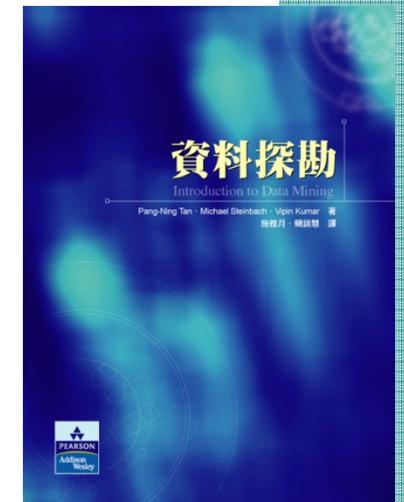


第 4 章

分類法：基本概念、 決策樹及模式的評估



© 2008 台灣培生教育出版 (Pearson Education Taiwan)

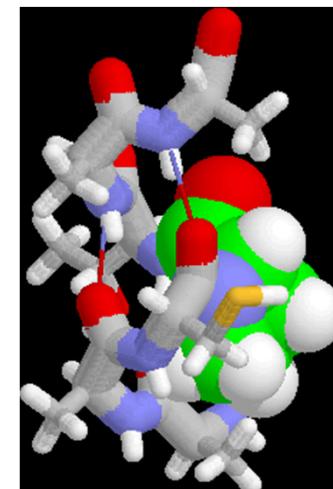
分類法：定義

- 目的是將一個物件指定至其中一個已預設的分類中
- 分類是指建立一個學習目標函數 f ，使得這個學習函數可以藉由 X 屬性對應至 y 的類別
- 適合預測二元分類或是名目分類的問題
- 具順序特性的類別其效果較差
- (例：要分類一個人的收入情形，如高、中、低)



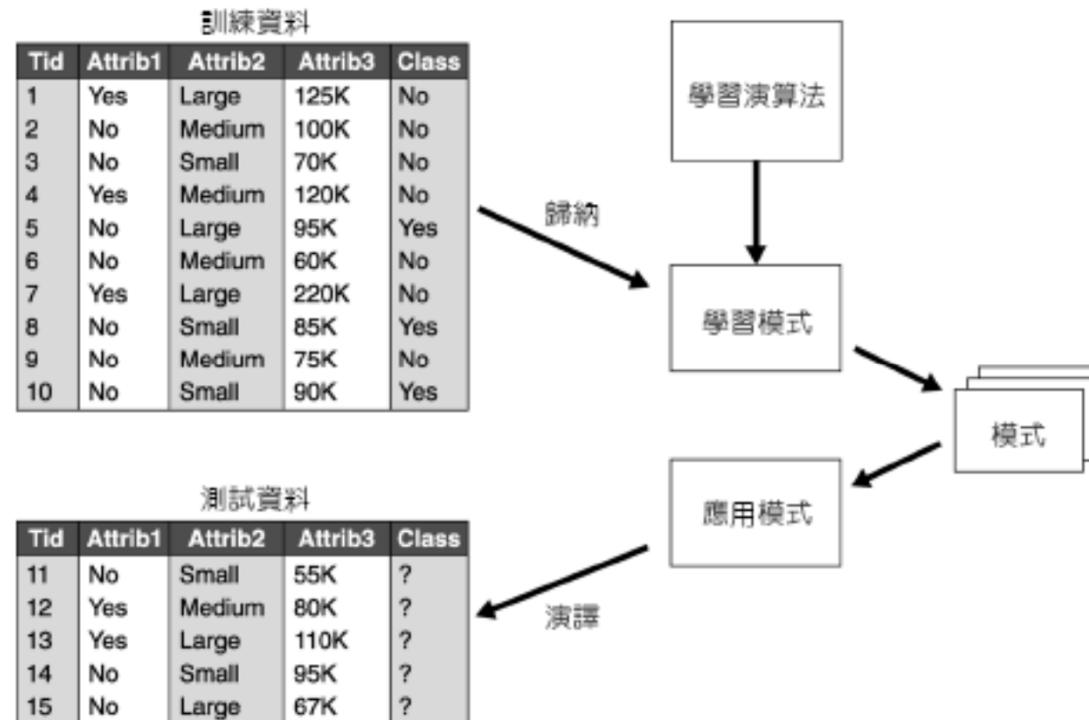
分類技術

- 從輸入資料中建立分類模式的系統化方法
 - 決策樹 (Decision tree)
 - 類神經網路 (artificial neural network)
 - 支援向量機 (support vector machines)
 - 及單純貝氏 (Naïve Bayes)



建立分類模式的做法

- 訓練資料是由一些已知類別標記的資料所組成
- 訓練資料主要是用來**建立分類模式**
- 用此模式來對**未知類別**標記的測試資料進行預測



解決分類問題

表 4.1 脊椎動物

| 名稱 | 身體溫度 | 外表 | 胎生 | 水生 | 飛行能力 | 有無四肢 | 冬眠 | 類別標記 |
|------|------|----|----|----|------|------|----|------|
| 人類 | 恆溫 | 毛髮 | 是 | 否 | 否 | 是 | 否 | 哺乳類 |
| 蟒蛇 | 冷血 | 鱗片 | 否 | 否 | 否 | 否 | 是 | 爬蟲類 |
| 鮭魚 | 冷血 | 鱗片 | 否 | 是 | 否 | 否 | 否 | 魚類 |
| 鯨魚 | 恆溫 | 毛髮 | 是 | 是 | 否 | 否 | 否 | 哺乳類 |
| 青蛙 | 冷血 | 無 | 否 | 一半 | 否 | 是 | 是 | 兩棲類 |
| 科摩多龍 | 冷血 | 鱗片 | 否 | 否 | 否 | 是 | 否 | 爬蟲類 |
| 蝙蝠 | 恆溫 | 毛髮 | 是 | 否 | 是 | 是 | 是 | 哺乳類 |
| 鴿子 | 恆溫 | 羽毛 | 否 | 否 | 是 | 是 | 否 | 鳥類 |
| 貓 | 恆溫 | 軟毛 | 是 | 否 | 否 | 是 | 否 | 哺乳類 |
| 豹鯊 | 冷血 | 鱗片 | 是 | 是 | 否 | 否 | 否 | 魚類 |
| 海龜 | 冷血 | 鱗片 | 否 | 一半 | 否 | 是 | 否 | 爬蟲類 |
| 企鵝 | 恆溫 | 羽毛 | 否 | 一半 | 否 | 是 | 否 | 鳥類 |
| 豪豬 | 恆溫 | 刺 | 是 | 否 | 否 | 是 | 是 | 哺乳類 |
| 鰻魚 | 冷血 | 鱗片 | 否 | 是 | 否 | 否 | 否 | 魚類 |
| 蠑螈 | 冷血 | 無 | 否 | 一半 | 否 | 是 | 是 | 兩棲類 |

| 名稱 | 身體溫度 | 外表 | 胎生 | 水生 | 飛行能力 | 有無四肢 | 冬眠 | 類別標記 |
|------|------|----|----|----|------|------|----|------|
| 大毒蜥蜴 | 冷血 | 鱗片 | 否 | 否 | 否 | 是 | 是 | ? |

