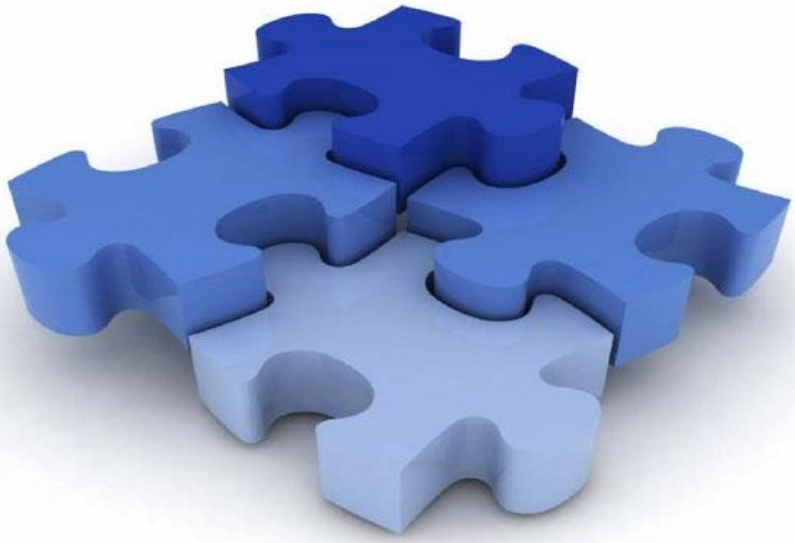
Chương 4:

Phân lớp dữ liệu (Data Classification)



Nội dung

1. Phân lớp và dự đoán?
2. Quy nạp trên cây quyết định
3. Phân lớp Bayes
4. Các phương pháp phân lớp khác

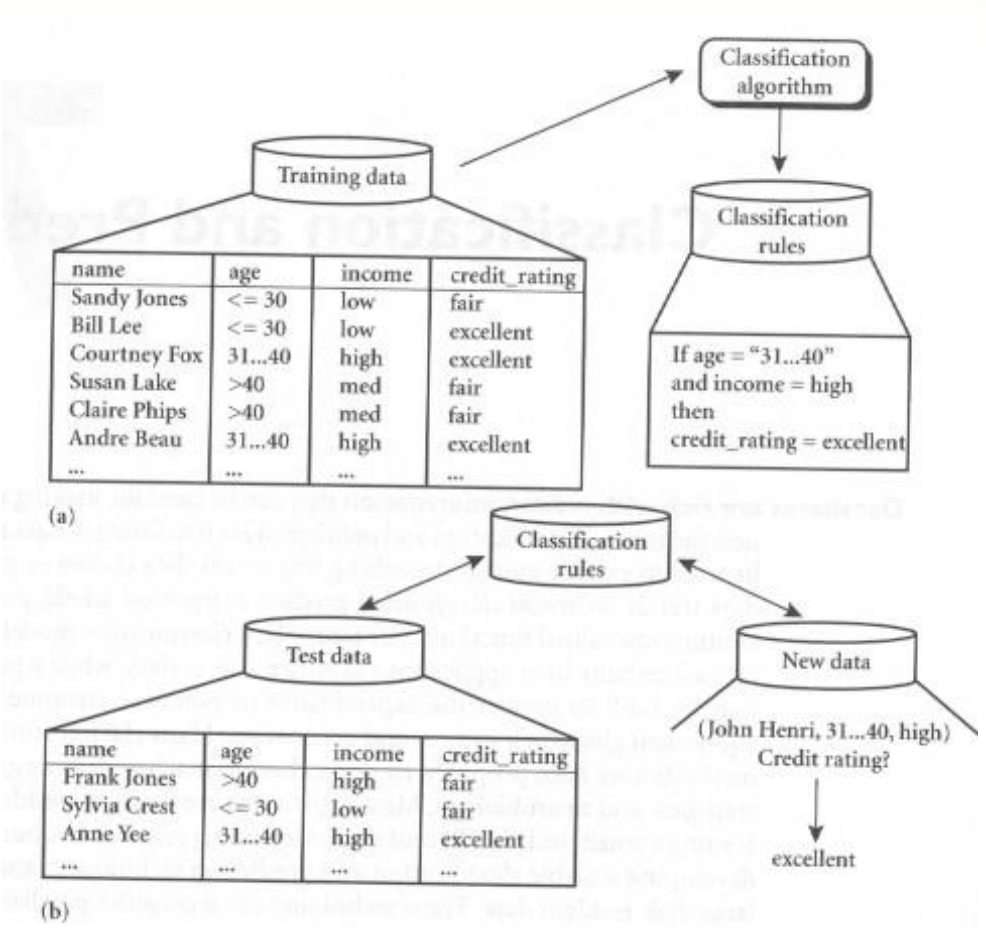
Phân lớp là gì ? Dự đoán là gì?

- Có thể dùng phân lớp và dự đoán để xác lập mô hình/mẫu nhằm mô tả các lớp quan trọng hay dự đoán khuynh hướng dữ liệu trong tương lai.
- Phân lớp(classification) dự đoán các nhãn phân loại.
- Dự đoán (prediction) hàm giá trị liên tục.

Phân lớp và Dự đoán

Phân lớp dữ liệu là tiến trình có 2 bước

- **Huấn luyện**: Dữ liệu huấn luyện được phân tích bởi thuật tóan phân lớp (có thuộc tính nhãn lớp)
- **Phân lớp:** Dữ liệu kiểm tra được dùng để ước lượng độ chính xác của bộ phân lớp. Nếu độ chính xác là chấp nhận được thì có thể dùng bộ phân lớp để phân lớp các mẫu dữ liệu mới.



Phân lớp và Dự đoán?

