(D,a)}=\frac{\text{Gain}(D,a)}{IV(a)}=\frac{0.2467}{1.5774}=0.1564$$

- **Income:** $D^1=4, D^2=6, D^3=4$

$$IV(a)=-\sum_{v=1}^{V}\frac{|D^v|}{|D|}log_2\frac{|D^v|}{|D|}=1.5567$$

$$\text{Gain}_{\text{ratio}(D,a)}=\frac{\text{Gain}(D,a)}{IV(a)}=\frac{0.0291}{1.5567}=0.0187$$

- **Student:** $IV(a)=1$

$$\text{Gain}_{\text{ratio}(D,a)}=\frac{\text{Gain}(D,a)}{IV(a)}=\frac{0.1519}{1}=0.1519$$

- **Credit:** $IV(a)=1.0478$

$$\text{Gain}_{\text{ratio}(D,a)}=\frac{\text{Gain}(D,a)}{IV(a)}=\frac{0.0481}{1.0478}=0.0459$$

CART决策树算法

- CART（Classification And Regression Tree）
- 判断准则——基尼指数（Gini Index）：

$$\mathrm{Gini}(D^v) = 1 - \sum_{k=1}^{|y|} p_k^2$$

$$\mathrm{Gini\_index(D)} = \sum_{v=1}^{|V|} \frac{|D^v|}{|D|} \mathrm{Gini}(D^v)$$

CART算法

age	income	student	credit	Buyer
<30	high	no	fair	no
<30	high	no	excellent	no
30~40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
30~40	low	yes	excellent	yes
<30	medium	no	fair	no
<30	low	yes	fair	yes
>40	medium	yes	fair	yes
<30	medium	yes	excellent	yes
30~40	medium	no	excellent	yes
30~40	high	yes	fair	yes
>40	medium	no	excellent	no

$$\mathrm{Gini(D)} = \sum_{k=1}^{m}\sum_{k'\neq k} p_k p_{k'} = 1 - \sum_{k=1}^{|y|} p_k^2$$

$$\mathrm{Gini\_index(D)} = \sum_{v=1}^{|V|} \frac{|D^v|}{|D|}\mathrm{Gini}(D^v)$$

- **Age:** $D^1 = 5, D^2 = 4, D^3 = 5$

$$\mathbf{Gini}(D^1) = 1 - \sum_{k=1}^{|y|} p_k^2 = 0.48$$

$$\mathbf{Gini}(D^2) = 1 - \sum_{k=1}^{|y|} p_k^2 = 0$$

$$\mathbf{Gini}(D^3) = 1 - \sum_{k=1}^{|y|} p_k^2 = 0.48$$

$$\mathrm{Gini_{index(D,}}a) = \sum_{v=1}^{|V|} \frac{|D^v|}{|D|}\mathrm{Gini}(D^v)$$

$$= \frac{5}{14} \times 0.48 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.48 = 0.3286$$

- **Income:** $D^1 = 4, D^2 = 6, D^3 = 4$

$$\mathbf{Gini}(D^1) = 1 - \sum_{k=1}^{|y|} p_k^2 = 0.5$$

$$\mathbf{Gini}(D^2) = 1 - \sum_{k=1}^{|y|} p_k^2 = 0.4444$$

$$\mathbf{Gini}(D^3) = 1 - \sum_{k=1}^{|y|} p_k^2 = 0.48$$

$$\mathrm{Gini_{index(D,}}a) = \frac{4}{14} \times 0.5 + \frac{6}{14} \times 0.4444 + \frac{4}{14} \times 0.48 = 0.4405$$

CART算法

age	income	student	credit	Buyer
<30	high	no	fair	no
<30	high	no	excellent	no
30~40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
30~40	low	yes	excellent	yes
<30	medium	no	fair	no
<30	low	yes	fair	yes
>40	medium	yes	fair	yes
<30	medium	yes	excellent	yes
30~40	medium	no	excellent	yes
30~40	high	yes	fair	yes
>40	medium	no	excellent	no

$$\mathrm{Gini(D)} = \sum_{k=1}^{m}\sum_{k'\neq k} p_k p_{k'} = 1 - \sum_{k=1}^{|y|} p_k^2$$

$$\mathrm{Gini\_index(D)} = \sum_{v=1}^{|V|} \frac{|D^v|}{|D|}\mathrm{Gini}(D^v)$$