分類評估標準

決定於模式能夠正確預測測試資料的比例

混亂矩陣 (Confusion matrix)

| | | 預測類別 | |
|------|-----------|-----------|-----------|
| | | Class = 1 | Class = 0 |
| 實際類別 | Class = 1 | f_{11} | f_{10} |
| | Class = 0 | f_{01} | f_{00} |

$$accuracy = \frac{\text{正確預測的個數}}{\text{總預測個數}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

$$\text{錯誤率} = \frac{\text{錯誤預測的個數}}{\text{總預測個數}} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

大部份的演算法都期望能夠得到最高的正確率，或是最低的錯誤率

決策樹

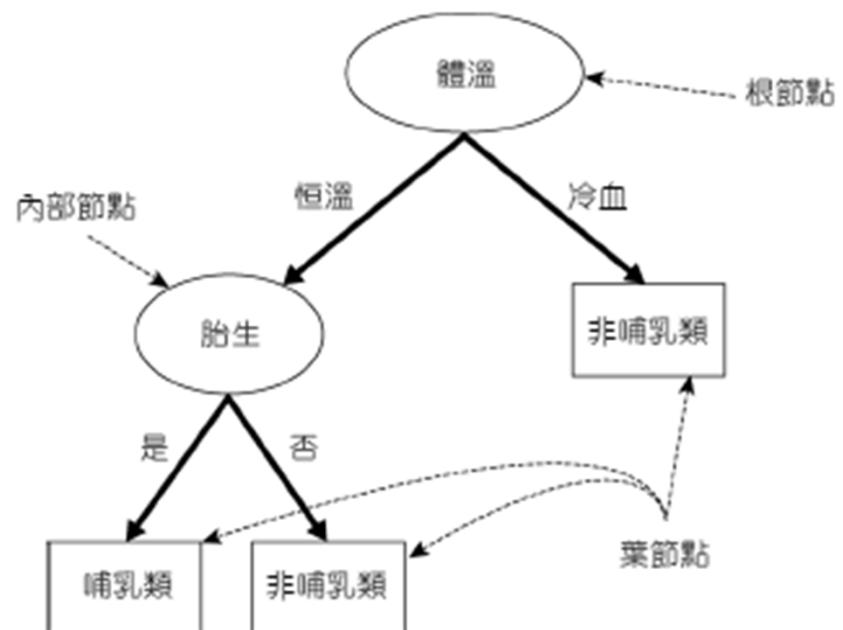
- 樹包含三個節點：

- **根節點**：沒有任何進入的邊，而且有0個或是輸出的邊
- **內部節點**：每個節點都有一個輸入的邊，以及二個或多個輸出的邊
- **葉節點或是終端節點**：每個節點都有一個輸入的邊，但沒有輸出的邊

每個葉節點都是一个**類別標記**

- 決策樹範例

- 哺乳類動物分類的問題



如何建立決策樹

- 決策樹可從已知**屬性集合**中建構起來
- 採用**貪婪策略**（greedy strategy）建立決策樹
- 決策樹演算法：其中一種是Hunt's 演算法，它是一些現存方法的基礎，包含ID3、C4.5 及CART

Hunt's 演算法

- 決策樹是用遞迴的方式不斷地將訓練資料分割至後繼的子集合中
- 假設 D_t 是與節點 t 的相關的訓練資料，而 $y = \{y_1, y_2, \dots, y_c\}$ 是類別標記
- Hunt's 演算法的遞迴定義
 - 步驟1：如果在 D_t 中的所有記錄都屬於相同類別 y_t ，那麼 t 就是一個葉節點，標記為 y_t
 - 步驟2：如果 D_t 包含一些屬於一個以上類別記錄時，則會選取一個屬性測試條件作為測試節點，以便將資料分割至較小的子集合中

Hunt's 演算法

| 編號 | 二元屬性 | | 類別屬性 | | 連續值 | 類別 |
|----|------|----------|------|-----|-----|----|
| | 有房子 | 婚姻狀況 | 年收入 | 未還款 | | |
| 1 | Yes | Single | 125K | No | | |
| 2 | No | Married | 100K | No | | |
| 3 | No | Single | 70K | No | | |
| 4 | Yes | Married | 120K | No | | |
| 5 | No | Divorced | 95K | Yes | | |
| 6 | No | Married | 60K | No | | |
| 7 | Yes | Divorced | 220K | No | | |
| 8 | No | Single | 85K | Yes | | |
| 9 | No | Married | 75K | No | | |
| 10 | No | Single | 90K | Yes | | |

圖 4.6 ▶ 預測償還貸款能力的訓練資料

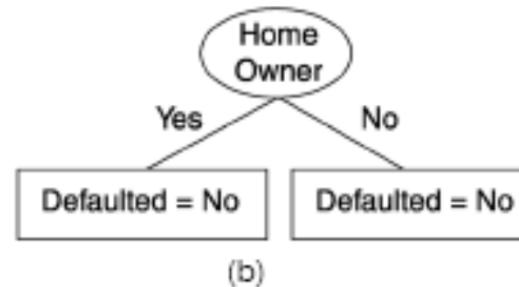
Hunt's 演算法 (以預測貸款者為例)