- Độ chính xác (accuracy) của bộ phân lớp trên tập kiểm tra cho trước là phần trăm của các mẫu trong tập kiểm tra được bộ phân lớp xếp lớp đúng

$$\text{Accuracy} = \frac{\text{correctly classified test sample}}{\text{total number of test sampl}}$$

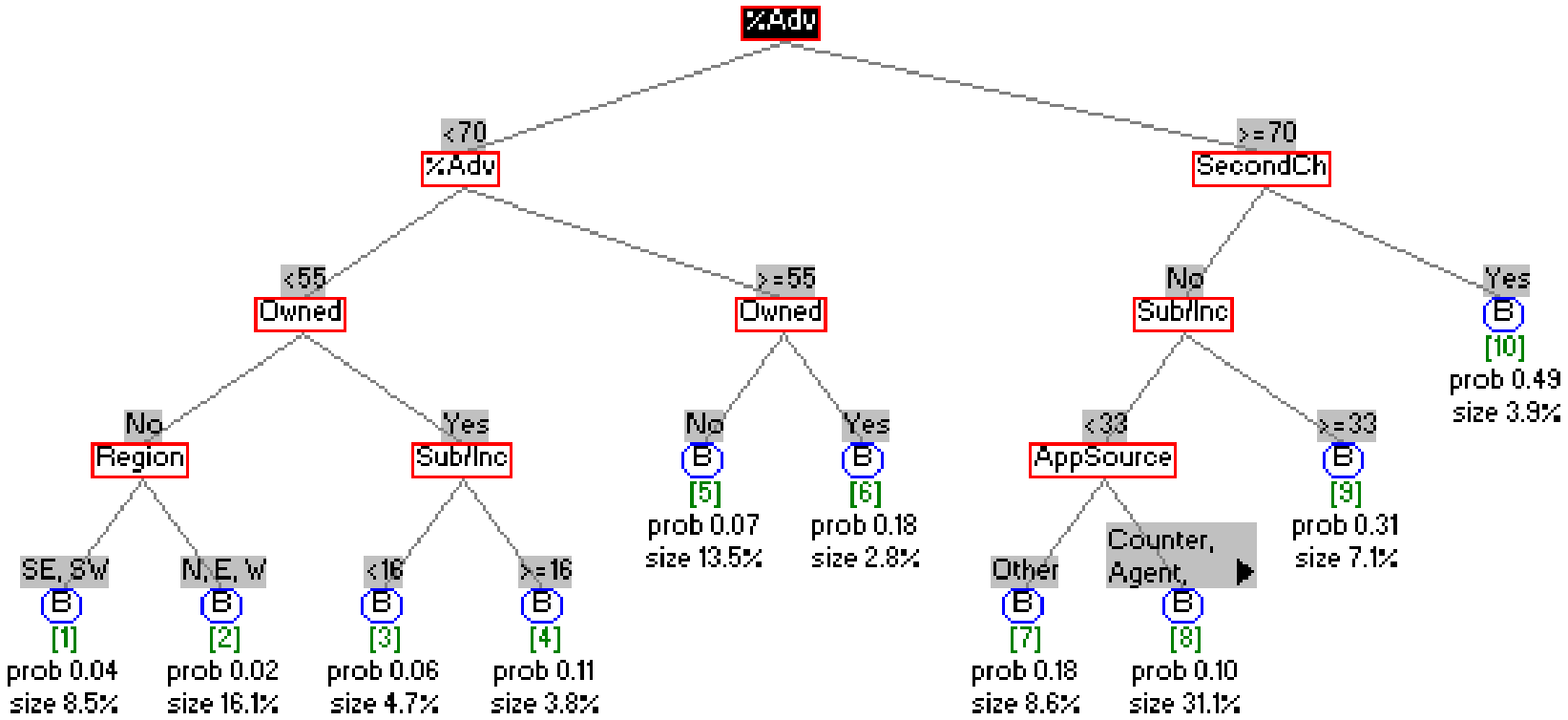
Chuẩn bị dữ liệu

- Làm sách dữ liệu
  - Nhiễu
  - Thiếu giá trị
- Phân tích liên quan (chọn đặc trưng)
  - Các thuộc tính không liên quan
  - Các thuộc tính dư thừa
- Biến đổi dữ liệu

So sánh các phương pháp phân lớp

- Độ chính xác của dự đoán: khả năng bộ phân lớp dự đoán đúng dữ liệu chưa thấy
- Tính bền vững: khả năng của bộ phân lớp thực hiện dự đoán đúng với dữ liệu có nhiễu hay thiếu giá trị
- Tính kích cỡ (scalability): khả năng tạo bộ phân lớp hiệu quả với số lượng dữ liệu lớn
- Khả năng diễn giải: bộ phân lớp cung cấp tri thức có thể hiểu được

Cây quyết định



Cây quyết định

- Cây quyết định là cấu trúc cây sao cho:
- Mỗi nút trong ứng với một phép kiểm tra trên một thuộc tính
- Mỗi nhánh biểu diễn kết quả phép kiểm tra
- Các nút lá biểu diễn các lớp hay các phân bố lớp
- Nút cao nhất trong cây là nút gốc.

