

機器學習

Lecture 5 Decision Tree (決策樹)

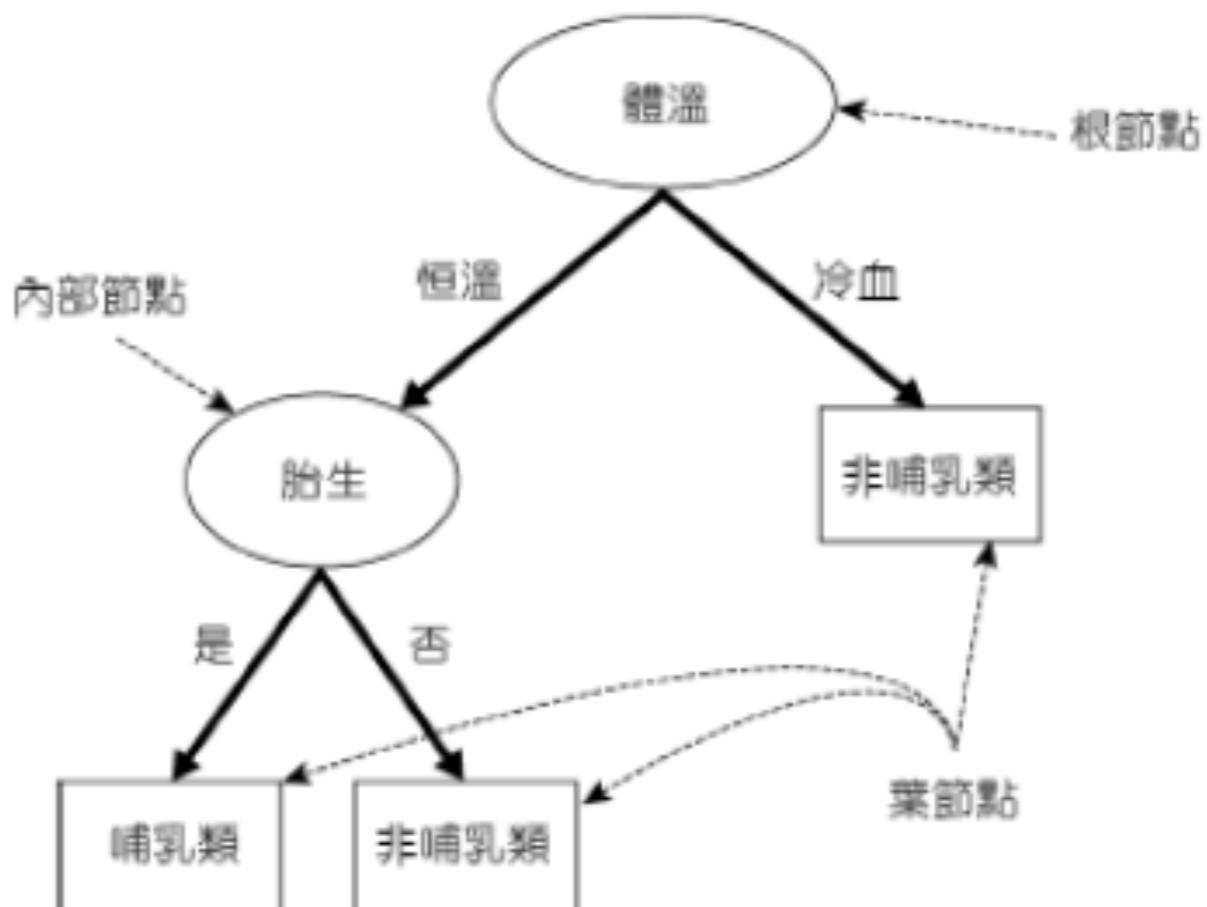
分類：決策樹

● 什麼是決策樹(Decision Tree)？

- 用來處理分類問題的**樹狀結構**
- 每個內部節點表示一個**評估欄位**
- 每個分枝代表一個可能的**欄位輸出結果**
- 每個樹葉節點代表不同分類的**類別標記**

● 決策樹範例

- 哺乳類動物分類的問題
(鴨嘴獸和食蟻獸不適用!!)

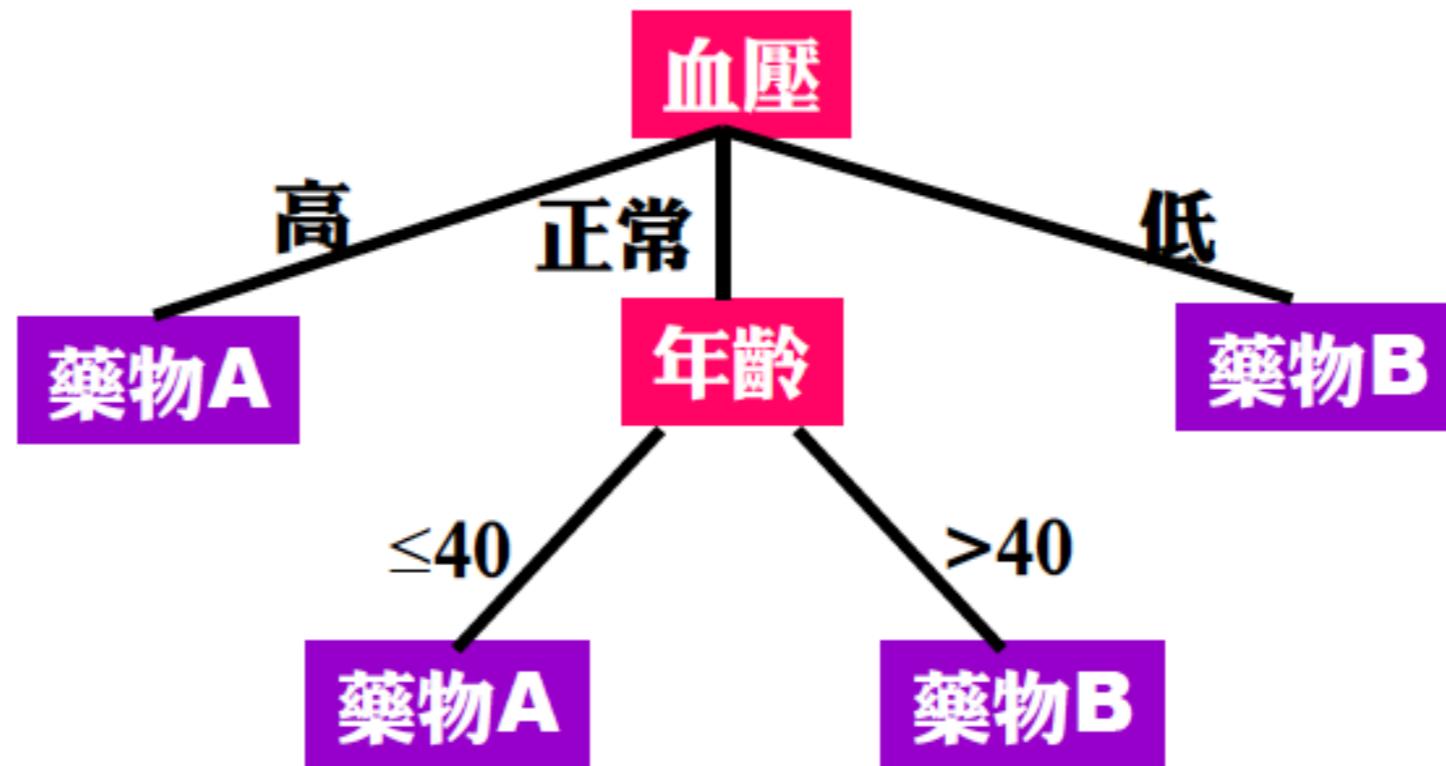


醫療資訊上的應用

編號	性別	年齡	血壓	藥物
1	男	20	正常	A
2	女	73	正常	B
3	男	37	高	A
4	男	33	低	B
5	女	48	高	A
6	男	29	正常	A
7	女	52	正常	B
8	男	42	低	B
9	男	61	正常	B
10	女	30	正常	A
11	女	26	低	B
12	男	54	高	A