- **Student:** $D^1 = 7, D^2 = 7$

$$\mathbf{Gini}(D^1) = 1 - \sum_{k=1}^{|y|} p_k^2 = 0.2449$$

$$\mathbf{Gini}(D^2) = 1 - \sum_{k=1}^{|y|} p_k^2 = 0.4898$$

$$\mathrm{Gini}_{\mathrm{index(D},a)} = \sum_{v=1}^{|V|} \frac{|D^v|}{|D|}\mathrm{Gini}(D^v)$$

$$= \frac{7}{14} \times 0.2449 + \frac{7}{14} \times 0.4898 = 0.3673$$

- **Credit:** $D^1 = 6, D^2 = 8$

$$\mathbf{Gini}(D^1) = 1 - \sum_{k=1}^{|y|} p_k^2 = 0.5$$

$$\mathbf{Gini}(D^2) = 1 - \sum_{k=1}^{|y|} p_k^2 = 0.375$$

$$\mathrm{Gini}_{\mathrm{index(D},a)} = \frac{6}{14} \times 0.5 + \frac{8}{14} \times 0.375 = 0.4286$$

连续值处理

动机：利用决策树解决连续属性分类问题。

方法： 连续属性离散化（二分法）。

假设连续属性 a 在数据集上出现n个不同的取值$\{a^1, a^2, \ldots, a^n\}$。

定义候选划分点集合：

$$T_a = \left\{ \frac{a^i + a^{i+1}}{2} \mid \quad 1 \leq i \leq n-1 \right\}$$

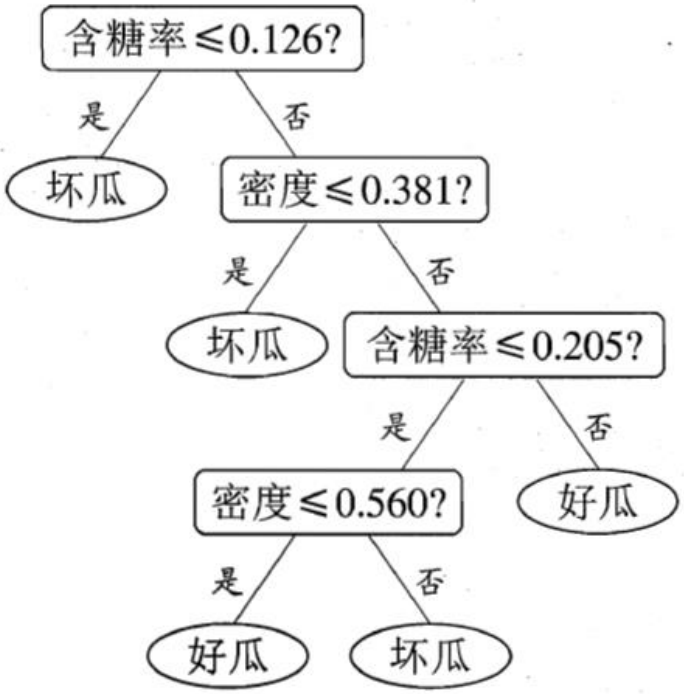
连续值处理

$$\mathrm{Gain}(D,a) = \max_{t \in T_a} \mathrm{Ent}(D) - \sum_{\lambda \in \{-,+\}} \frac{|D_t^{\lambda}|}{|D|} \mathrm{Ent}(D_t^{\lambda})$$