Cây quyết định

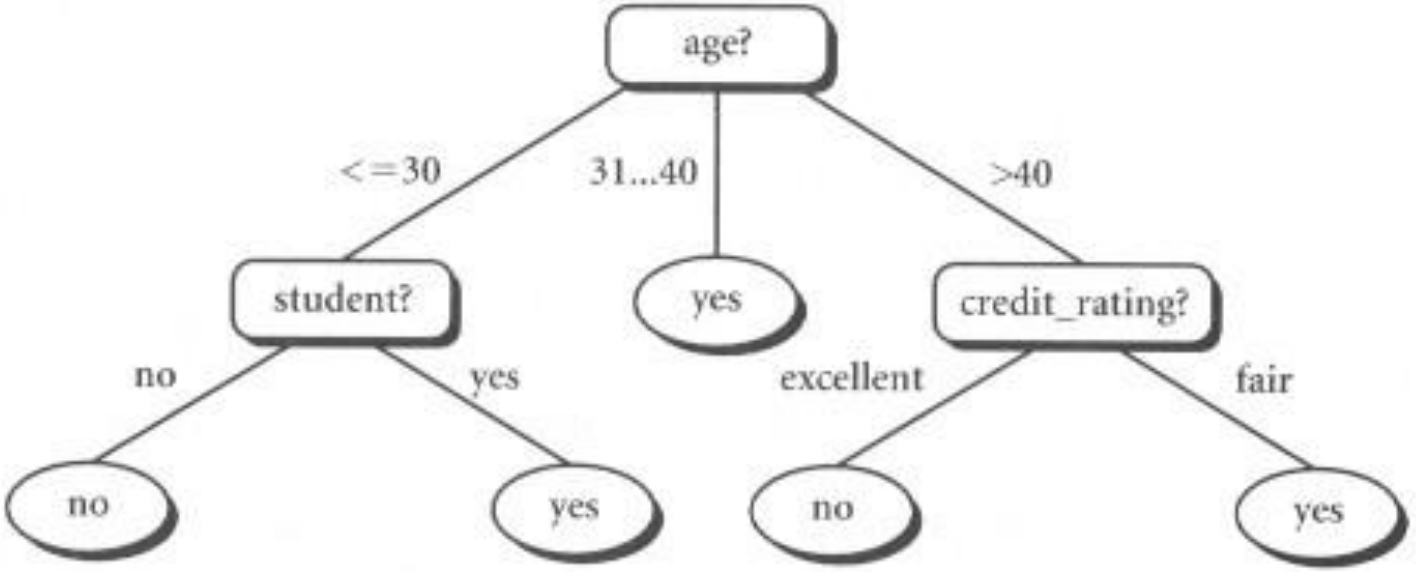
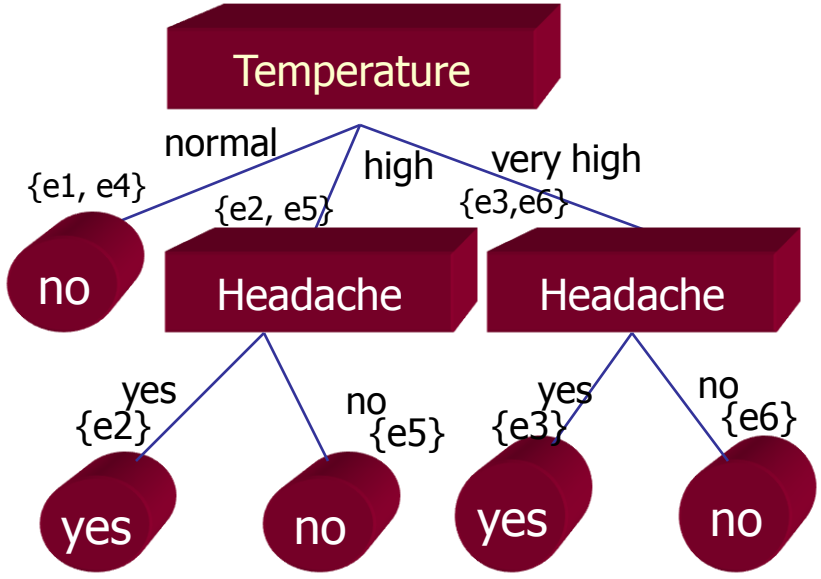


Figure 7.2 A decision tree for the concept *buys_computer*, indicating whether or not a customer at *AllElectronics* is likely to purchase a computer. Each internal (nonleaf) node represents a test on an attribute. Each leaf node represents a class (either *buys_computer* = *yes* or *buys_computer* = *no*).

Quy nạp trên cây quyết định

1. Chọn thuộc tính "tốt nhất" theo một độ đo chọn lựa cho trước
2. Mở rộng cây bằng cách thêm các nhánh mới cho từng giá trị thuộc tính
3. Sắp xếp các ví dụ học vào nút lá
4. Nếu các ví dụ được phân lớp rõ Thì Stop ngược lại lặp lại các bước 1-4 cho các nút lá
5. Tỉa các nút lá không ổn định

	Headache	Temperature	Flu
e1	yes	normal	no
e2	yes	high	yes
e3	yes	very high	yes
e4	no	normal	no
e5	no	high	no
e6	no	very high	no