類別欄位

醫療資訊上的應用



- 上述決策樹可看出以下規則：

- 如果血壓高，則採用藥物A
- 如果血壓低，則採用藥物B
- 如果血壓正常且年齡 ≤ 40 ，則採用藥物A
- 如果血壓正常且年齡 > 40 ，則採用藥物B

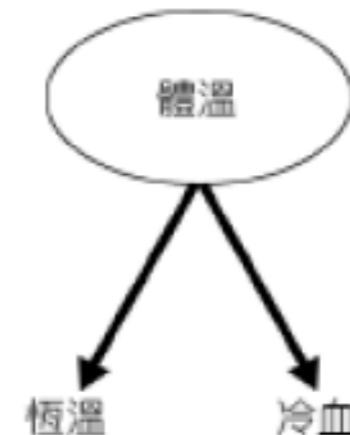
建立決策樹

- 初始值：根節點包含所有的訓練數據
- 重複將每一個內部節點依據**所選擇的特徵**去分割節點，直到達到停止條件

不同類型的節點分割

● 二元屬性

- 二元屬性的資料只有兩個不同的值
- 分割後會產生兩個不同方向的分支



● 名目屬性

- 名目屬性的資料沒有前後次序關係
- 可分割出不同值域的分支
- 每個分支的表示亦可以子集合的型態表示



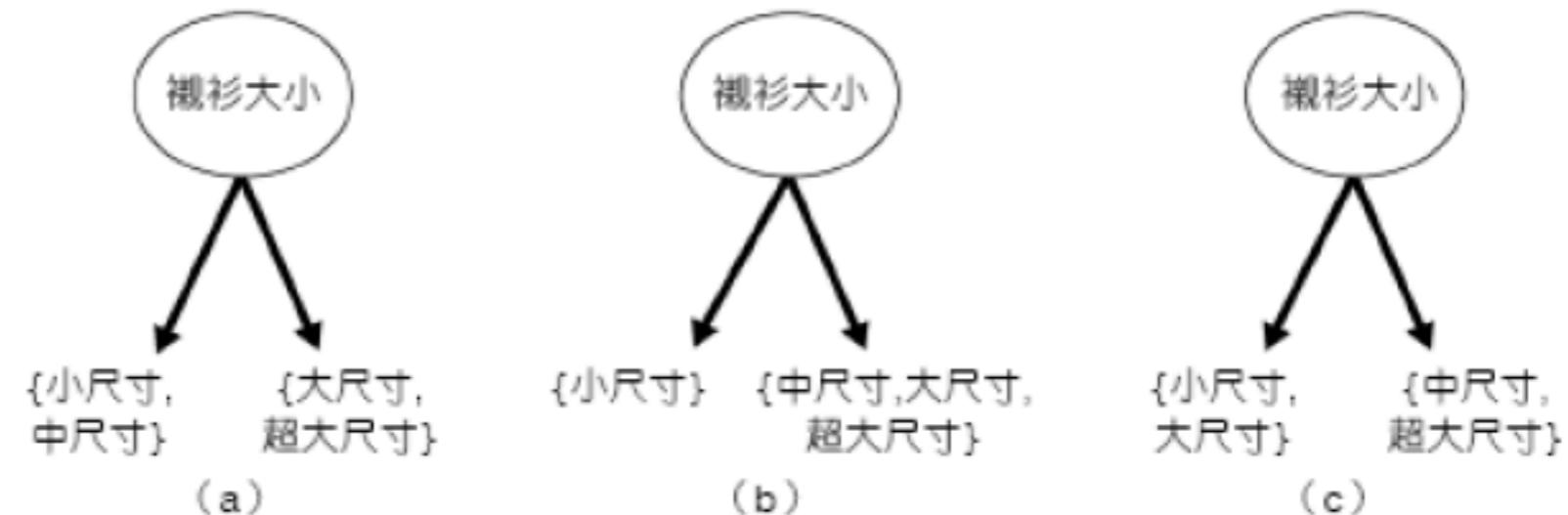
(a) 多元分割



不同類型的節點分割

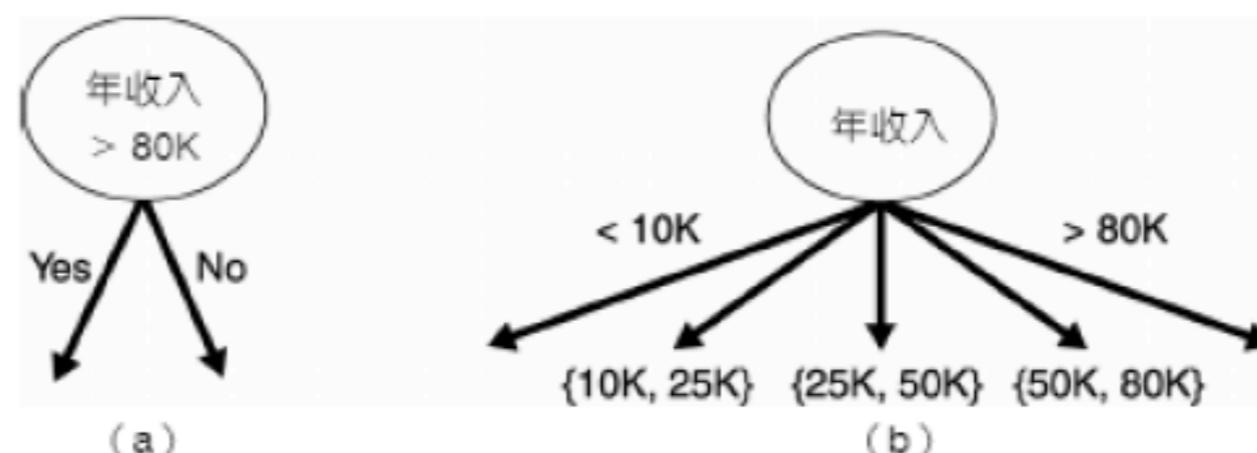
● 順序屬性

- 順序屬性的資料有前後次序關係
- 可產生出二元或是多元分割，順序屬性內的值可以被群組，但是在群組時必須沒有違反屬性值的順序



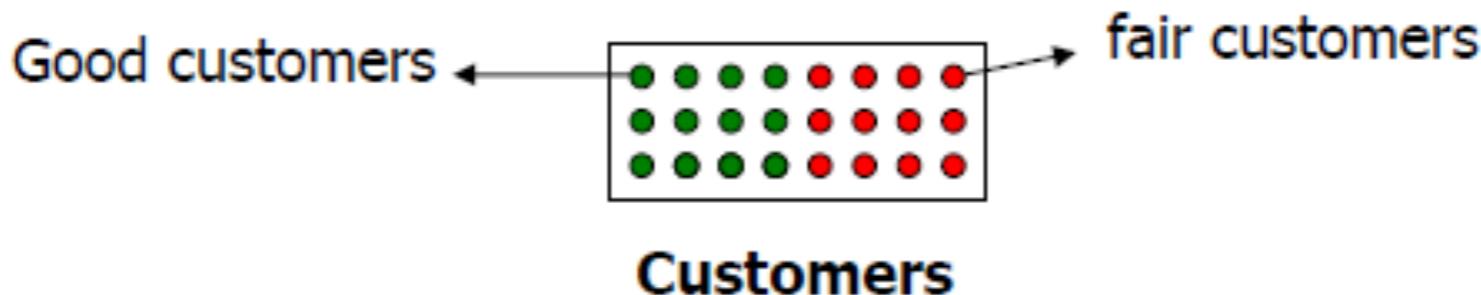
● 連續性屬性

- 可採用離散化的方式，將資料分割成許多區間，但是仍需要保持資料的順序性。

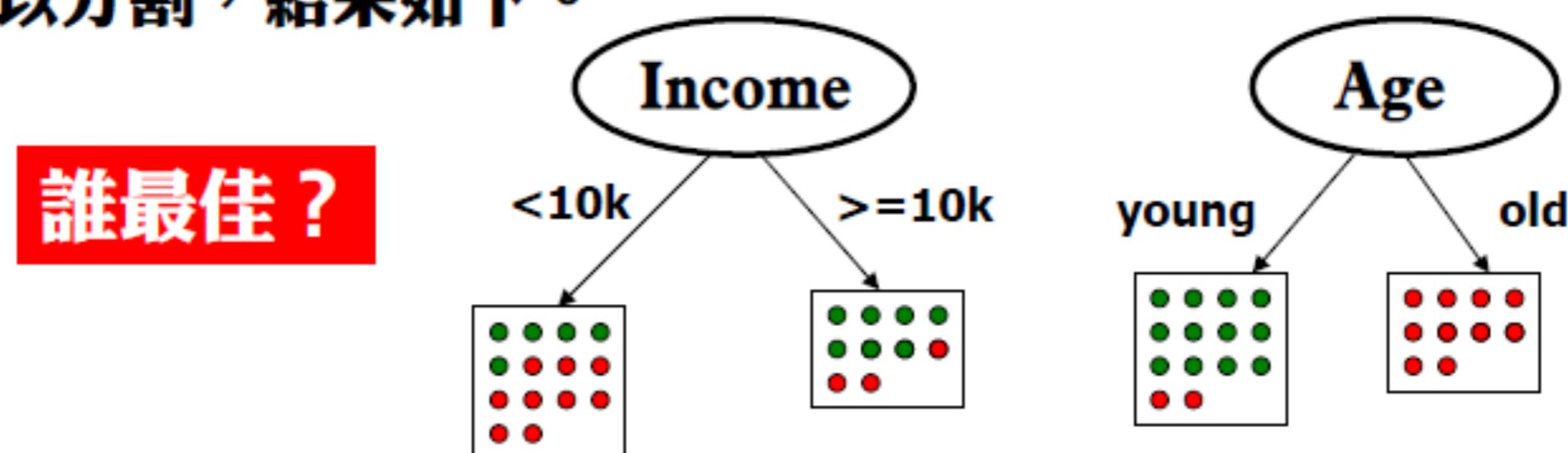


Decision Tree Induction – how to split is better?

- 假設有一個表格共有24筆顧客資料。其類別欄位為“Customers”，可分成“好客人 Good Customers”與“一般客人 Fair Customer”兩類。



- 分別用Income和Age兩個欄位，對這24筆顧客資料加以分割，結果如下。



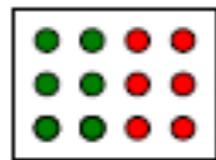
Decision Tree Induction – how to split is better?