其中D_t^+包含所有在属性a上取值大于 t 的样本，而D_t^-包含所有在属性a上取值小于 t 的样本。

注意：和离散情况不同，属性a划分完之后还可作为后代结点的划分属性。

决策树示意图



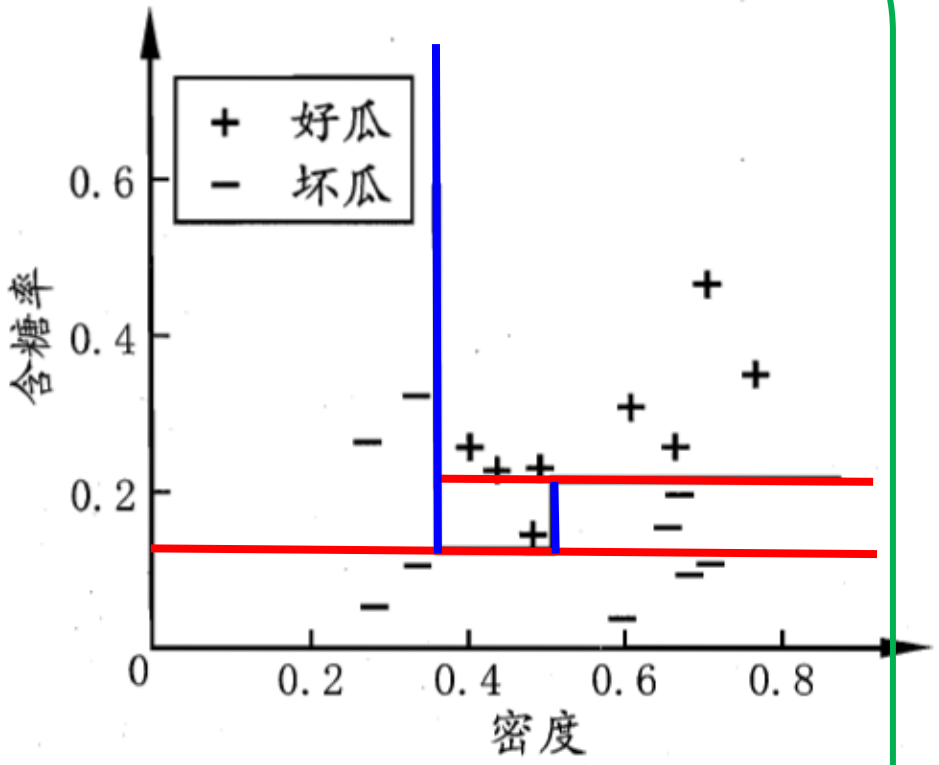


图 4.10 在西瓜数据集 3.0α 上生成的决策树

决策树示意图

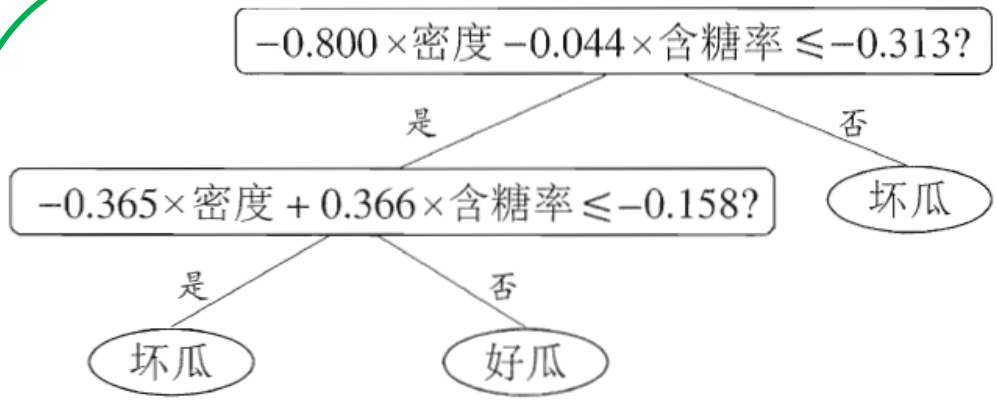


图 4.13 在西瓜数据集 3.0α 上生成的多变量决策树

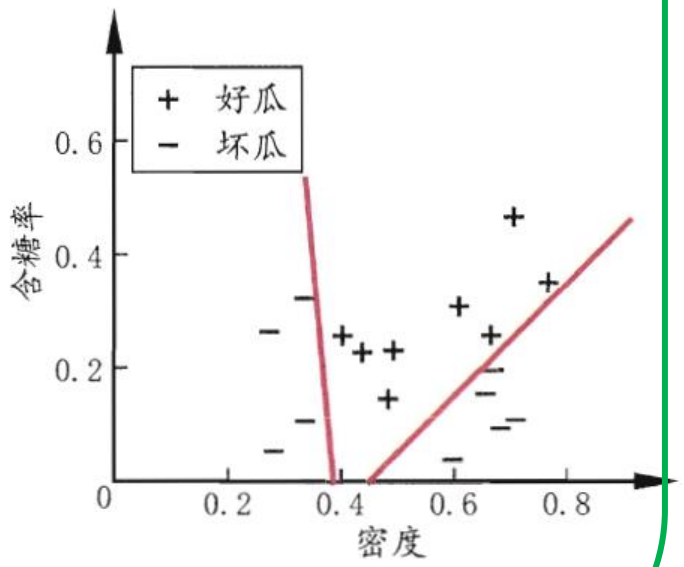


图 4.14 图 4.13 多变量决策树对应的分类边界

决策树的裁剪

- 剪枝(Pruning)处理——避免训练过拟合。
  - 预剪枝(pre-pruning)