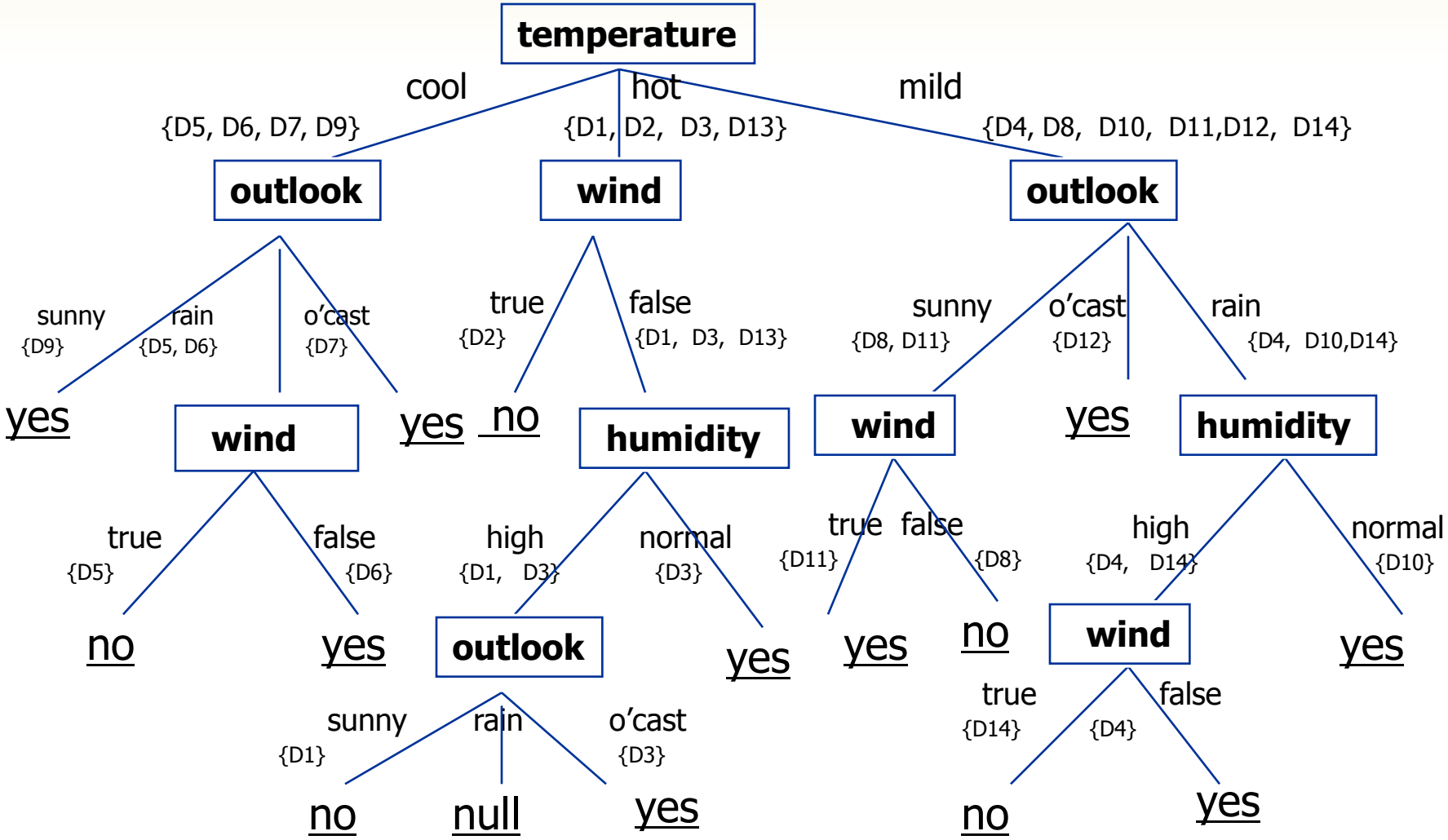
Chiến lược cơ bản

- Bắt đầu từ nút đơn biểu diễn tất cả các mẫu
- Nếu các mẫu thuộc về cùng một lớp, nút trở thành nút lá và được gán nhãn bằng lớp đó
- Ngược lại, dùng độ đo thuộc tính để chọn thuộc tính sẽ phân tách tốt nhất các mẫu vào các lớp
- Một nhánh được tạo cho từng giá trị của thuộc tính được chọn và các mẫu được phân hoạch theo
- Dùng đệ quy cùng một quá trình để tạo cây quyết định
- Tiến trình kết thúc chỉ khi bất kỳ điều kiện nào sau đây là đúng
  - Tất cả các mẫu cho một nút cho trước đều thuộc về cùng một lớp.
  - Không còn thuộc tính nào mà mẫu có thể dựa vào để phân hoạch xa hơn.
  - Không còn mẫu nào cho nhánh test_attribute = a_i

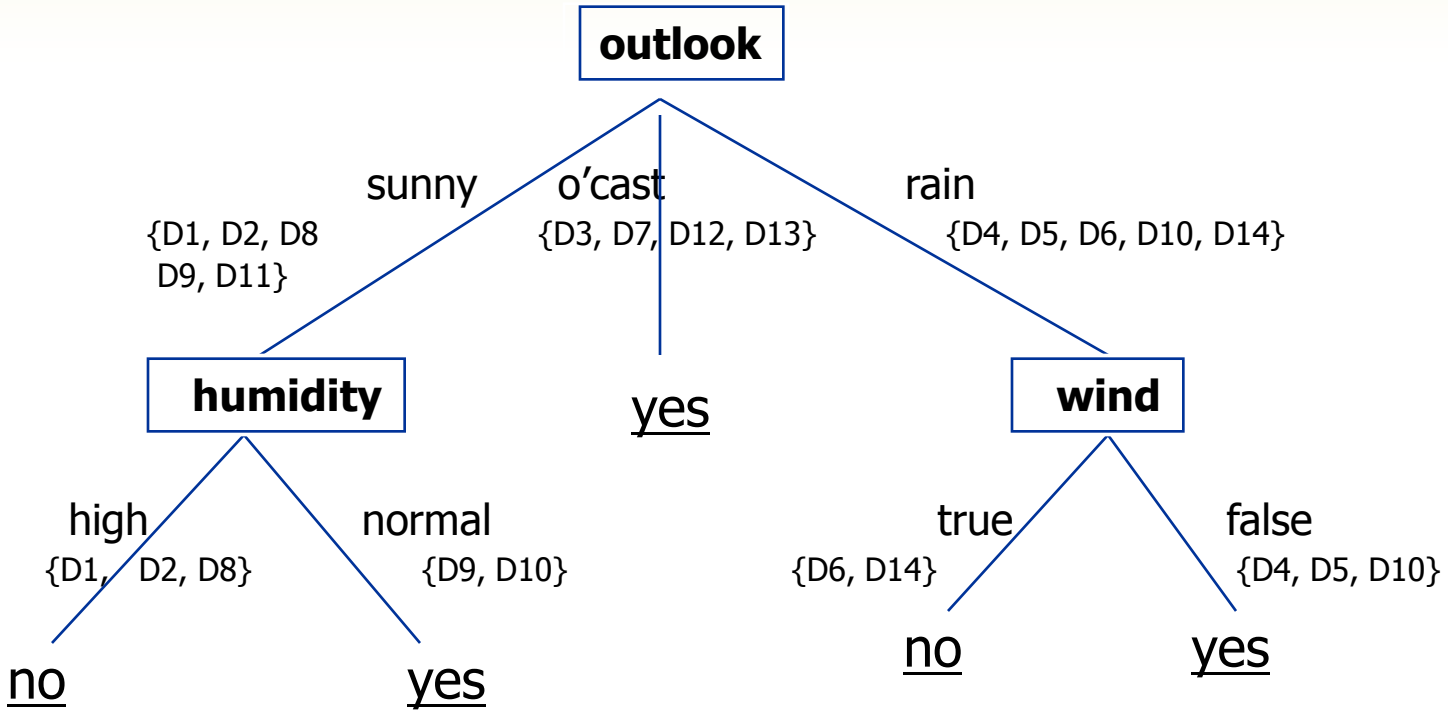
Bảng dữ liệu huấn luyện

Day	Outlook	Temp	Humidity	Wind	PlayTennis
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	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	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

Cây quyết định cho bài toán chơi tennis



Cây quyết định đơn giản



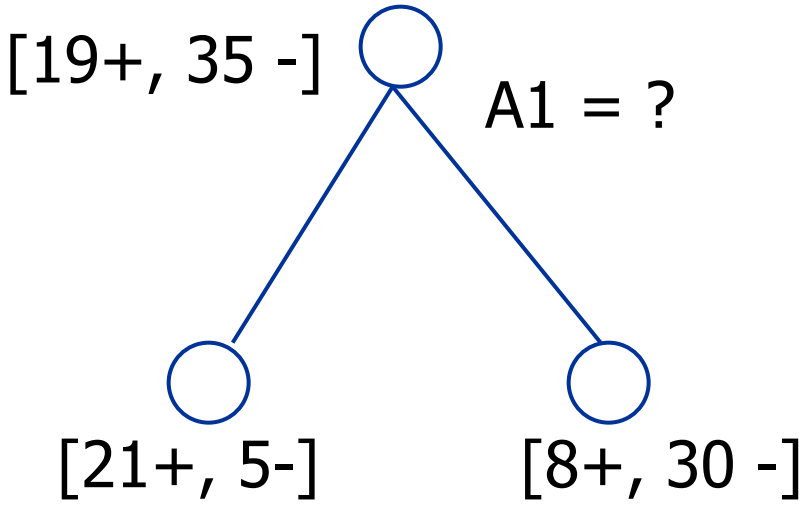
Cây sẽ đơn giản hơn nếu "outlook" được chọn làm gốc.