- 如何決定分割結果何者較佳：

- 分割結果中，若具有較高**同質性**(Homogeneous)類別的節點，則該分割結果愈佳。

- 因此，需要檢驗節點的**不純度**(Node Impurity):

- 不純度愈低愈好。



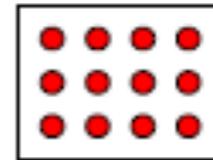
50% red
50% green

High degree
of impurity



75% red
25% green

Low degree
of impurity



100% red
0% green

pure

常用的屬性選擇指標

- 資訊獲利 (Information Gain) : ID3
- 獲利比例 (Gain ratio) : C4.5
- 吉尼係數 (Gini Index) : CART
只能建二元樹

資訊獲利 (Information Gain)

<https://blog.xuite.net/metafun/life/69851478-資訊的度量-+Information+Entropy>

- 熵(亂度)，可當作資訊量的凌亂程度(不確定性)指標，當熵值愈大，則代表資訊的凌亂程度愈高。

- 【範例】丟銅板

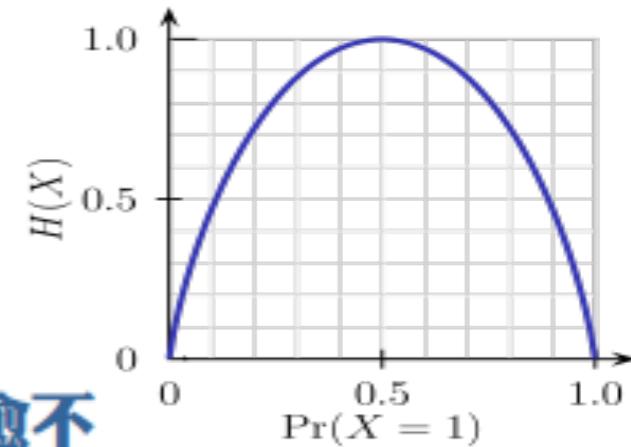
- 若銅板是公平的，則丟出正面與反面的機率是一樣的(最凌亂)
 - 若銅板是動過手腳的，則丟出正面與反面的機率不會是一樣的(愈不凌亂)
 - 紿定一組丟銅板後之資料集合S，該組資料的熵值計算公式為

$$\text{Entropy}(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

- Ex: 若丢了14次銅板，出現了9個正面與5個反面(記為 $[9_+, 5_-]$)，則這個範例的熵為：

$$\text{Entropy}([9_+, 5_-]) = -(9/14) \log_2 (9/14) - (5/14) \log_2 (5/14) = 0.94$$

- 若銅板丟出正面與反面的數量是一樣，則熵為 1(最凌亂)
 - 若銅板是動過手腳的，不論怎麼丟都只會出現正面(或反面)，則熵為 0(最不凌亂)



資訊獲利 (Information Gain)

- 如果資料集合 S 具有 c 個不同的類別，那麼資料集合 S 的熵值計算方式為：

$$\text{Entropy}(S) = \sum_{i=1}^c -p_i \log_2 p_i$$

其中 p_i 為類別 i 在資料集合 S 出現的機率

資訊獲利 (Information Gain)

- ID3演算法是利用資訊獲利來衡量屬性於分類資料的能力。
- 屬性A在資料集合S的資訊獲利Gain(S, A)被定義為：

$$Gain(S, A) = Entropy(S) - \sum_{j=1}^v \frac{|S_j|}{|S|} Entropy(S_j)$$

- 假設屬性 A 中有 v 個不同值 $\{a_1, a_2, \dots, a_v\}$ ，而資料集合 S 會因為這些不同值而產生(分割)出 v 個不同的資料子集合 $\{S_1, S_2, \dots, S_v\}$

資訊獲利 (Information Gain)

● 【範例】天氣評估

- 假設有一套天氣評估系統，它有一些評估屬性 (如: 風力、濕度、...)，用以評估該天氣是否適合打網球。
- 以風力 (Wind)為例，它在所有的訓練資料中所會出現的值為: weak, strong
- 若目前的資料集合 S 有14筆資料，其中有9個正例與5個反例 (記為 $[9_+, 5_-]$)
- 這14個範例資料中，關於風力的資料:
 - Wind = weak 有8筆資料 (S_{weak})，其中有6個正例與2個反例 $[6_+, 2_-]$
 - Wind = strong 有6筆資料 (S_{strong})，其中有3個正例與3個反例 $[3_+, 3_-]$

● 我們想要得知風力這個屬性的資訊獲利為多少。

Information Gain: Example

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Information Gain: Example

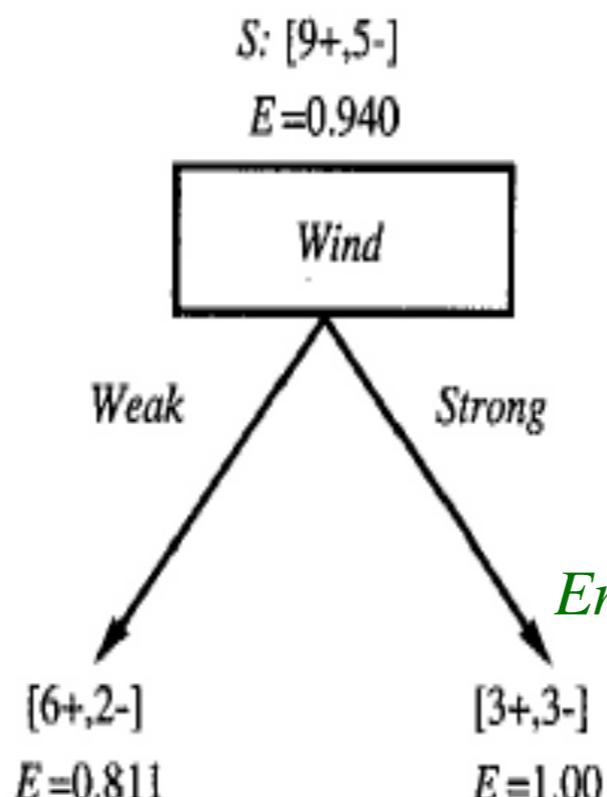
$Values(Wind) = Weak, Strong$

$$S = [9+, 5-]$$

$$S_{Weak} \leftarrow [6+, 2-]$$

$$S_{Strong} \leftarrow [3+, 3-]$$

$$Entropy(S) = -\frac{9}{14} \log \frac{9}{14} - \frac{5}{14} \log \frac{5}{14}$$



$$Entropy(S_{weak}) = -\frac{6}{8} \log \frac{6}{8} - \frac{2}{8} \log \frac{2}{8}$$

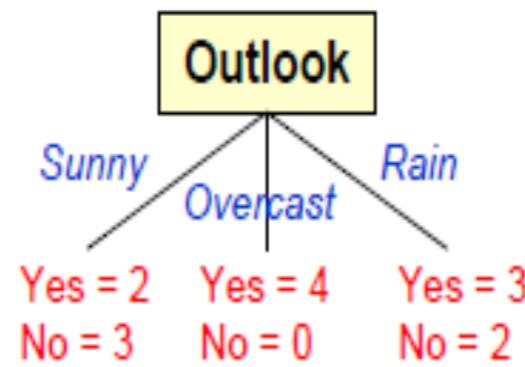
$$Entropy(S_{strong}) = -\frac{3}{6} \log \frac{3}{6} - \frac{3}{6} \log \frac{3}{6}$$

$Gain(S, Wind)$

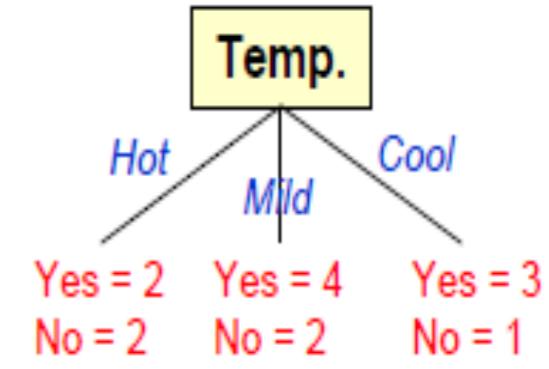
$$\begin{aligned} &= .940 - (8/14).811 - (6/14)1.0 \\ &= .048 \end{aligned}$$

Information Gain: Example

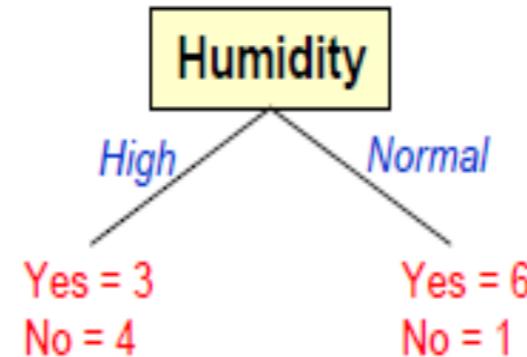
Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