    - 预剪枝是指在决策树生成过程中，对每个结点在划分前后进行估计，若当前结点划分不能提升决策树泛化性能，则进行裁剪，把结点标记为叶结点。
  - 后剪枝(post-pruning)
    - 后剪枝是在生成一颗完整的决策树后，对非叶结点自底向上地对非叶结点进行考察，若将该结点对应的子树被替换为叶节点能提升决策树泛化能力，则进行裁剪。

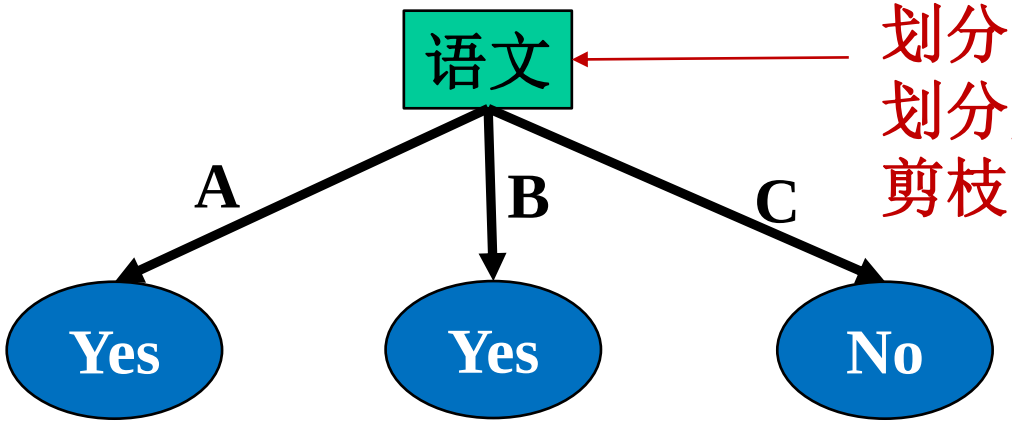
预剪枝

学号	数学	英语	语文	录取
1	A	C	B	Yes
2	A	B	B	Yes
3	A	B	C	No
4	B	B	B	Yes
5	B	C	B	No
6	C	C	B	No
7	C	A	A	Yes
8	A	A	C	No
9	B	B	C	No
10	B	B	A	Yes
11	B	A	B	Yes