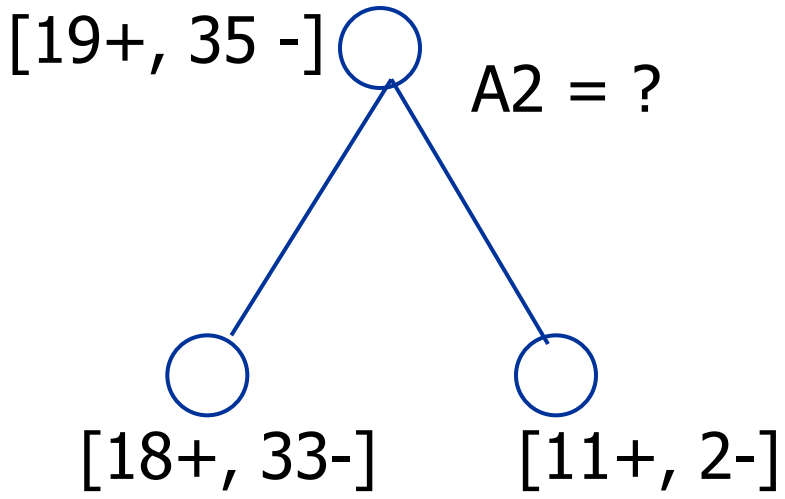
Cách chọn thuộc tính tốt để tách nút quyết định?

Thuộc tính nào là tốt nhất?

Nút quyết định S có 19 mẫu thuộc lớp cộng (+) và 35 mẫu thuộc lớp trừ (-), ta ký hiệu là [19+, 35-]

Nếu các thuộc tính A1 và A2 (mỗi thuộc tính có 2 giá trị) tách S thành các nút con với tỷ lệ của mẫu dương và mẫu âm như sau, thuộc tính nào là tốt hơn?





Entropy

Entropy đặc trưng độ bất định / hỗn tạp của tập bất kỳ các ví dụ.

S là tập các mẫu thuộc lớp âm và lớp dương

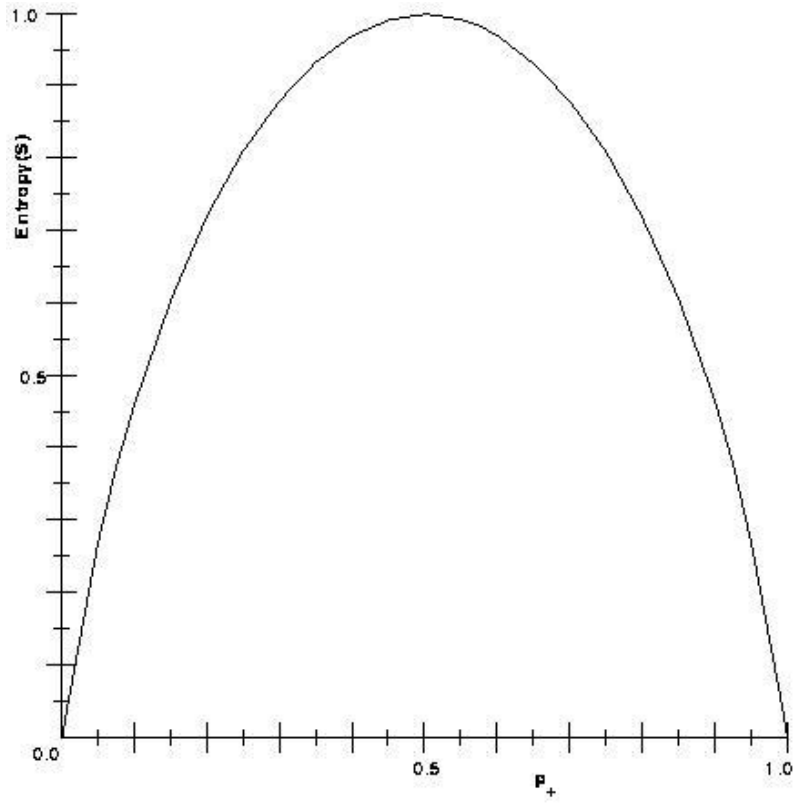
$P_{\oplus}$ là tỷ lệ các mẫu thuộc lớp dương trong S

$p_{\ominus}$ là tỷ lệ các mẫu thuộc lớp âm trong S

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

Entropy

Hàm entropy tương ứng với phân lớp boolean,khi tỷ lệ của p_+ các ví dụ thuộc lớp dương thay đổi giữa 0 và 1.



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

Ví dụ

Từ 14 mẫu của bảng Play-Tennis, 9 thuộc lớp dương và 5 mẫu âm (ký hiệu là [9+, 5-])

$$\text{Entropy}([9+, 5-]) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0.940$$

Lưu ý:

1. Entropy là 0 nếu tất cả các thành viên của S đều thuộc về cùng một lớp. Ví dụ, nếu tất cả các thành viên đều thuộc về lớp dương ($p_{\ominus}=0$) thì p là 0 và $\text{Entropy}(S) = -1.\log_2(1) - 0.\log_2(0) = -1.0 - 0 . \log_2(0) = 0$.

2. Entropy là 1 nếu tập hợp chứa số lượng bằng nhau các thành viên thuộc lớp dương và lớp âm. Nếu các số này là khác nhau, entropy sẽ nằm giữa 0 và 1.