- Hunt's 演算法所產生的決策樹

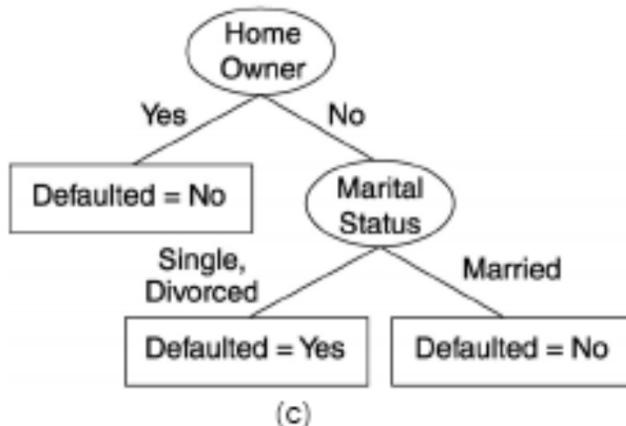
表示貸款者
有如期歸還

Defaulted = No

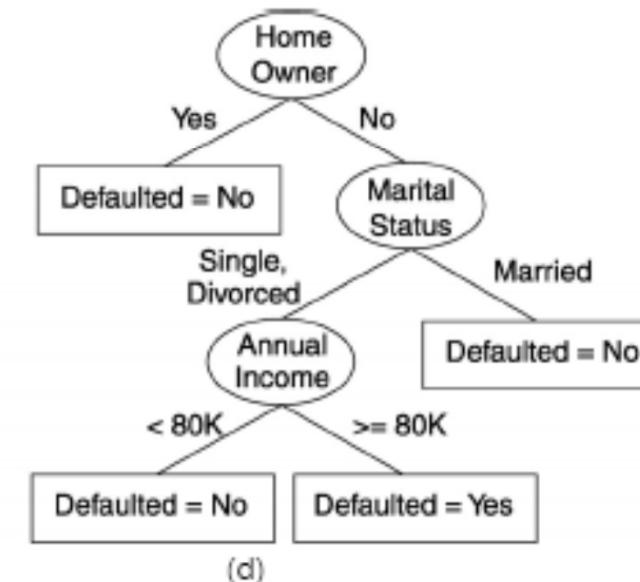
(a)



(b)



(c)



(d)

設計決策樹的問題

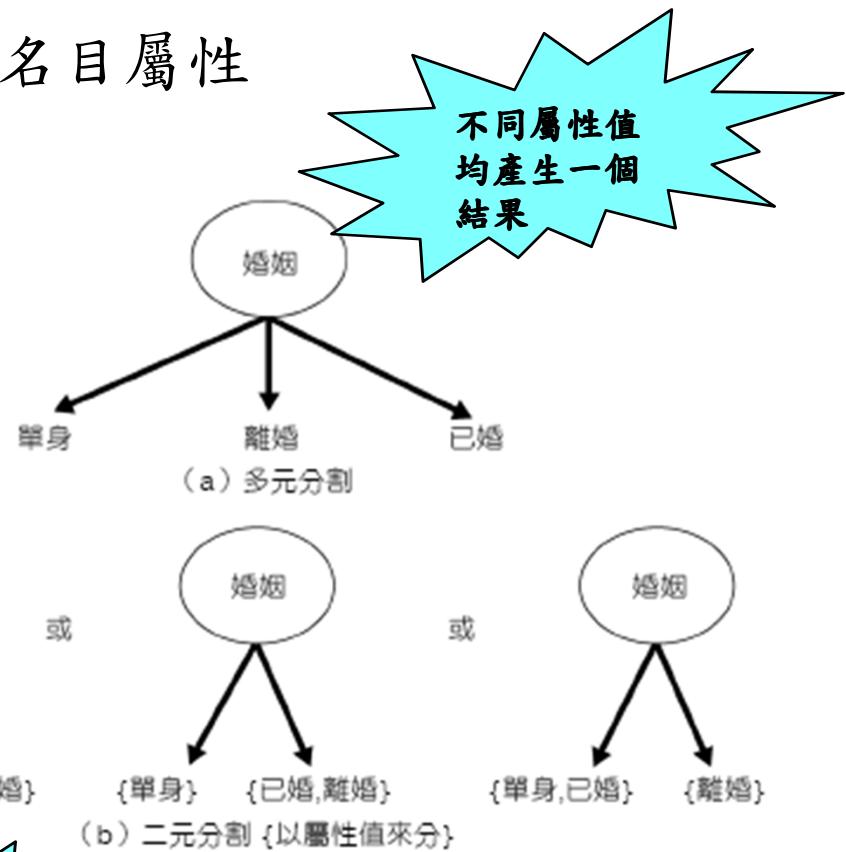
- 決策樹演算法要能處理下列問題
 - 訓練資料如何分割?
 - ◆ 演算法要提供一個方法處理不同屬性型態的測試條件
 - 分割何時停止?
 - ◆ 終止樹的成長

屬性測試條件的表示方法(I)

- 二元屬性



- 名目屬性

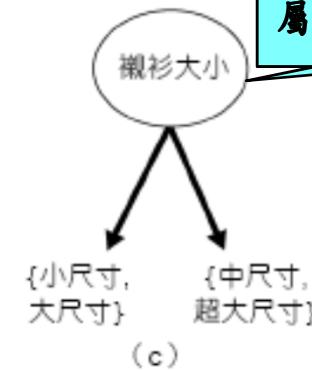
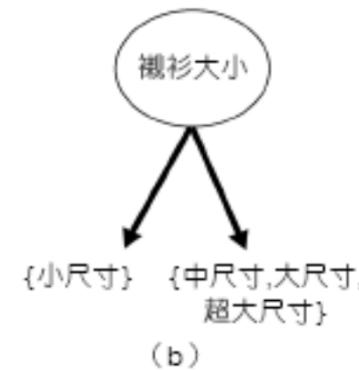


CART演算
法即屬於此
種

屬性測試條件的表示方法(II)

● 順序屬性

順序屬性值可以被分群，但在分群時必須保持原本屬性值之順序

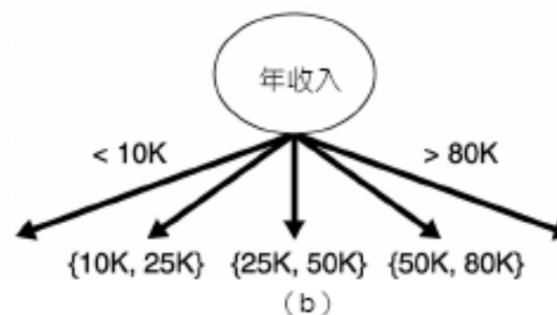
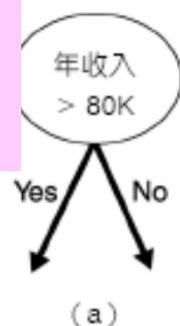


此法違反了原有的屬性順序

● 連續性屬性

需考量所有可能可以進行分割的點，並找出最好的分割點

注意 仍需保持順序性



如何選擇最好的分割點(I)