验证集

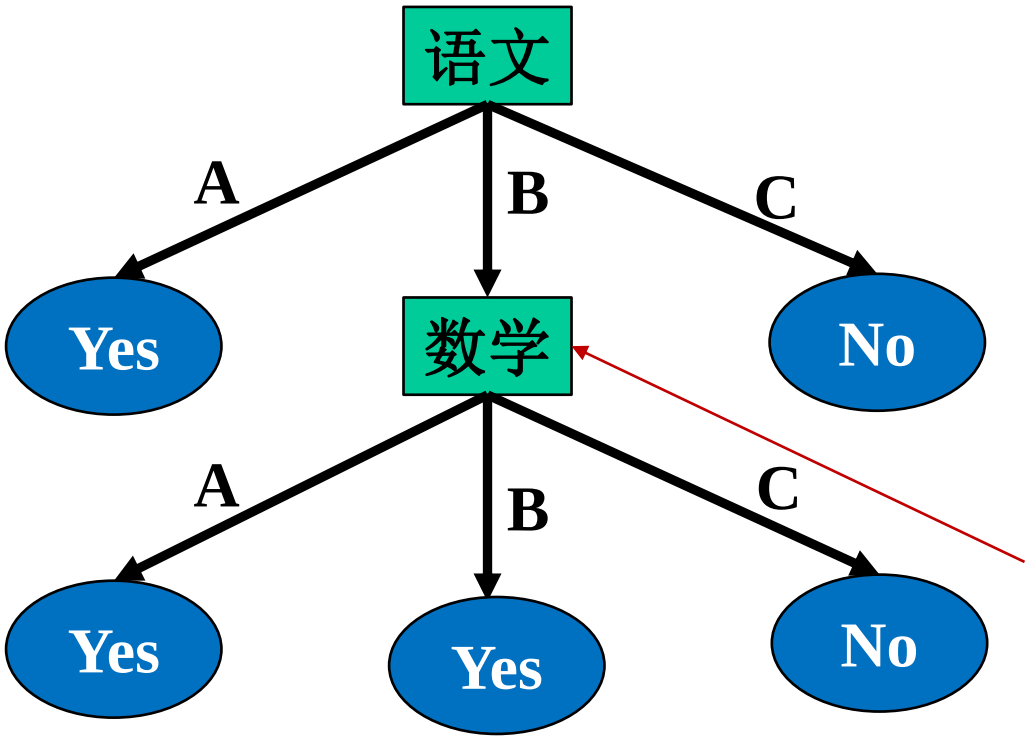
预剪枝

验证集精度：
划分前：50%
划分后：100%
剪枝决策：划分



学号	数学	英语	语文	录取
8	A	A	C	No
9	B	B	C	No
10	B	B	A	Yes
11	B	A	B	Yes