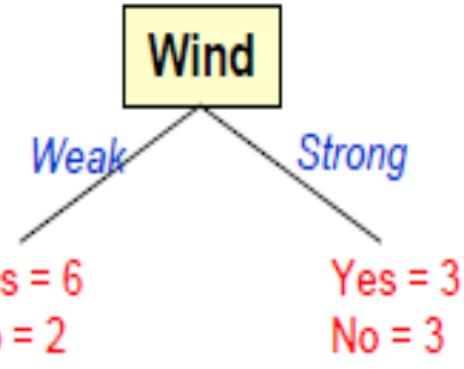
$$Gain(S, \text{Outlook}) = 0.246$$



$$Gain(S, \text{Temperature}) = 0.029$$



$$Gain(S, \text{Humidity}) = 0.151$$



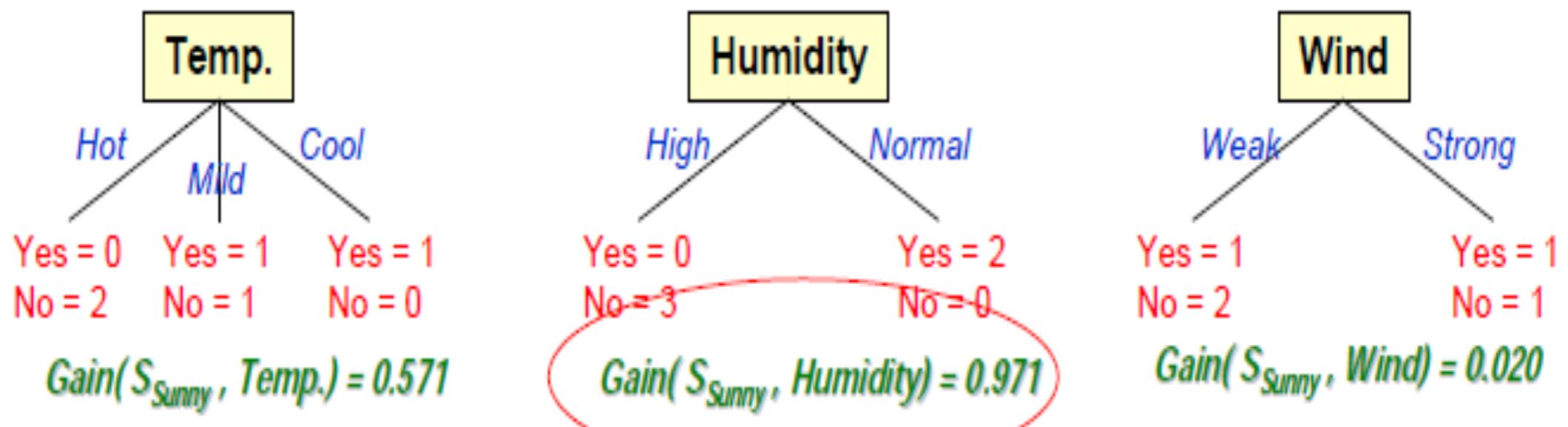
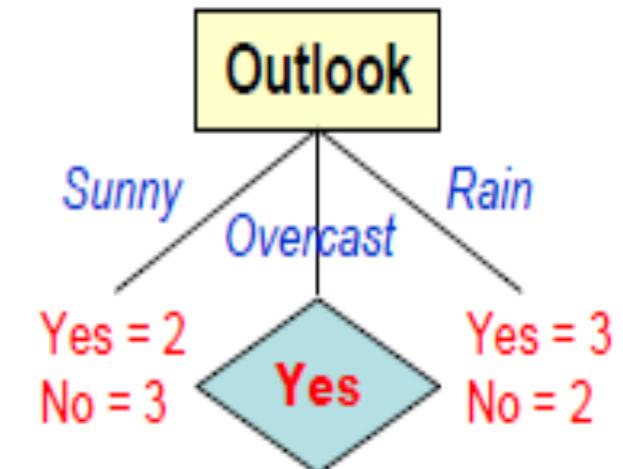
$$Gain(S, \text{Wind}) = 0.048$$

- 挑出具**最大資訊獲利**的屬性，因此以Outlook為根節點 (root)
- 由於Outlook的三個評估值中，Overcast(多雲)的這個評估值得到4個正例 (Yes)，沒有任何反例，因此Outlook = Overcast可得到一個葉子節點 “Yes”。

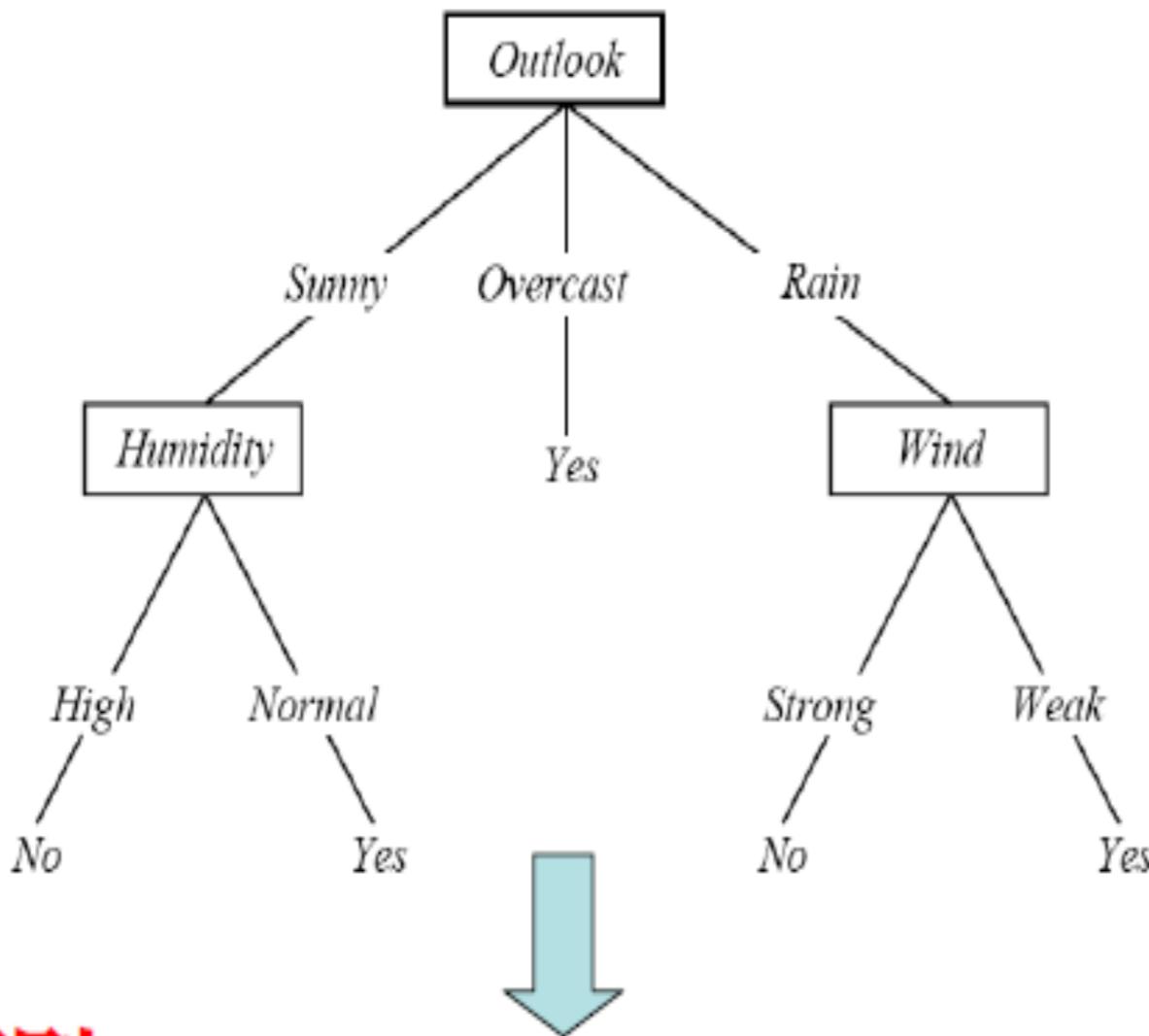
Information Gain: Example

接著，將原始資料中，Outlook=Sunny的所有資料列出，並對Outlook以外的其它所有內部欄位計算其資訊獲利。

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes



Information Gain: Example



分類規則：

If Outlook = Sunny and Humidity = High Then Play Tennis = No

If Outlook = Sunny and Humidity = Normal Then Play Tennis = Yes

If Outlook = Overcast Then Play Tennis = Yes

If Outlook = Rain and Wind = Strong Then Play Tennis = No

If Outlook = Rain and Wind = Weak Then Play Tennis = Yes

C4.5 - gain ratio

- ID3演算法所使用的資訊獲利會傾向選擇**擁有許多不同數值的屬性**
 - 例如：“產品編號”欄位中，每一個產品的產品編號皆不同。
 - 若依產品編號進行分割，會產生出許多分支，且每一個分支都是很單一的結果，其資訊獲利會最大。但這個屬性對於建立決策樹是沒有意義的。

不會考C4.5

- C4.5演算法利用屬性的**獲利比率(Gain Ratio)**克服問題 (資訊獲利正規化)。而求算某屬性A的獲利比率時，除**資訊獲利**外，尚需計算該屬性的**分割資訊值(Split Information)**：

$$SplitInfo_A(S) = -\sum_{j=1}^v \frac{|S_j|}{|S|} \times \log_2\left(\frac{|S_j|}{|S|}\right)$$