- 在分割前後，由類別的分配情形來決定

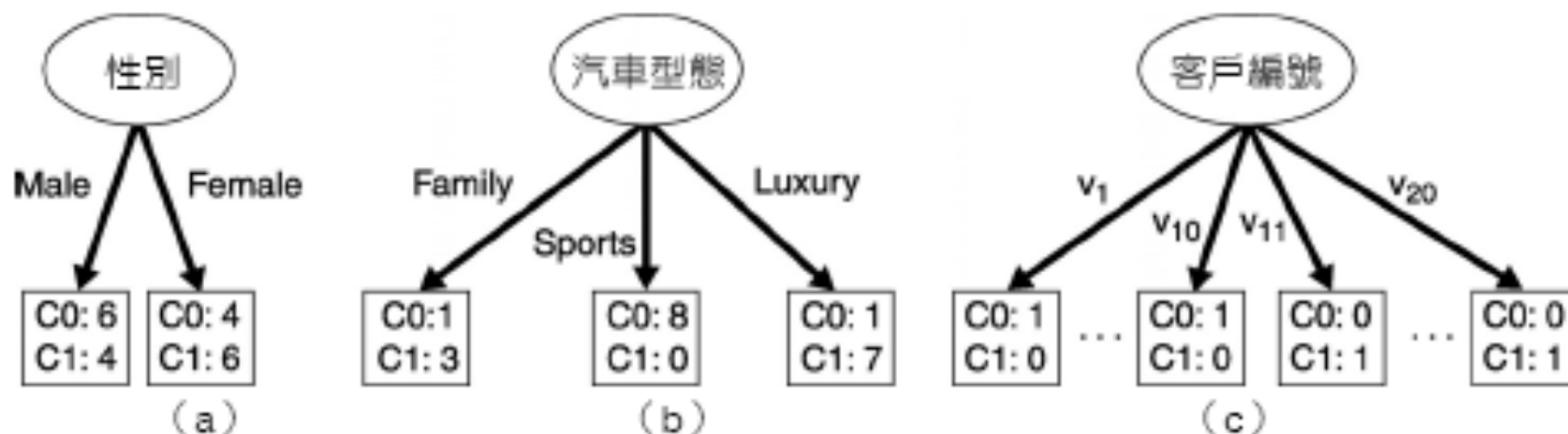


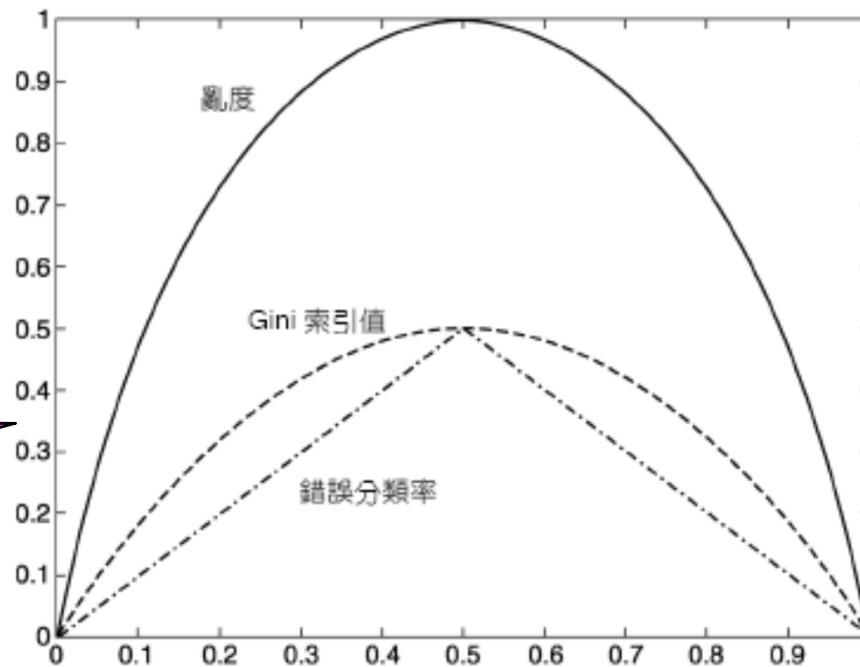
圖 多元分割與二元分割

如何選擇最好的分割點(II)

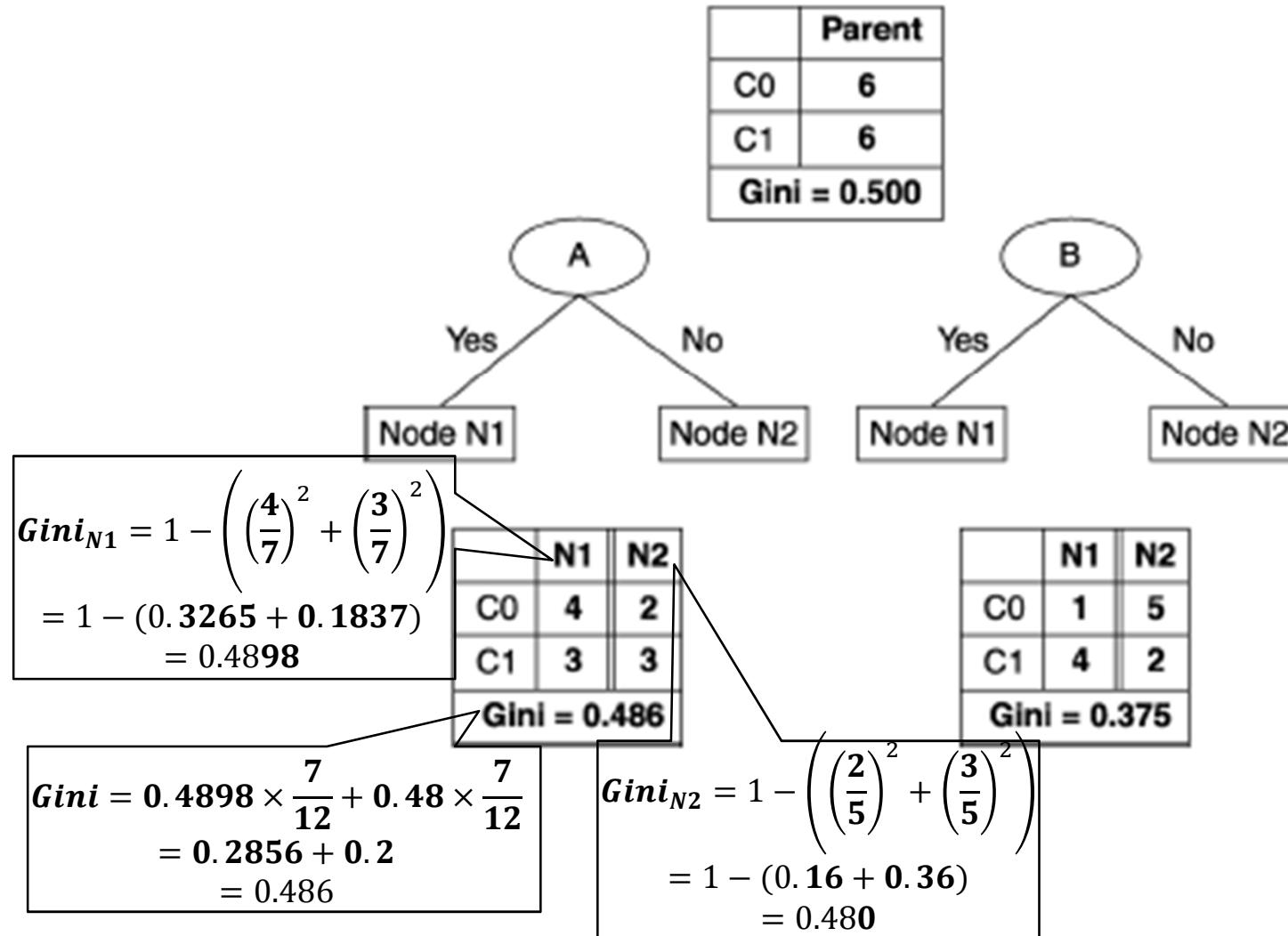
● 不純程度 (degree of impurity) 的衡量

- 亂度 = $-\sum_{i=0}^{c-1} p(i|t) \log_2 p(i|t)$ C表類別個數
- Gini 索引值 = $1 - \sum_{i=0}^{c-1} [p(i|t)]^2$ p表類別比例
- 錯誤分類率 = $1 - \max_i[p(i|t)]$