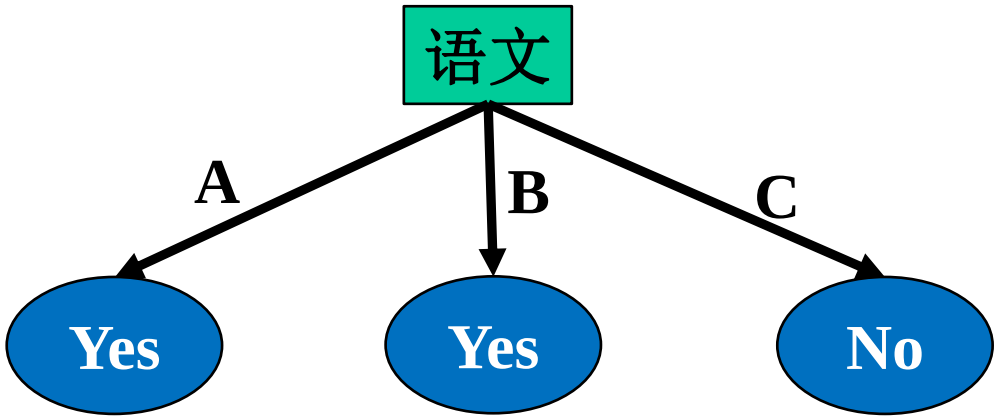
预剪枝



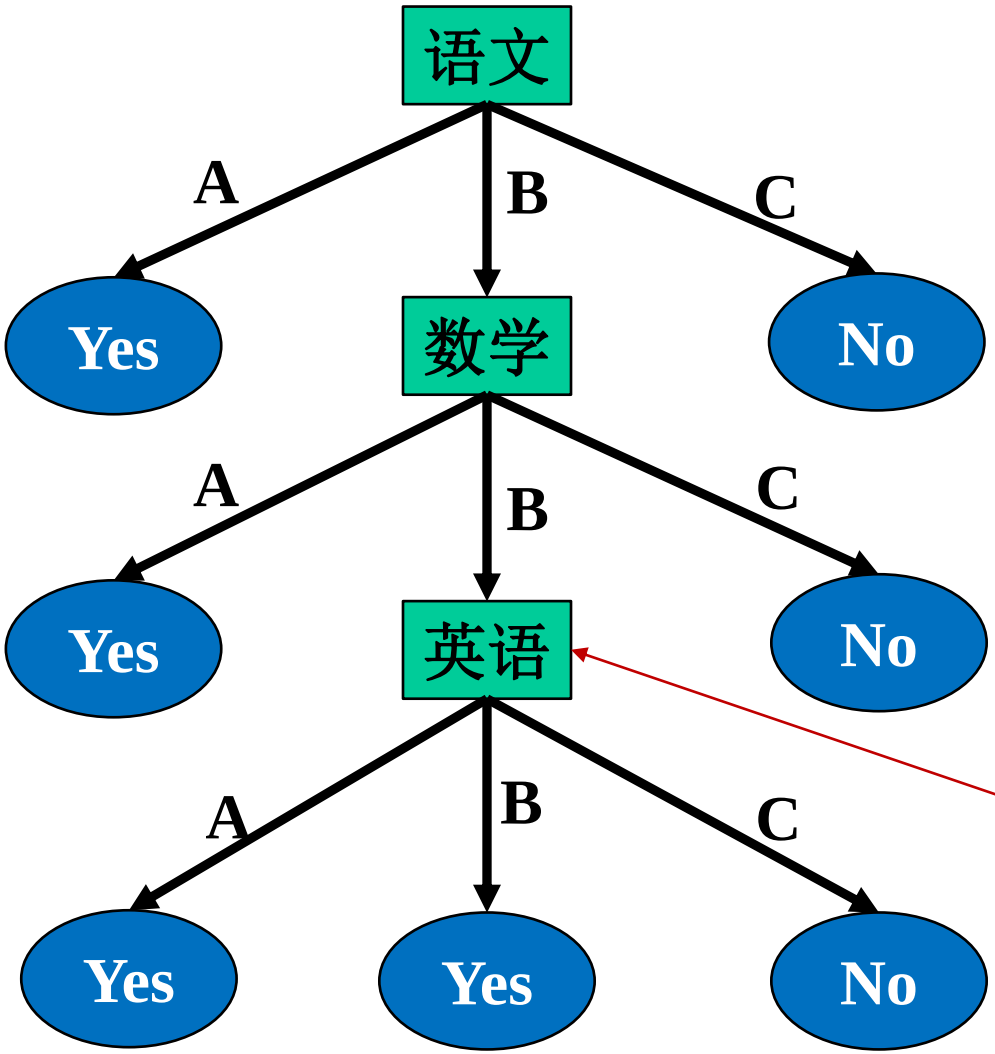
验证精度：
划分前：100%
划分后：100%
剪枝决策：禁止划分

预剪枝结果

- 最后结果-决策树桩(decision Stump)