- 獲利比率 **GainRatio(A) = Gain(S, A)/SplitInfo_A(S)**
- 擁有**最大獲利比例**的屬性被設為分割屬性

C4.5 - gain ratio

- 試用前述天氣評估系統的訓練資料，計算屬性“Wind”的獲利比率。

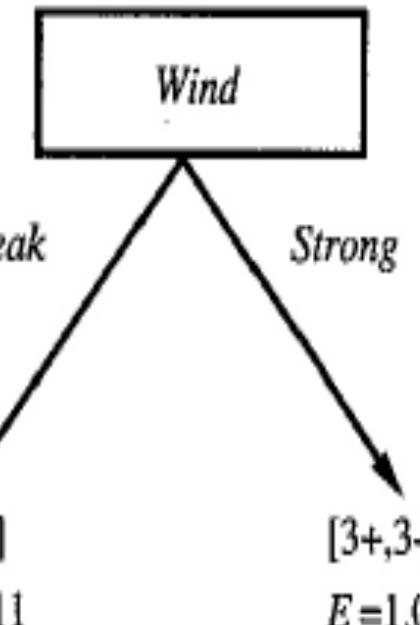
- 由前述得知，欄位Wind的資訊獲利為Gain(S, Wind) = 0.048
- 此欄位的分割資訊值為

$$SplitInfo_{Wind}(S) = -\frac{8}{14} \times \log_2\left(\frac{8}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) = 0.985$$

- GainRatio(Wind) = 0.048/0.985 = 0.049

S: [9+,5-]

E=0.940



$$Gain(S, Wind)$$

$$\begin{aligned}&= .940 - (8/14).811 - (6/14)1.0 \\&= .048\end{aligned}$$

C4.5 - gain ratio

- 試用前述天氣評估系統的訓練資料，計算屬性“Wind”的獲利比率。
 - 由前述得知，欄位Wind的資訊獲利為 $\text{Gain}(S, \text{Wind}) = 0.048$
 - 此欄位的分割資訊值為 $\text{SplitInfo}_{\text{Wind}}(S) = -\frac{8}{14} \times \log_2(\frac{8}{14}) - \frac{6}{14} \times \log_2(\frac{6}{14}) = 0.985$
 - $\text{GainRatio}(\text{Wind}) = 0.048 / 0.985 = 0.049$
- 其它三個屬性的相關資訊：
 - Outlook: $\text{Gain}(S, \text{Outlook}) = 0.246$; $\text{SplitInfo} = 1.577$; $\text{Gain Ratio} = 0.156$
 - Temp: $\text{Gain}(S, \text{Temp}) = 0.029$; $\text{SplitInfo} = 1.362$; $\text{Gain Ratio} = 0.021$
 - Humidity: $\text{Gain}(S, \text{Humidity}) = 0.151$; $\text{SplitInfo} = 1$; $\text{Gain Ratio} = 0.151$

使用Outlook仍是最好的，但是現在Humidity成了有力的競爭者。

吉尼係數 (Gini Index)

- CART (Classification and Regression Tree)由Friedman等人於1980年代提出，是一種產生二元樹的技術，以吉尼係數做為選擇屬性的依據。
- CART與ID3、C4.5、C5.0演算法的最大相異之處是其在每一個節點上都是採用二分法，也就是一次只能夠有兩個子節點，ID3、C4.5、C5.0則在每一個節點上可以產生不同數量的分枝。
- 假設資料集合 S 包含 n 個類別，吉尼係數 $Gini(S)$ 定義為

$$Gini(S)=1-\sum_{j=1}^n p_j^2$$

p_j 為在S中的值組屬於類別j的機率

吉尼係數 (Gini Index)

- 利用屬性A分割資料集合 S 為 S_1 與 S_2 (二元分割)。則根據此一分割要件的吉尼係數 $Gini_A(S)$ 為

$$Gini_A(S) = \frac{|S_1|}{|S|} Gini(S_1) + \frac{|S_2|}{|S|} Gini(S_2)$$

其中， S_1 與 S_2 是針對欄位 A 內的不同數值所構成的兩組資料子集合。

- 不純度的降低值為：

$$\Delta Gini(A) = Gini(S) - Gini_A(S)$$

- 挑選擁有**最大不純度的降低值**、或**吉尼係數 $Gini_A(S)$ 最小**的屬性作為分割屬性。

吉尼係數 (Gini Index)

- 例：試用前述天氣評估系統的訓練資料，使用CART建構分類樹。
 - 目前的資料集合S有9筆資料為打網球 = yes，剩下5筆為no，故目前資料集合S的吉尼係數Gini(S)為：

$$Gini(S) = 1 - \left(\frac{9}{14} \right)^2 - \left(\frac{5}{14} \right)^2 = 0.459$$

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

吉尼係數 (Gini Index)

- 假設 “Outlook”屬性分割 S 中 10 筆資料被分類到 S_1 : {Sunny, Rain} 剩下 4 筆到 S_2 : {Overcast} (註：這三個數值合併成兩組的組合方式很多，在此僅舉其中一組為例)

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$Gini(S_{Outlook \in \{Sunny, Rain\}}) = 1 - \left[\left(\frac{5}{10} \right)^2 + \left(\frac{5}{10} \right)^2 \right] = \frac{1}{2}$$

吉尼係數 (Gini Index)

- 假設 “Outlook”屬性分割 S 中 10 筆資料被分類到 S_1 : {Sunny, Rain} 剩下 4 筆到 S_2 : {Overcast} (註：這三個數值合併成兩組的組合方式很多，在此僅舉其中一組為例)

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$Gini(S_{Outlook \in \{Sunny, Rain\}}) = 1 - \left[\left(\frac{5}{10} \right)^2 + \left(\frac{5}{10} \right)^2 \right] = \frac{1}{2}$$

$$Gini(S_{Outlook \in \{Overcast\}}) = 1 - \left[\left(\frac{4}{4} \right)^2 + \left(\frac{0}{4} \right)^2 \right] = 0$$

吉尼係數 (Gini Index)

- 假設 “Outlook”屬性分割 S 中 10 筆資料被分類到 S_1 : {Sunny, Rain} 剩下 4 筆到 S_2 : {Overcast} (註：這三個數值合併成兩組的組合方式很多，在此僅舉其中一組為例)

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$Gini(S_{Outlook \in \{Sunny, Rain\}}) = 1 - \left[\left(\frac{5}{10} \right)^2 + \left(\frac{5}{10} \right)^2 \right] = \frac{1}{2}$$

$$Gini(S_{Outlook \in \{Overcast\}}) = 1 - \left[\left(\frac{4}{4} \right)^2 + \left(\frac{0}{4} \right)^2 \right] = 0$$

$$Gini_{Outlook}(S) = \left(\frac{10}{14} \right) \times \frac{1}{2} + \left(\frac{4}{14} \right) \times 0 = 0.3571$$

吉尼係數 (Gini Index)

- 挑選擁有**最大不純度的降低值**、或**吉尼係數** $Gini_A(S)$ **最小**的屬性作為分割屬性。
 - 根據上述計算結果，得知Outlook有**最小的吉尼係數** (即：**0.3571**)，故挑選它作為根節點。

建立決策樹

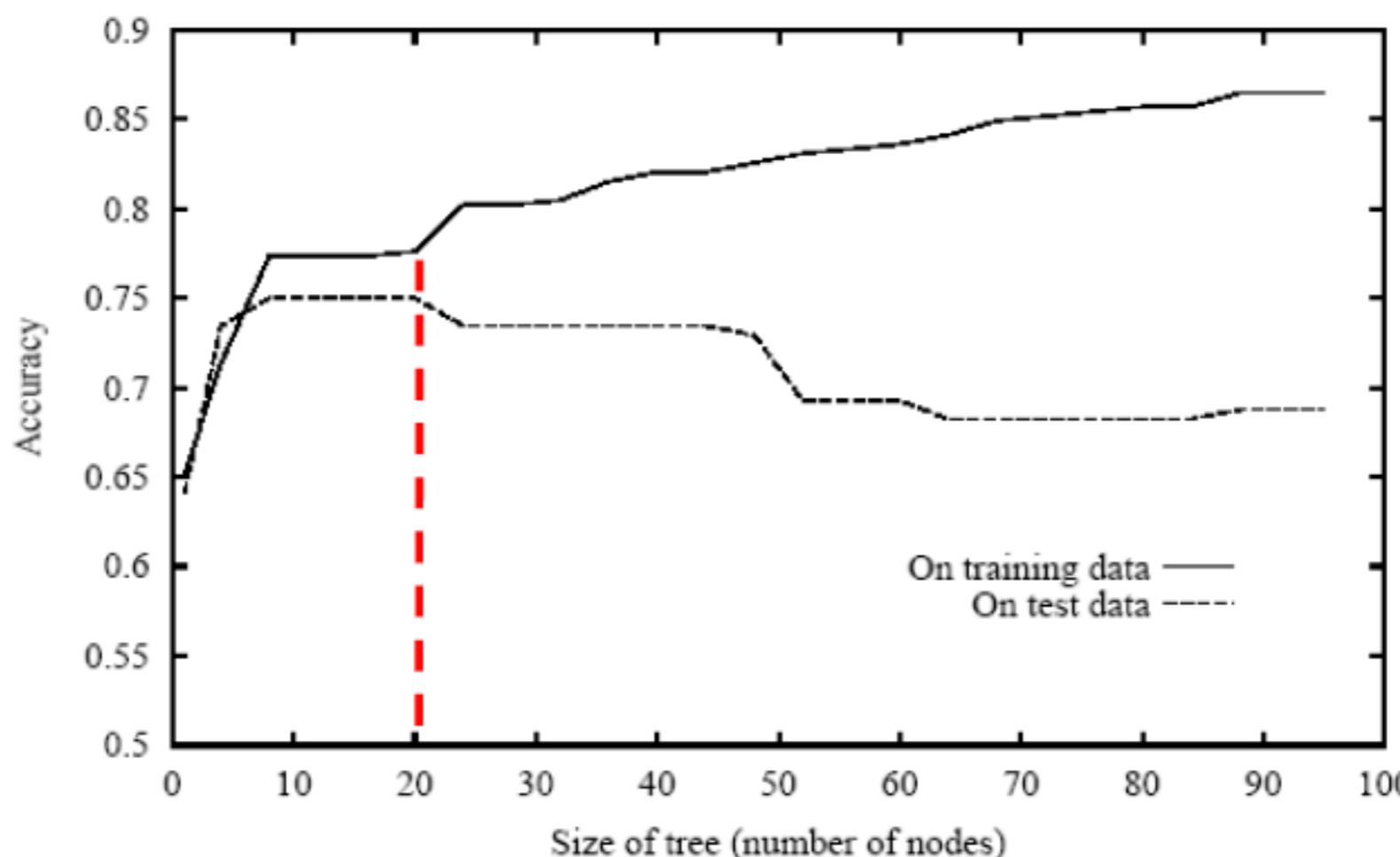
- Step 1: 找出最能將資料點均勻區分的問題作為樹的內部節點，並將節點分割以產生對應的分支
- Step 2: 在每一個葉節點重複Step 1，直到達到停止條件