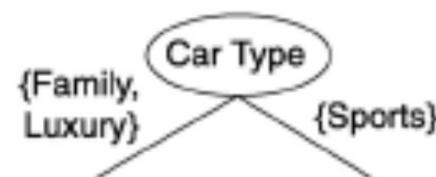
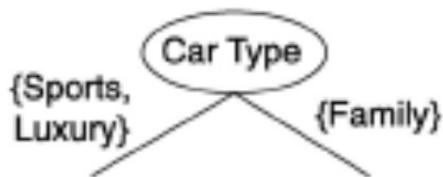
不純程度愈小，其類別分配將愈歪斜



二元屬性的分割



名目屬性的分割



| Car Type | | |
|----------|---------------------|----------|
| | {Sports, Luxury} | {Family} |
| C0 | 9 | 1 |
| C1 | 7 | 3 |
| Gini | 0.468 | |

| Car Type | | |
|----------|----------|---------------------|
| | {Sports} | {Family, Luxury} |
| C0 | 8 | 2 |
| C1 | 0 | 10 |
| Gini | 0.167 | |

(a) 二元分割

| Car Type | | | |
|----------|--------|--------|--------|
| | Family | Sports | Luxury |
| C0 | 1 | 8 | 1 |
| C1 | 3 | 0 | 7 |
| Gini | 0.163 | | |

(b) 多元分割

連續屬性的分割

| Class | No | No | No | Yes | Yes | Yes | Yes | No | No | No | No | No |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 年收入 | | | | | | | | | | | | |
| 排序值 → | 60 | 70 | 75 | 85 | 90 | 95 | 100 | 120 | 125 | 220 | | |
| 分割值 → | 55 | 65 | 72 | 80 | 87 | 92 | 97 | 110 | 122 | 172 | 230 | |
| | <= | > | <= | > | <= | > | <= | > | <= | > | <= | > |
| Yes | 0 | 3 | 0 | 3 | 0 | 3 | 1 | 2 | 2 | 1 | 3 | 0 |
| No | 0 | 7 | 1 | 6 | 2 | 5 | 3 | 4 | 3 | 4 | 3 | 4 |
| Gini | 0.420 | 0.400 | 0.375 | 0.343 | 0.417 | 0.400 | 0.300 | 0.343 | 0.375 | 0.400 | 0.420 | 0.420 |

為降低運算複雜度，可將年收入進行排序再以相鄰兩值之中間值進行測試

獲利率

- 利用獲利率的分割條件來決定分割的好壞

$$\text{獲利率} = \frac{\Delta_{\text{info}}}{\text{Split Info}}$$

$$\text{Split Info} = - \sum_{i=1}^k P(v_i) \log_2 P(v_i)$$

決策樹演算法

演算法 4.1 基本的決策樹演算法

TreeGrowth (E, F)

演算法的輸入值為E，F為屬性集合

```
1: if stopping cond(E,F) = true then  
2:   leaf = createNode().  
3:   leaf.label = Classify(E).  
4:   return leaf.  
5: else  
6:   root = createNode().  
7:   root.test cond = find best split(E, F).    演算法將遞迴式的選取最好的分割屬性  
8:   let V = {v | v is a possible outcome of root.test cond }.  
9:   for each v ∈ V do  
10:    Ev = {e | root.test cond(e) = v and e ∈ E}.  
11:    child = TreeGrowth(Ev, F).  
12:    add child as descendent of root and label the edge (root → child) as v.  
13: end for  
14: end if  
15: return root.
```

範例：網頁機器人偵測

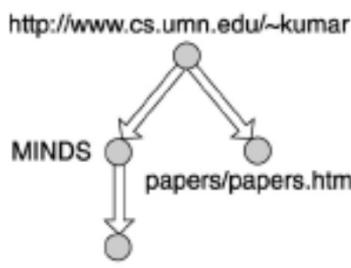
- 探討如何區分使用者以及網站機器人所存取的資訊

網站機器人的輸入資料

用決策樹偵測網站機器人

| Session | IP Address | Timestamp | Request Method | Requested Web Page | Protocol | Status | Number of Bytes | Referrer | User Agent |
|---------|--------------|----------------------|----------------|---|----------|--------|-----------------|------------------------------------|--|
| 1 | 160.11.11.11 | 08/Aug/2004 10:15:21 | GET | http://www.cs.umn.edu/~kumar | HTTP/1.1 | 200 | 6424 | | Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0) |
| 1 | 160.11.11.11 | 08/Aug/2004 10:15:34 | GET | http://www.cs.umn.edu/~kumar/MINDS | HTTP/1.1 | 200 | 41378 | http://www.cs.umn.edu/~kumar | Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0) |
| 1 | 160.11.11.11 | 08/Aug/2004 10:15:41 | GET | http://www.cs.umn.edu/~kumar/MINDS/MINDS_papers.htm | HTTP/1.1 | 200 | 1018516 | http://www.cs.umn.edu/~kumar/MINDS | Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0) |
| 1 | 160.11.11.11 | 08/Aug/2004 10:16:11 | GET | http://www.cs.umn.edu/~kumar/papers/papers.html | HTTP/1.1 | 200 | 7463 | http://www.cs.umn.edu/~kumar | Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0) |
| 2 | 35.9.2.2 | 08/Aug/2004 10:16:15 | GET | http://www.cs.umn.edu/~steinbac | HTTP/1.0 | 200 | 3149 | | Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7) Gecko/20040618 |

(a) 網站伺服器日誌的例子



(b) 網頁會期的圖形

| Attribute Name | Description |
|----------------|---|
| totalPages | Total number of pages retrieved in a Web session |
| ImagePages | Total number of image pages retrieved in a Web session |
| TotalTime | Total amount of time spent by Web site visitor |
| RepeatedAccess | The same page requested more than once in a Web session |
| ErrorRequest | Errors in requesting for Web pages |
| GET | Percentage of requests made using GET method |
| POST | Percentage of requests made using POST method |
| HEAD | Percentage of requests made using HEAD method |
| Breadth | Breadth of Web traversal |
| Depth | Depth of Web traversal |
| MultiIP | Session with multiple IP addresses |
| MultiAgent | Session with multiple user agents |