Information Gain đo sự rút giảm mong muốn của Entropy

Ta định nghĩa độ đo information gain, phản ánh mức độ hiệu quả của một thuộc tính trong phân lớp. Đó là sự rút giảm mong muốn của entropy gây ra bởi sự phân hoạch các ví dụ theo thuộc tính này

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Value(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Gía tri Value(A) là tập các giá trị có thể cho thuộc tính A, *và* S_v là tập con của S mà A nhận giá trị v.

Information Gain đo sự rút giảm trong Entropy

Values(Wind) = {Weak, Strong}, S = [9+, 5-]

S_{weak} là nút con với trị "weak" là [6+, 2-]

S_{strong} , là nút con với trị "strong", là [3+, 3-]

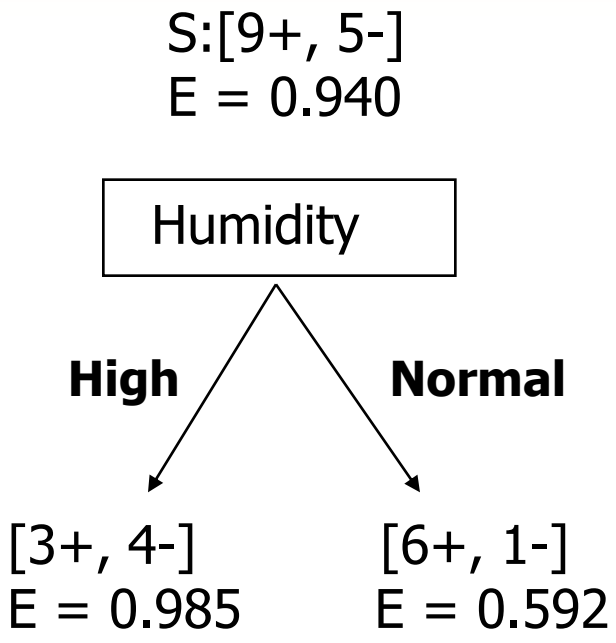
$$Gain(S, Wind) = Entropy(S) - \sum_{v \in \{Weak, Strong\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

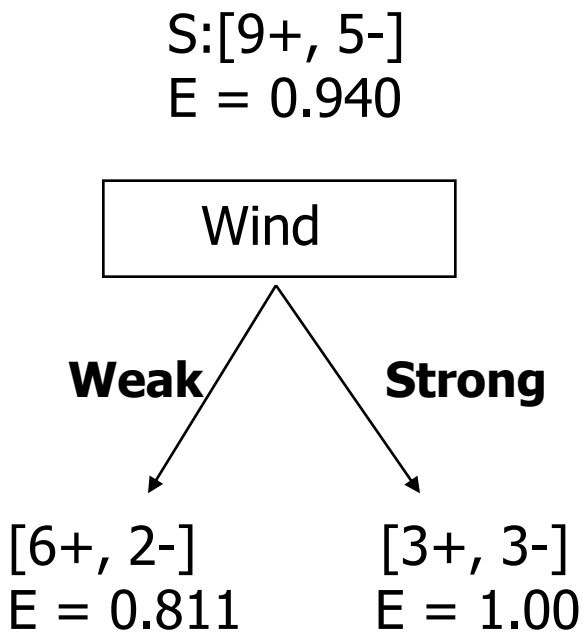
$$= Entropy(S) - (8/14)Entropy(S_{weak}) - (6/14)Entropy(S_{Strong})$$

$$= 0.940 - (8/14)0.811 - (6/14)1.00$$

$$= 0.048$$

Thuộc tính nào là phân lớp tốt nhất?





Gain(S, Humidity)
= .940 - (7/14).985 - (7/14).592
= .151

Gain(S, Wind)
= .940 - (8/14).811 - (6/14)1.00
= .048

Information gain của tất cả thuộc tính

Gain (S, Outlook) = 0.246

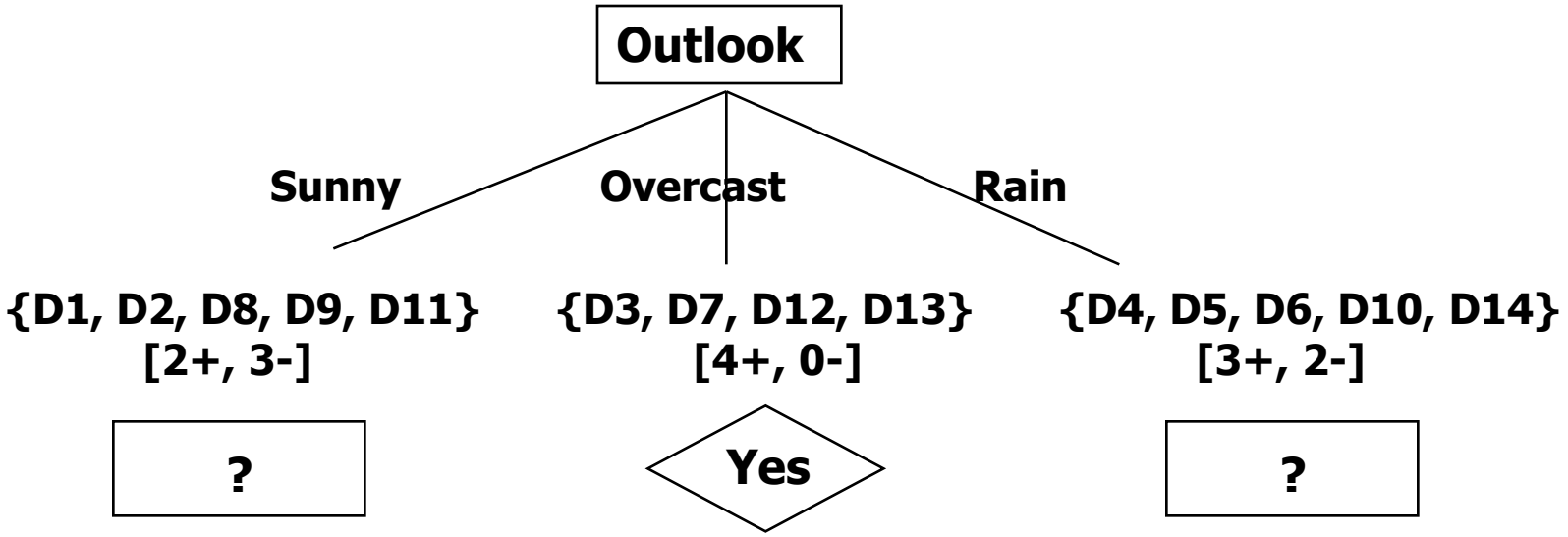
Gain (S, Humidity) = 0.151

Gain (S, Wind) = 0.048

Gain (S, Temperature) = 0.029

Bước kế tiếp trong tiến trình tăng trưởng trên cây quyết định

{D1, D2, ..., D14} [9+, 5-]



Thuộc tính nào cần được kiểm tra?