使用決策樹常見的問題

- 避免過度適配資料
- 合併連續值屬性
- 屬性選擇指標的其它度量標準

過度配適 (Overfitting)

- 決策樹學習可能遭遇模型**過度配適** (overfitting) 的問題
 - 過度配適是指**模型對於範例的過度訓練，導致模型記住的不是訓練資料的一般特性，反而是訓練資料的局部特性**。對測試樣本的分類將會變得很不精確
 - 過度適配訓練資料：



過度配適 (Overfitting)

● 導致過度適配的原因：

- 一種可能原因是訓練範例含有**雜訊**和離異值
- 當訓練數據沒有雜訊時，過度適配也有可能發生，特別是當訓練範例的**數量太少**，使得某一些屬性“恰巧”可以很好地分割目前的訓練範例，但卻與實際的狀況並無太多關係。

修剪

■ 事先修剪 (pre-pruning)

- 事先設定停止決策樹生長的門檻值，當分割的評估值未達此門檻值時，就會停止擴長
- 優點：較具有執行效率
- 缺點：可能**過度修剪 (over-pruning)**、門檻值設定不易

■ 事後修剪 (post-pruning)

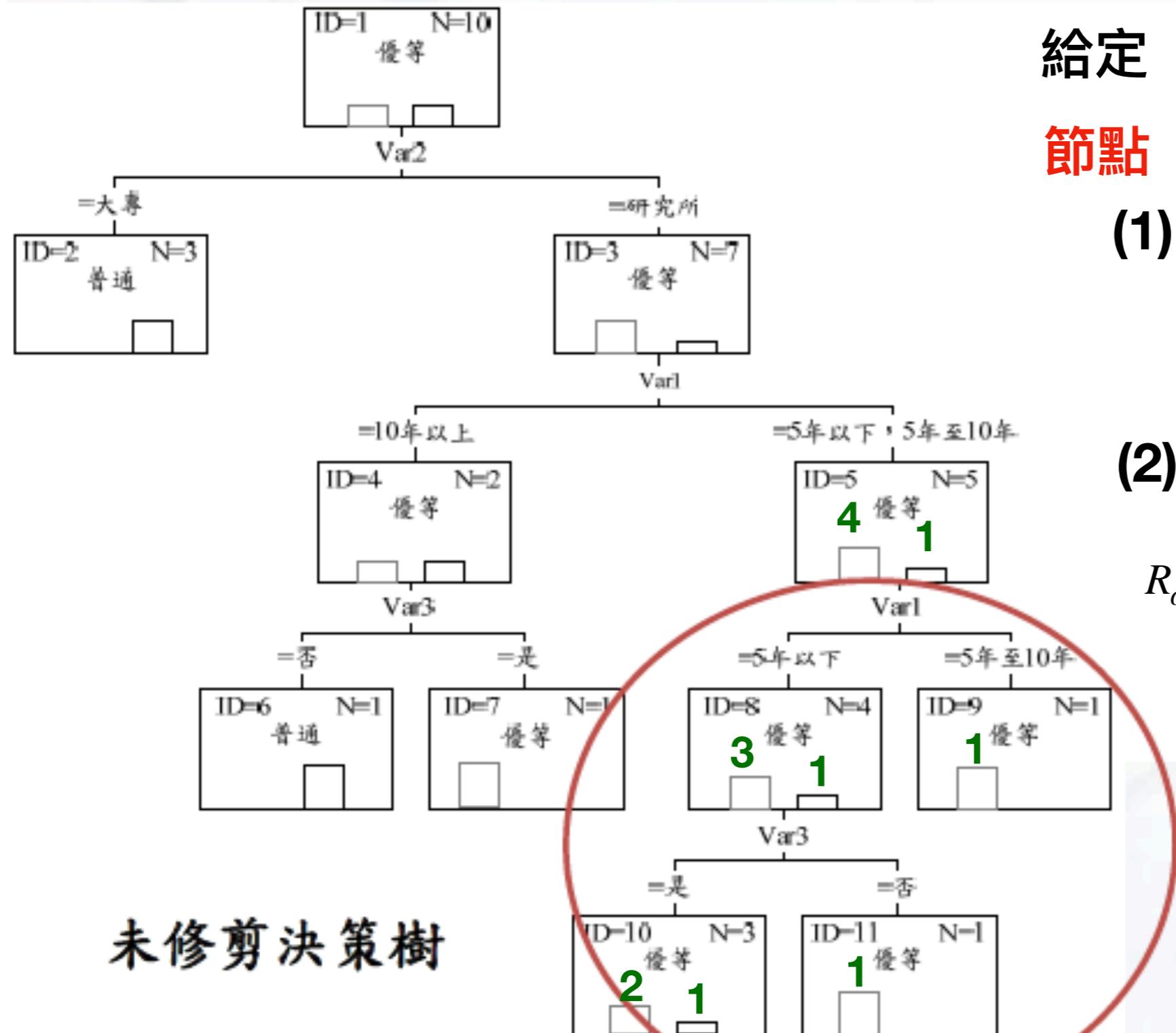
- 在樹完全長成後再修剪，引入測試組樣本來評估決策樹對於新輸入資料的分類與預測結果
- 優點：可解決**過度配適**，避免產生稀少樣本樹的葉節點，以及加強對雜訊的忍受程度
- 缺點：效率較低

最小成本複雜修剪(minimal cost-complexity pruning)

- 為事後修剪方法
- 同時考慮**分類錯誤率**以及**決策樹的規模大小**
 1. 以排列組合的方式列出數種修剪後的決策樹；
 2. 計算這些樹的分類錯誤率(classification error)與決策樹複雜度，即節點個數，並找出具有最小誤差的決策樹
- 分類錯誤率會隨著修剪分枝的數目呈正比遞增
- 對某一棵決策樹其成本—複雜性的定義為決策樹節點個數與分類錯誤率的函數