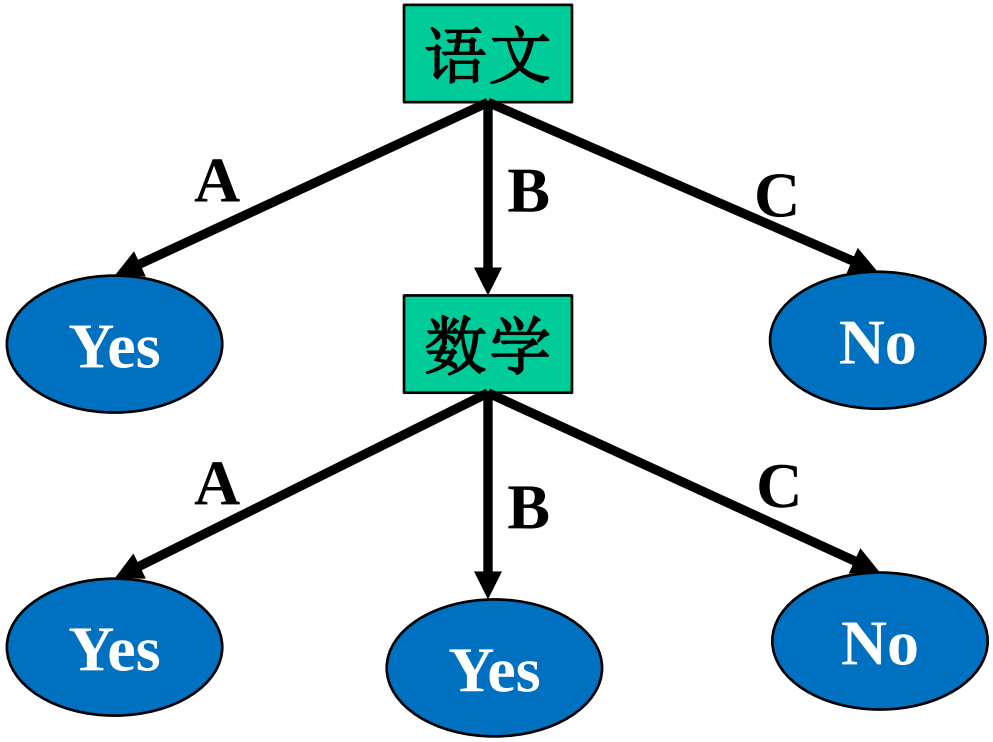


后剪枝



验证精度：
划分前：100%
划分后：100%
剪枝决策：剪枝

后剪枝结果



决策树的裁剪

- 剪枝策略分析：
- 预剪枝：
  - 优点：减少属性划分与测试时间开销。
  - 缺点：可能造成欠拟合。
- 后剪枝：
  - 优点：减少欠拟合风险！
  - 缺点：时间开销大。

思考

- 决策树ID3算法能不能进一步优化？
  - $Gain(D,a) = \boldsymbol{Ent(D)} - Ent(D,a)$ 减少开销
- 如何设计自己决策树算法？
  - 设计自己的划分属性优劣性目标函数。
- 现有决策树框架的分类是否是全局最优？
  - 决策树采用贪心法则，只得局部最优。
- 如何提升决策树算法框架的性能？
  - 融入随机性，提升泛化能力，Example：随机森林。

扩展阅读

- 随机森林(Random Forest)
  - ① 利用自助法(Boostrap)随机采样, 反复随机采用多次，构建多个训练集。
  - ② 利用多个训练集分别训练多个决策树，从而构成决策森林(Decision Forest)
  - ③ 测试样本的标签通过多个决策树输出结果投票(Voting)决定。

参考文献：

Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.

决策树小节

- 决策树构建-递归算法（重点）
- 划分属性优劣度量（重点）
  - 信息增益——ID3算法
  - 增益率 ——C4.5算法
  - 基尼指数——CART算法
- 决策树的裁剪（掌握）
  - 预剪枝
  - 后剪枝
- 决策树连续值与缺失值处理（自学）
- 进阶阅读： 多变量决策树与随机森林

课后习题

- 阅读周志华《机器学习》第四章决策树的内容，掌握其列举的例子。
- 尝试利用C4.5与CART算法重新根据ppt中computer sale例子构建决策树。
- 用任意语言实现任意一个决策树算法，并在教材表4.1《西瓜数据集2.0》上进行测试。