S_{sunny} = {D1, D2, D3, D9, D11}

Gain(S_{sunny}, Humidity) = .970 - (3/5)0.0 - (2/5)0.0 = 0.970

Gain(S_{sunny}, Temperature) = .970 - (2/5)0.0 - (2/5)1.0 - (1/5)0.0 = 0.570

Gain(S_{sunny}, Wind) = .970 - (2/5)1.0 - (3/5)0.918 = 0.019

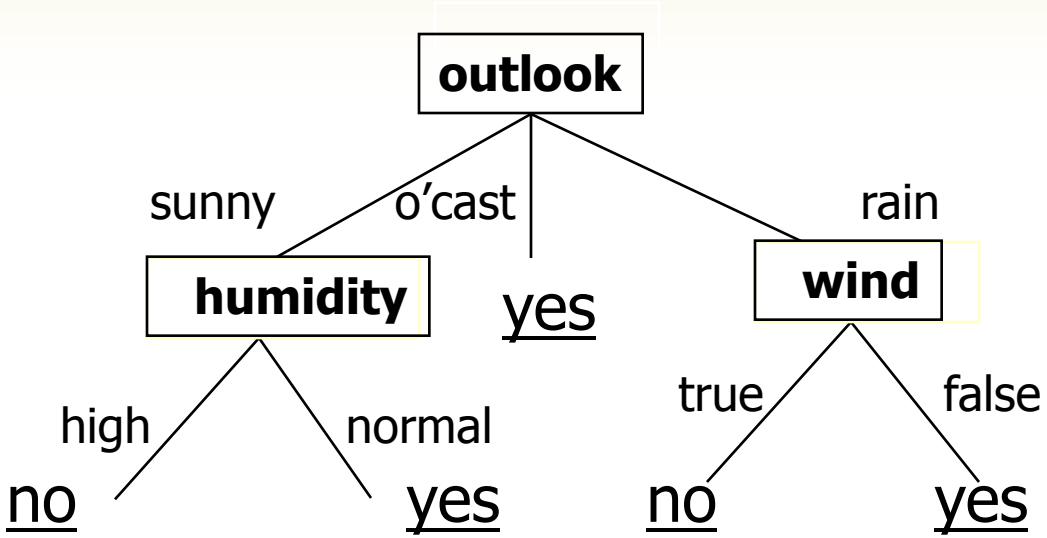
Điều kiện dừng

1. Từng thuộc tính đã được đưa vào dọc theo con đường trên cây

2. Các mẫu huấn luyện ứng với nút lá có cùng giá trị thuộc tính đích (chẳng hạn, chúng có entropy bằng zero)

Lưu ý: Thuật toán ID3 dùng Information Gain và C4.5, thuật toán được phát triển sau nó, dùng Gain Ratio (một biến thể của Information Gain)

Đổi cây thành luật



IF (Outlook = Sunny) and (Humidity = High)
THEN PlayTennis = No

IF (Outlook = Sunny) and (Humidity = Normal)
THEN PlayTennis = Yes

.....

Các thuộc tính với nhiều giá trị

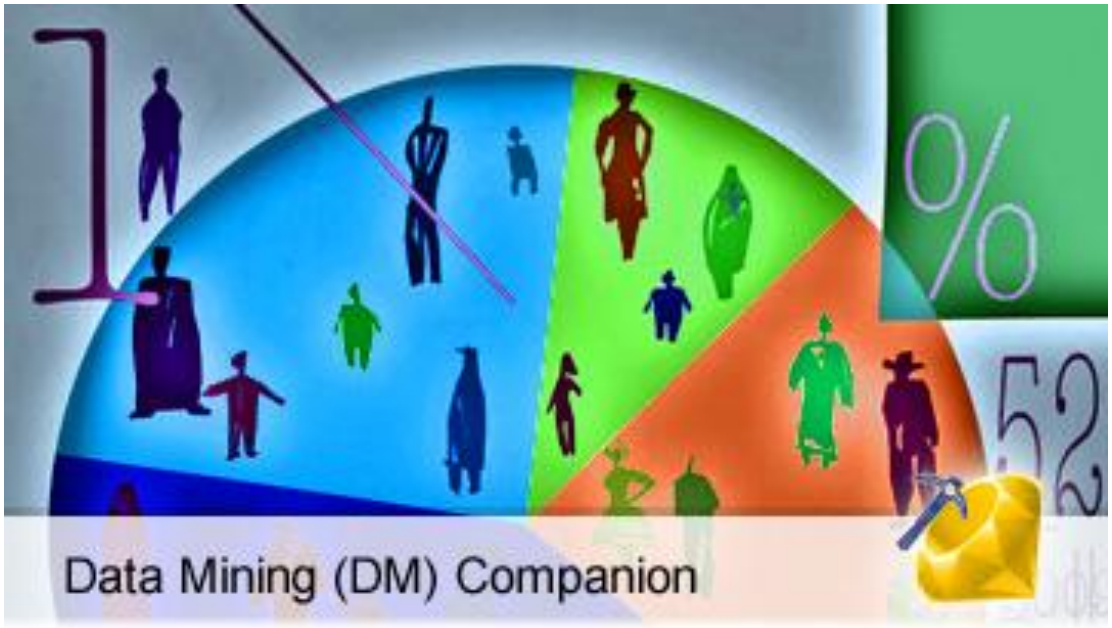
- Nếu thuộc tính có nhiều giá trị (ví dụ, các ngày trong tháng, ID3 sẽ chọn nó
- C4.5 dùng GainRatio

$$GainRatio(S,A) \equiv \frac{Gain(S,A)}{SplitInformation(S,A)}$$

$$SplitInformation(S,A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

where S_i is subset of S with A has value v_i

Phân lớp Bayes



Phân lớp Bayes

Bộ phân lớp Bayes có thể dự báo các xác suất là thành viên của lớp, chẳng hạn xác suất mẫu cho trước thuộc về một lớp xác định

Bộ phân lớp Naïve Bayes là có thể so sánh được về công năng với Bộ phân lớp với cây quyết định và mạng nơron. Chúng giả định các thuộc tính là độc lập nhau (độc lập điều kiện lớp)

Định lý Bayes

X là mẫu dữ liệu chưa biết nhãn lớp

H là giả thuyết sao cho X thuộc về lớp C

Ấn định xác suất hậu nghiệm posterior probability P(H|X) sao cho H đúng khi cho trước quan sát X (H conditioned on X)

Giả sử thế giới các mẫu dữ liệu gồm trái cây, được mô tả bằng màu sắc và hình dáng.