(c) 網站機器人偵測所得到的屬性

Decision Tree:

```
depth = 1:  
| breadth > 7 : class 1  
| breadth <= 7:  
| | breadth <= 3:  
| | | ImagePages > 0.375: class 0  
| | | ImagePages <= 0.375:  
| | | | totalPages <= 6: class 1  
| | | | totalPages > 6:  
| | | | | breadth <= 1: class 1  
| | | | | breadth > 1: class 0  
| | width > 3:  
| | MultiIP = 0:  
| | | ImagePages <= 0.1333: class 1  
| | | ImagePages > 0.1333:  
| | | | breadth <= 6: class 0  
| | | | breadth > 6: class 1  
| | | MultiIP = 1:  
| | | | TotalTime <= 361: class 0  
| | | | TotalTime > 361: class 1  
depth > 1:  
| MultiAgent = 0:  
| | depth > 2: class 0  
| | depth < 2:  
| | | MultiIP = 1: class 0  
| | | MultiIP = 0:  
| | | | breadth <= 6: class 0  
| | | | breadth > 6:  
| | | | | RepeatedAccess <= 0.322: class 0  
| | | | | RepeatedAccess > 0.322: class 1  
| | | MultiAgent = 1:  
| | | | totalPages <= 81: class 0  
| | | | totalPages > 81: class 1
```

決策樹的特性

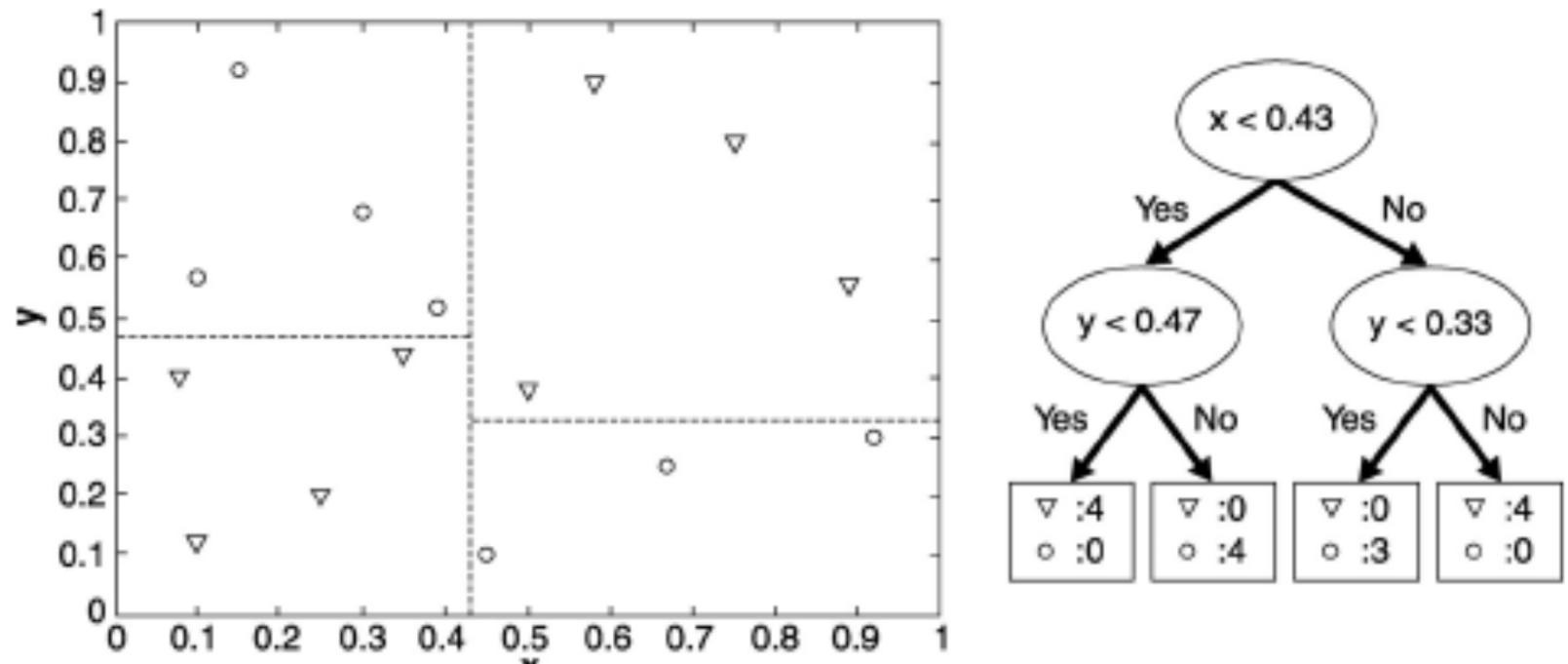
1. 是用無母數來建立分類模式的方法(即不需要滿足任何機率分配)
2. 多用經驗法則在大量的假設空間中進行搜尋(因為要找到最佳解是NP-Complete 的問題)
3. 資料量大時建構決策樹不難，且速度也不慢
4. 較小的樹，解釋較為容易，樹的正確性可和其他分類技術進行比較
5. 提供一個學習離散值函數的表示方式

決策樹的特性

6. 決策樹演算法可處理雜訊值問題，避免過度學習
7. 重複的屬性並不會影響決策樹的正確性
8. 資料分割的問題(葉節點中，可能會因資料量太少而無法達到統計的顯著性)
9. 子樹可在決策樹中重複多次(有可能變得不容易解釋)
10. 決策界限的問題
11. 不純程度測量公式的選擇對決策樹分類結果影響不大