$$R_\alpha(t) = R(t) + \alpha \times N_{\text{leaf}}$$

Example



給定 $\alpha = 0.01$

節點 8

(1) 修剪節點 10 & 11

$$R(t = 8) = \frac{1}{4} \times \frac{4}{10} = 0.1$$

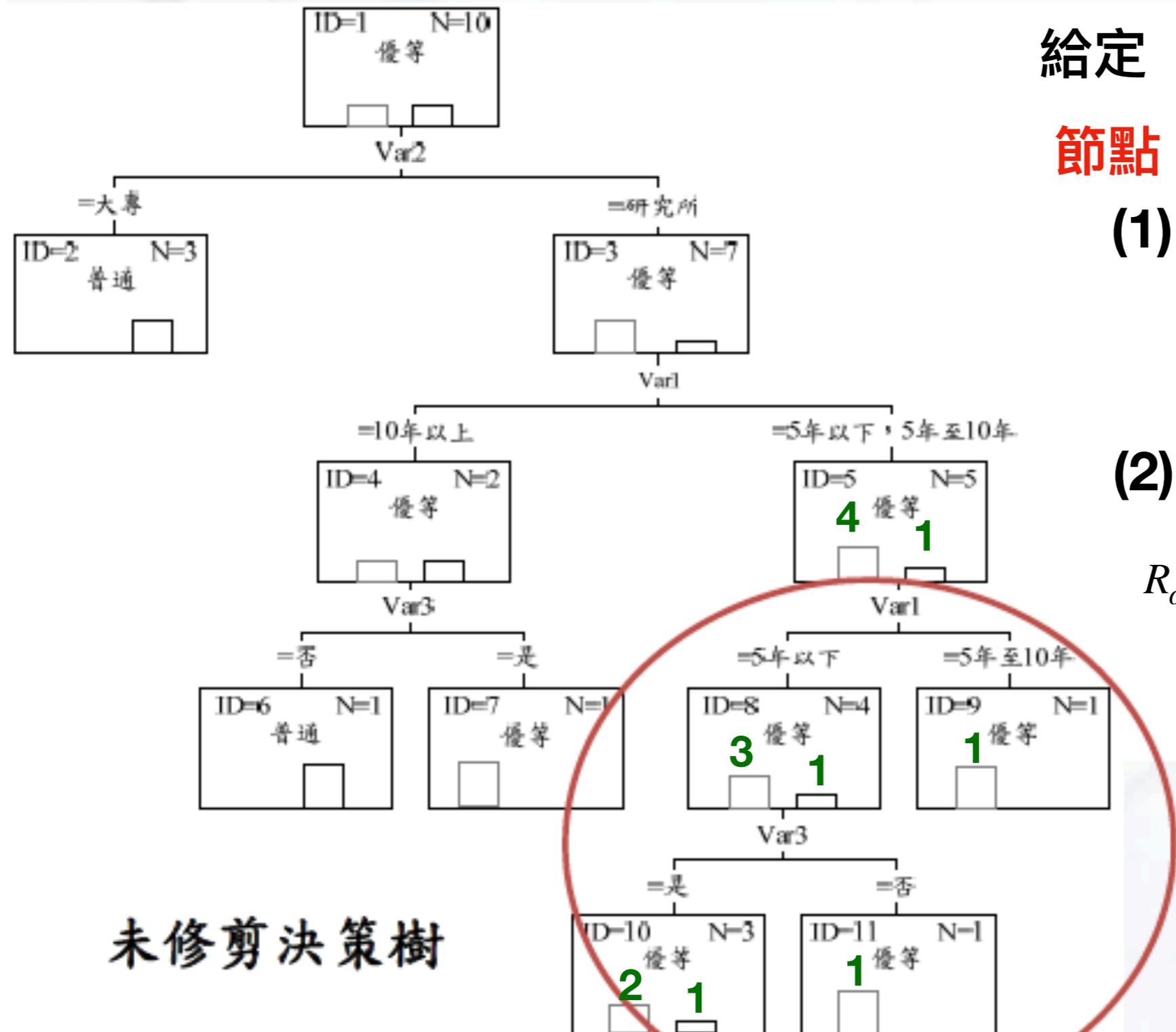
(2) 不修剪節點 10 & 11

$$\begin{aligned} R_\alpha(t = 8) &= \left(\frac{1}{3} \times \frac{3}{10} + 0 \right) + 0.01 \times 2 \\ &= 0.12 \end{aligned}$$

$R_\alpha(t = 8) > R(t = 8)$ 修剪

未修剪決策樹

Example



給定 $\alpha = 0.01$

節點 5

(1) 修剪節點 8 & 9

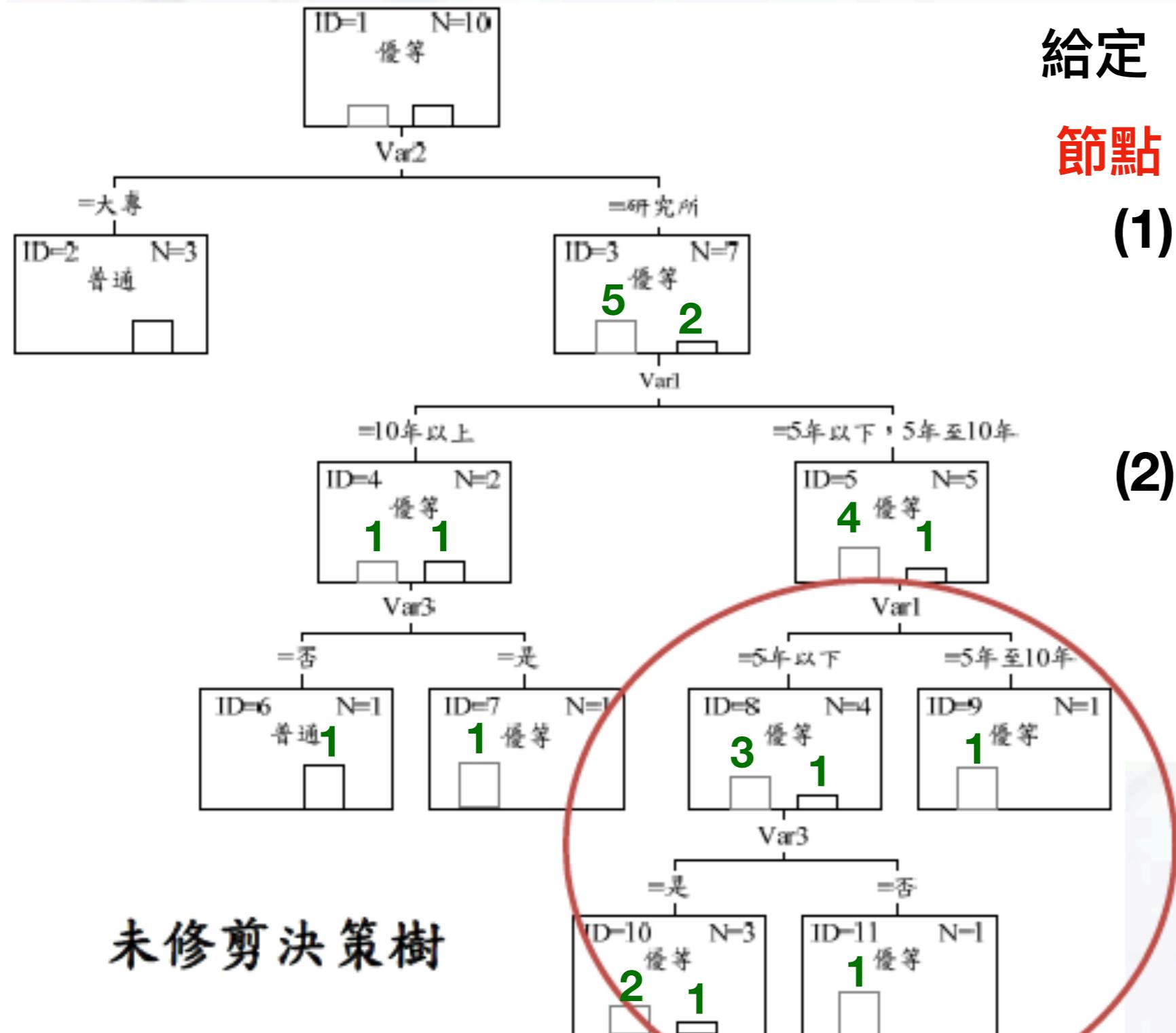
$$R(t = 5) = \frac{1}{5} \times \frac{5}{10} = 0.1$$

(2) 不修剪節點 8 & 9

$$\begin{aligned} R_\alpha(t = 5) &= \left(\frac{1}{4} \times \frac{4}{10} + 0 \right) + 0.01 \times 2 \\ &= 0.12 \end{aligned}$$

$R_\alpha(t = 5) > R(t = 5)$ 修剪

Example



給定 $\alpha = 0.01$

節點 4

(1) 修剪節點 6 & 7

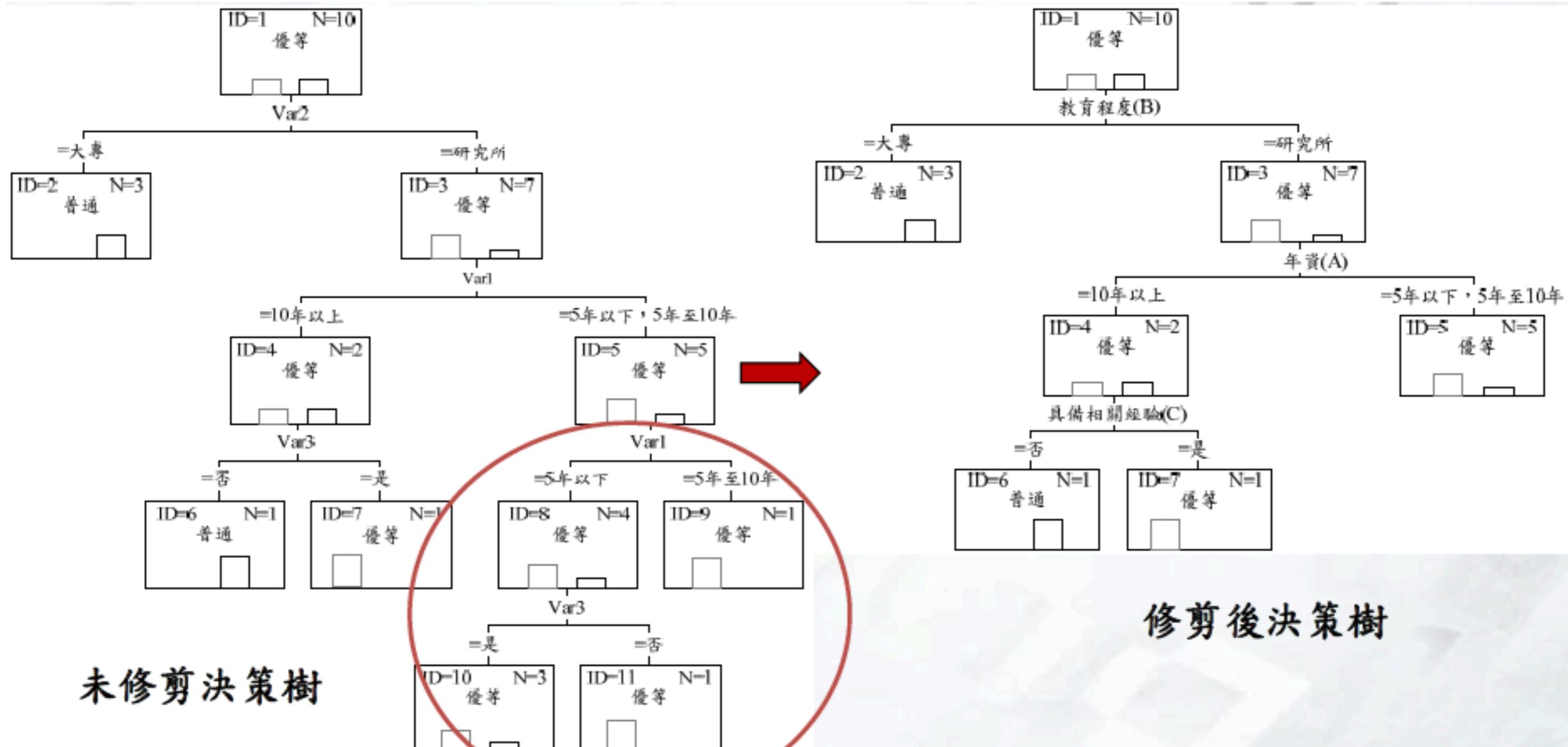
$$R(t = 4) = \frac{1}{2} \times \frac{2}{10} = 0.1$$

(2) 不修剪節點 6 & 7

$$\begin{aligned} R_\alpha(t = 4) &= (0 + 0) + 0.01 \times 2 \\ &= 0.02 \end{aligned}$$

$R_\alpha(t = 4) < R(t = 4)$ 不修剪

example

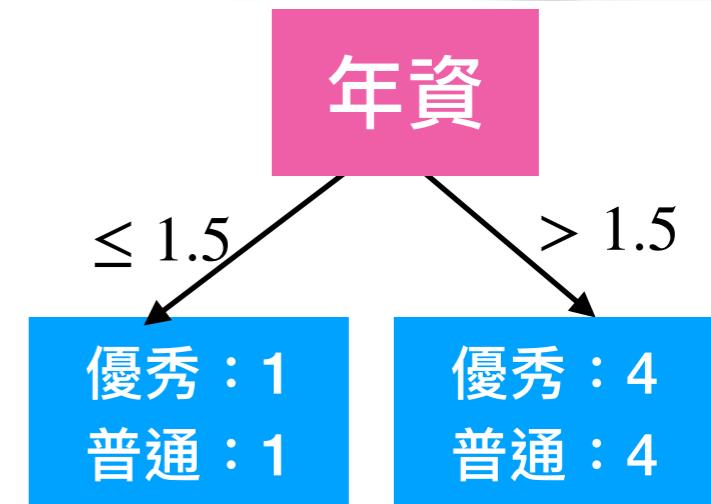


連續屬性

- 許多決策樹學習方法限制為取**離散值**的屬性
 - 決策樹要預測的目標屬性必須是離散的
 - 樹的內部節點的屬性也必須是離散的
- 簡單刪除上面第2個限制的方法
 - 透過動態地定義新的離散值屬性來實現，即先把連續值屬性的值域分割為離散的區間集合，或設定門檻值以進行二分法
- 例子: Age
 - 使用門檻值，大於門檻值的資料為yes，小於門檻值的為no。
 - 使用區間值，以區分出多個離散區間。

Example

假設某公司人力資源部門欲瞭解職員的表現，抽取10位現職員工為樣本



兩兩資料點的中間值

年資	1	2	4	6	8	12	15				
分割點		1.5	3	5	7	10	13.5				
評等		\leq	$>$	\leq	$>$	\leq	$>$	\leq	$>$	\leq	$>$
優秀	1	4	1	4	3	2	3	2	4	1	4
普通	1	4	2	3	2	3	3	2	4	1	5
Entropy	1	0.965	0.971	1	1	0.892					
Gini	0.500	0.476	0.480	0.500	0.500	0.444					

變異降低 (variance reduction)

- 目標變數為連續屬性時，分枝準則可改用變異降低
 - 變異數是量測資料值與平均值的差異（即該節點內的各筆資料目標值與目標平均值之均方差）
 - 檢視其分枝節點內資料的變異程度是否有顯著縮減
 - 在評估完所有屬性進行分枝後所計算出的變異數後，最後再比較候選屬性的變異數縮減量，並選出具有最大變異數縮減量的屬性為分枝變數

$$S_t^2 = \frac{\sum_{i=1}^{N_t} (y_{i,t} - \bar{y}_t)^2}{N_t}$$

Example

• 職員收入的資料

職員	年資(A)	教育程度(B)	具備相關經驗(C)	月收入(K)
001	5年以下	研究所	是	45
002	10年以上	研究所	否	60
003	5年以下	研究所	是	42
004	5年以下	大專	是	39
005	5年以下	研究所	否	42
006	10年以上	研究所	是	75
007	5年至10年	大專	否	40
008	5年至10年	研究所	是	45
009	5年至10年	大專	否	44
010	5年以下	研究所	是	38

年資

< 10
45, 42, 39,
42, 40, 45,
44, 38

≥ 10
60, 75

$$S^2 = 6.36 \quad S^2 = 56.25$$

• 各屬性分枝後的變異數

	年資	教育程度	具備相關經驗
分枝點	[5年以下 & 5年至10年]、[10年以上]	[大專]、[研究所]	[否]、[是]
變異數	$0.8 \times 6.36 + 0.2 \times 56.25 = 16.34$	$0.3 \times 4.67 + 0.7 \times 149.39 = 105.97$	$0.4 \times 62.75 + 0.6 \times 160.22 = 121.23$

屬性選擇指標

- **訊息獲利:**
 - 趨向於包含多個值的屬性
 - 以“天氣評估”為例，若考慮將屬性Date納入可能的決策樹節點之眾多候選者之一，因為它有大量的可能值，於是這個屬性會被選作樹的根節點。
- **獲利比率:**
 - 會產生不平均的分割，也就是分割的一邊會非常小於另一邊
- **吉尼係數:**
 - 傾向於包含多個值的屬性
 - 當類別個數很多時會有困難
 - 傾向那些會導致平衡切割並且兩邊均為純粹的測試
- 尚有其他的度量標準，也都各有利弊。

Python

■ Adult Data Set (重新命名為 adult.csv)

<https://archive.ics.uci.edu/ml/datasets/adult>

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

names = ['age', 'workclass', 'fnlwgt', 'education', 'education-num',
         'marital-status', 'occupation', 'relationship', 'race', 'sex',
         'capital-gain', 'capital-loss', 'hours-per-week', 'native-country',
         'income']
data = pd.read_csv("adult.csv", header = None, names=names)

df = data[['age', 'sex', 'hours-per-week', 'education', 'income']]

df.info()

df.describe()

df.head()
```

Python

```
# ----- Building a decision tree -----
from sklearn.tree import DecisionTreeClassifier

X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values

tree = DecisionTreeClassifier(criterion='gini',
                               random_state=1, max_depth = 5)
tree.fit(X, y)
```

Python

```
data_dummies = pd.get_dummies(df, drop_first = True)

data_dummies.info()

X = data_dummies.iloc[:, :-1].values
y = data_dummies.iloc[:, -1].values

tree = DecisionTreeClassifier(criterion='gini',
                               random_state=1, max_depth = 5)
tree.fit(X, y)

print('score:', tree.score(X,y))
```

正確率

Python

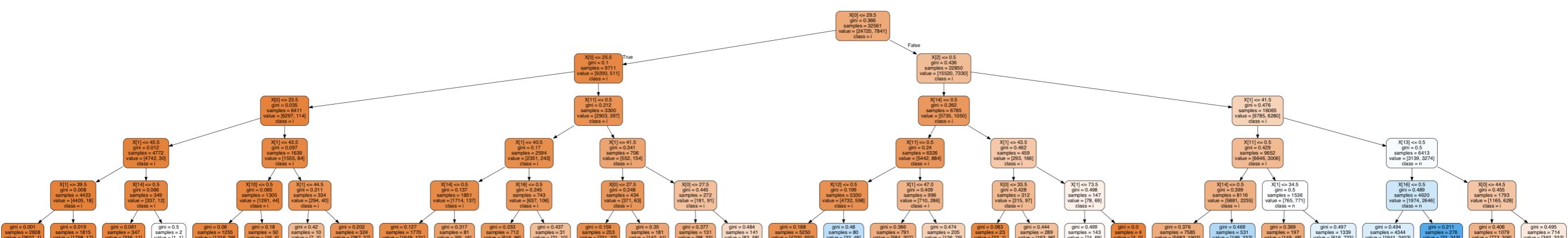
- 畫出決策樹
 - 須先安裝graphviz 及 pydotplus
 - <https://www.graphviz.org>
 - <https://pypi.org/project/pydotplus/>
 - Anaconda
 - <https://anaconda.org/anaconda/graphviz>
 - <https://anaconda.org/conda-forge/pydotplus>

Python

```
# 利用graphviz畫出決策樹
import graphviz
from sklearn.tree import export_graphviz
from pydotplus import graph_from_dot_data

dot_data = export_graphviz(tree, filled = True,
                           rounded = True,
                           class_names = data_dummies.iloc[:, -1].name,
                           out_file= None)

graph = graph_from_dot_data(dot_data)
graph.write_png('tree.png')
```



Pros & Cons

■ Pros:

- 簡單，直覺，容易解讀（容易視覺化）

■ Cons:

- 不穩定性：資料出現些許變動，可能就會得到不同的決策樹
- 容易出現 overfitting
- 不準確性：一開始時使用不太好的分割，可能會產生更好的預測