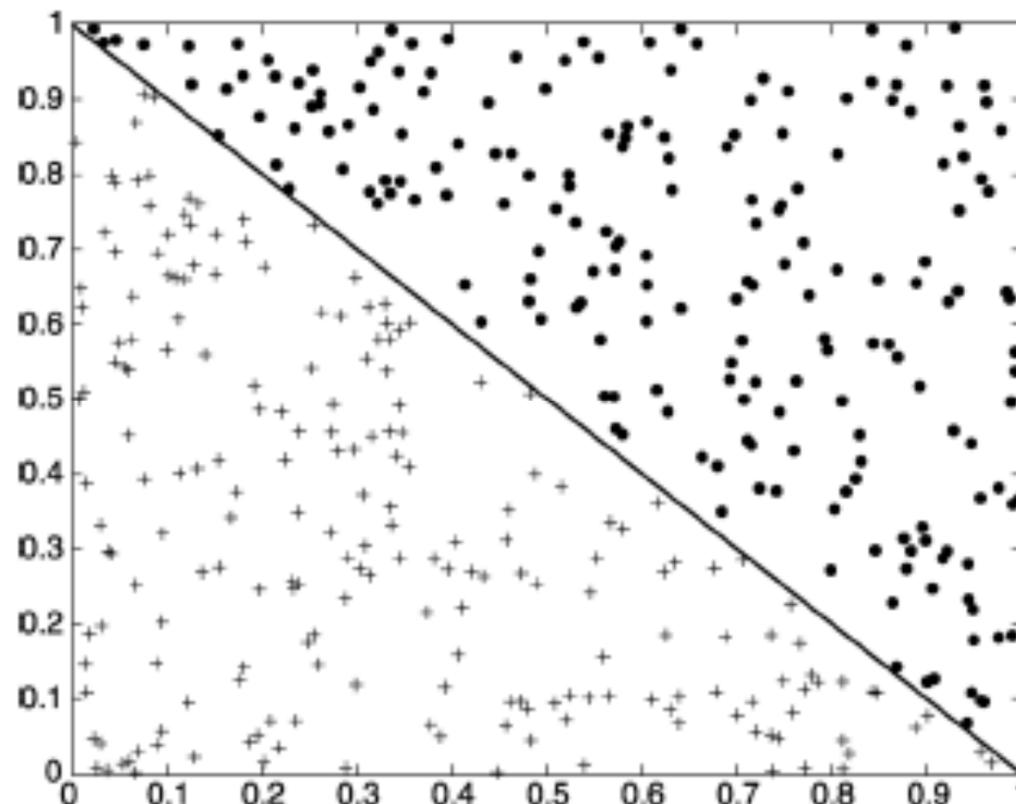
決策界限

- 二維資料的決策樹及其決策界限



範例：

- 無法用單一屬性作為測試條件來進行資料分割

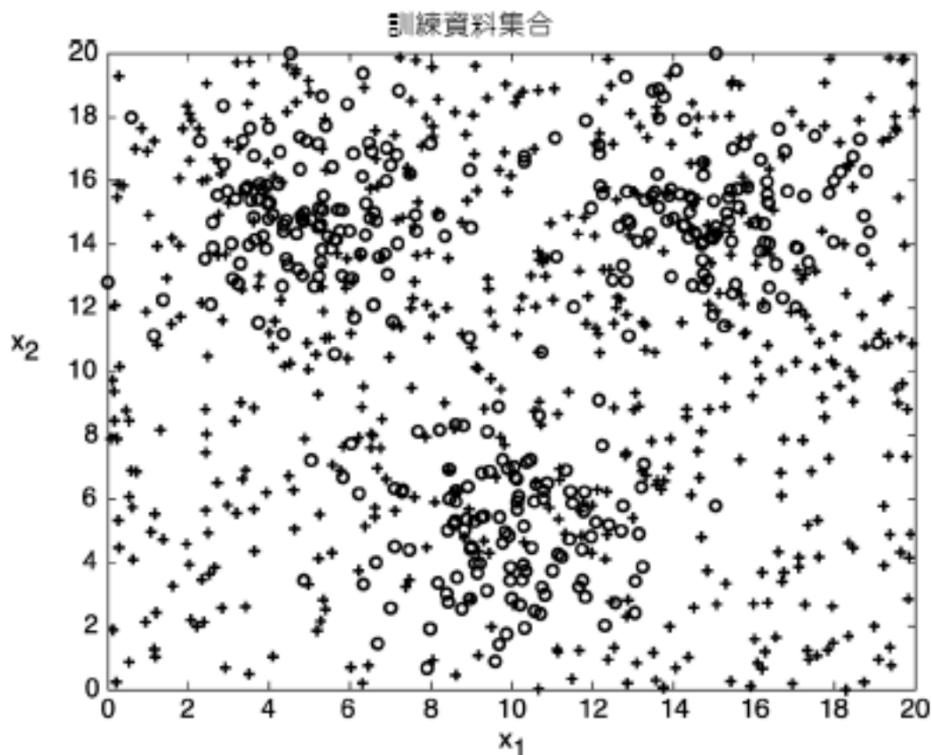


分類模式的錯誤

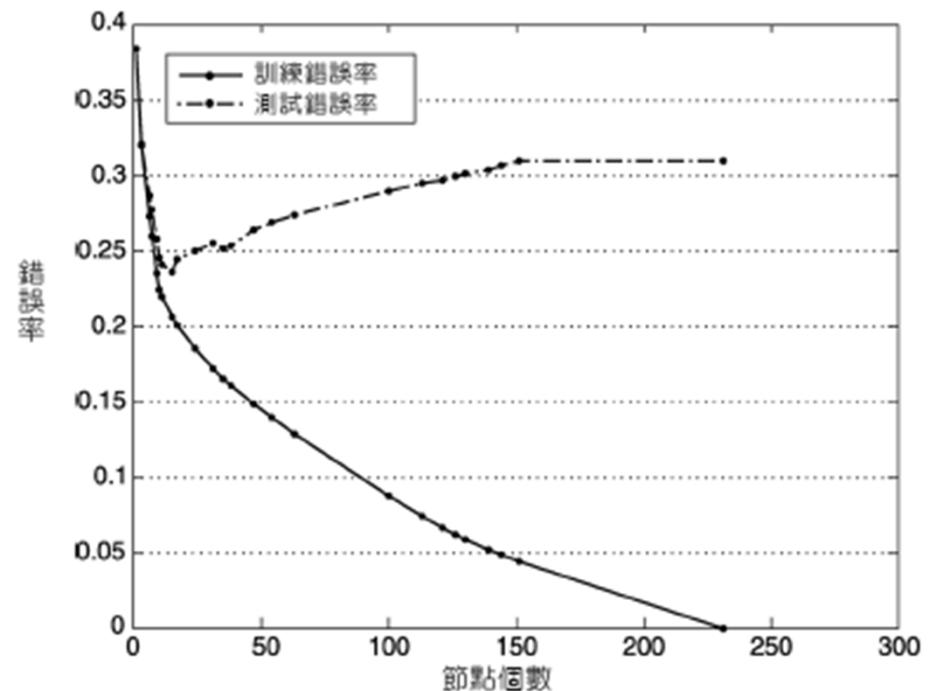
- 訓練錯誤：即重新帶入錯誤（resubstitution error）或是表面錯誤（apparent error），指訓練資料被誤判的個數
 - 過度學習（overfitting）：樹太大時，其測試錯誤率就會開始增加
 - 學習不足（underfitting）：樹太小時，其模式的訓練及測試的錯誤率會變得很大
- 推論錯誤：期望模式能夠推論至未見過資料的程度

範例：二維資料過度學習

- o : 1200 筆資料，+ : 1800 筆
- 30%訓練資料，70%測試資料



- 訓練及測試錯誤率



過度學習

- 樹太大時，其測試錯誤率就會開始增加
- 雜訊值對過度學習的影響
- 缺乏代表性的樣本對過度學習的影響
- 過度學習VS.多重比較法
- 避免的方法
 - 預先修剪（prepruning）：可用在發展決策樹的過程中，一方面可完全學習訓練資料，一方面可避免在過度學習的情形下先停止學習
 - 事後修剪（post-pruning）：其決策樹可以任意發展，待決策樹建立完成後，再將不必要或多餘的分支修剪掉

推論錯誤的估計

- 使用重新帶入估計
- 加上模式複雜度
 - Occam's Razor：在兩個相同推論錯誤率的模式下，愈簡單的模式愈好
- 估計統計的界限
- 使用驗證資料

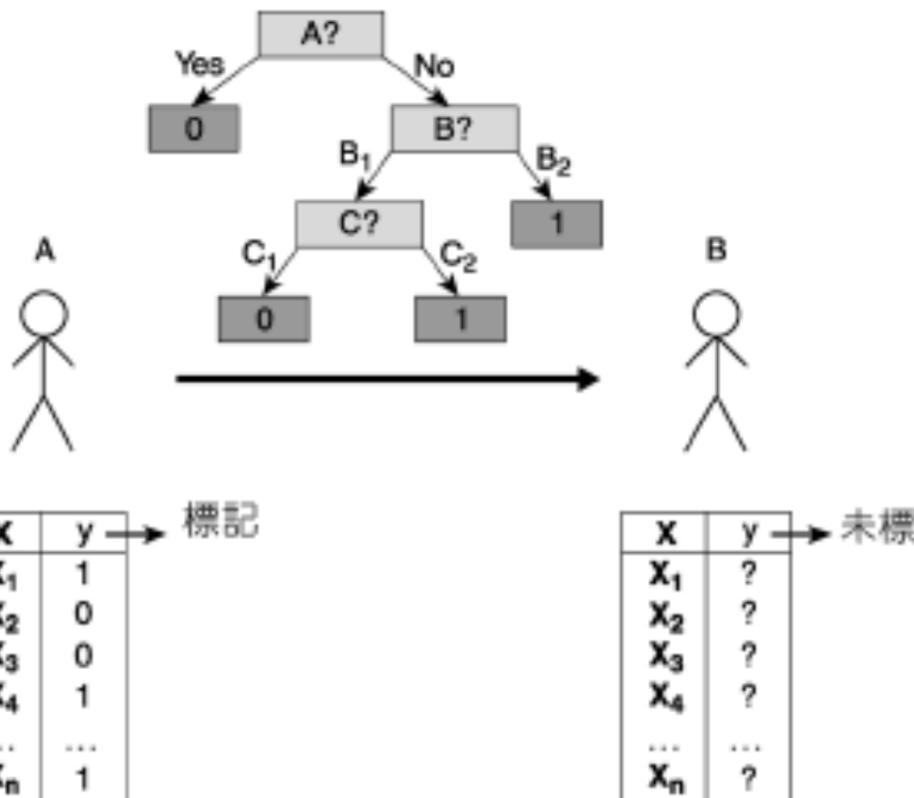
加上模式複雜度

- 悲觀錯誤率的估計：考量訓練錯誤的總和，以及將模式複雜度的懲罰值作為推論錯誤率的計算項目
- 例如 $n(t)$ 是節點 t 的訓練資料個數，而 $e(t)$ 是誤判的個數，用悲觀法來估計決策樹 T ，其 $eg(T)$ 計算如下：

$$eg(T) = \frac{\sum_{i=1}^k [e(t_i) + \Omega(t_i)]}{\sum_{i=1}^k n(t_i)} = \frac{e(T) + \Omega(T)}{N_t}$$

加上模式複雜度

- 最小描述長度原則：以資訊理論為主的最小描述長度原則
(minimum description length principle, MDL principle)



分類技術的評估

1. 保持（holdout）方法：在保持（holdout）方法中，原始的資料將被分成二個部分，稱為訓練集與測試集，而分類模式之後會從訓練資料中形成，然後再用測試資料來進行評估
2. 隨機次抽樣（subsampling）：holdout 方法可以重複多次，以改善分類技術效果的估計
3. 交叉驗證（cross-validation）：資料的訓練次數是相同的，取一半資料來訓練，剩下做為測試資料，然後二個資料的角色互換
4. 重抽法（bootstrap）：其訓練資料是要放回的，也就是再次抽樣的機率是相同的

範例4.4

- 假設其模式在評估100 筆測試資料後，得到80%正確性，在95%信心水準下，其正確性的信賴區間為何？95%信心水準相當於
- 將此項放入公式
$$\frac{2 \times N \times acc + Z_{\alpha/2}^2 \pm Z_{\alpha/2} \sqrt{Z_{\alpha/2}^2 + 4Nacc - 4Nacc^2}}{2(N + Z_{\alpha/2}^2)}$$
- 將產生介於71.1%–86.7%的信賴區間，下表是 N 筆料下的信賴區間

| N | 20 | 50 | 100 | 500 | 1000 | 5000 |
|------|------------------|------------------|------------------|------------------|------------------|------------------|
| 信賴區間 | 0.584 - 0.919 | 0.670 - 0.888 | 0.711 - 0.867 | 0.763 - 0.833 | 0.774 - 0.824 | 0.789 - 0.811 |

- 注意：當 N 增加時，其信賴區間的寬度變得較小

範例4.5

- 假設 MA 的錯誤率 $e1 = 0.15$ ，其測試資料 $N1 = 30$ ； MB 的錯誤率 $e2 = 0.25$ ，其測試資料 $N2 = 5000$ ，兩者錯誤率的差為 $d = |0.15 - 0.25| = 0.1$
- 用雙尾檢定是否 $d_t = 0$ 或是不等於 0，其變異數的估計如下：

$$\hat{\sigma}_d^2 = \frac{0.15(1 - 0.15)}{30} + \frac{0.25(1 - 0.25)}{5000} = 0.0043$$

- $\hat{\sigma}_d = 0.0655$ ，帶入 $d_t = d \pm z_{\alpha/2} \hat{\sigma}_d$ ，將可以得在 95% 信心水準下 d_t 的信賴區間
- 其區間的寬度為 0，其在 95% 信心水準下，兩者是無顯著差異

範例4.6

- 假設兩個分類技術的模式正確性之估計差的平均數為0.05，標準差為0.002，如果是用30次交叉驗證法來估計正確性，那麼在95%信心水準下，其正確性的差為：

$$d_t^{cv} = 0.05 \pm 2.04 \times 0.002$$

- 因為其信賴區間的寬度不為0，所以其差異是達統計上的顯著水準