- Giả sử X là màu đỏ và tròn
- H là gỉa thuyết mà X là quả táo
- Thì P(H|X) phản ánh độ tin cậy X là quả táo khi biết trước X có màu đỏ và tròn

Định lý Bayes

- P(X|H) là xác suất hậu nghiệm của X có điều kiện trên. Định lý Bayes

$$P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)}$$

- Khi có n giả thuyết

$$P(H_i \mid X) = \frac{P(X \mid H_i)P(H_i)}{\sum_{j=1}^{n} P(X \mid H_j)P(H_j)}$$

Phân lớp Naïve Bayesian (NBC)

Mỗi mẫu dữ liệu được biểu diễn bằng X= (x1, x2,..., xn) với các thuộc tính A1, A2,..., An

Các lớp C1, C2, ..., Cm. Cho trước mẫu chưa biết X. NBC gán X vào Ci iff P(Ci|X) > P(Cj|X) với $1 \leq j \leq m, j \neq i$. Do vậy, chúng ta cực đại P(Ci|X). Lớp Ci sao cho P(Ci|X) là cực đại được gọi là giả thuyết hậu nghiệm cực đại (maximum posterior hypothesis). Theo định lý Bayes

$$P(C_i \mid X) = \frac{P(X \mid C_i)P(C_i)}{P(X)}$$

Phân lớp Naïve Bayesian

Do P(X) là hằng cho tất cả các lớp, chỉ cần cực đại $P(X|C_i)\ P(C_i)$. Nếu chưa biết $P(C_i)$ cần giả định $P(C_1)=P(C_2)=...= P(C_m)$ và chúng ta sẽ cực đại $P(X|C_i)$. Ngược lại, ta cực đại $P(X|C_i)\ P(C_i)$

Nếu m là lớn, sẽ rất tốn kém khi tính $P(X|C_i)\ P(C_i)$. NBC giả định độc lớp điều kiện lớp

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i)$$

Phân lớp Naïve Bayesian

Có thể phỏng tính $P(x_1|C_i)$, ..., $P(x_n|C_i)$ từ các mẫu huấn luyện

Nếu A_k được phân lớp thì $P(x_k|C_i) = s_{ik}/s_i$ với s_{ik} là số mẫu huấn luyện của C_i có trị x_k cho A_k và s_i là số các mẫu thuộc về lớp C_i

Nếu Ak là liên tục thì nó được giả định có phân bố Gaussian

$$P(x_k \mid C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}) = \frac{1}{\sqrt{2\pi\sigma_{C_i}}} e^{-\frac{(x_k - \mu_{C_i})^2}{2\sigma_{C_i}^2}}$$

Phân lớp Naïve Bayesian

Để phân lớp mẫu chưa biết X, ta tính $P(X|C_i)$ $P(C_i)$ cho từng C_i. Sau đó mẫu X được gán vào C_i iff **$P(C_i|X) > P(C_j|X)$** for $1 \leq j \leq m$, $j \neq i$

Nói cách khác, NBC gán X vào lớp C_i sao cho P(X|Ci) P(Ci) là cực đại

CSDL Customer

Table 7.1 Training data tuples from the *AllElectronics* customer database.

RID	age	income	student	credit_rating	Class: buys_computer
1	<=30	high	no	fair	no
2	<=30	high	no	excellent	no
3	31 . . . 40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	31 . . . 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	31 . . . 40	medium	no	excellent	yes
13	31 . . . 40	high	yes	fair	yes
14	>40	medium	no	excellent	no

Dự báo nhãn lớp với phân lớp Bayesian

X = (age = ‶<=30″, income = ‶fair″, student = ‶yes″, credit_rating = ‶fair″)

P(buys_computer = ‶yes″) = 9/14 = 0.643
P(buys_computer = ‶no″) = 5/14 = 0.357

To compute $P(X|C_i)\ P(C_i)$, for i = 1, 2, we compute

P(age = ‶<30″| buys_computer = ‶yes″) = 2/9 = 0.222
P(age = ‶<30″| buys_computer = ‶no″) = 3/5 = 0.600
P(income = ‶medium″| buys_computer = ‶yes″) = 4/9 = 0.444
P(income = ‶medium″| buys_computer = ‶no″) = 2/5 = 0.444
P(student = ‶yes″| buys_computer = ‶yes″) = 6/9 = 0.667
P(student = ‶yes″| buys_computer = ‶no″) = 1/5 = 0.200
P(credit_rating = ‶yes″| buys_computer = ‶yes″) = 6/9 = 0.667
P(credit_rating = ‶yes″| buys_computer = ‶no″) = 2/5 = 0.400

Dự báo nhãn lớp với phân lớp Naive Bayesian

We obtain

P(X|buys_computer = ‟yes″) = 0.222 x 0.667 x 0.667 x 0.044 = 0.044

P(X|buys_computer = ‟no″) = 0.600 x 0.400 x 0.200 x 0.400 = 0.019

P(X|buys_computer = ‟yes″)P(buys_computer = ‟yes″) =

0.044 x 0.643 = 0.028

P(X|buys_computer = ‟no″)P(buys_computer = ‟no″) =

0.019 x 0.357 = 0.007

Therefore, NBC predicts buys_computer = ‟yes″ for sample X

Các phương pháp phân lớp

k-Nearest Neighbor Classifiers

Case-based Reasoning

Genetic Algorithms

Rough Set Approach

Fuzzy Set Approaches

Rough sets: the basic idea

Each set X is represented by a pair of two sets: a lower approximation X_* and a upper approximation X^*

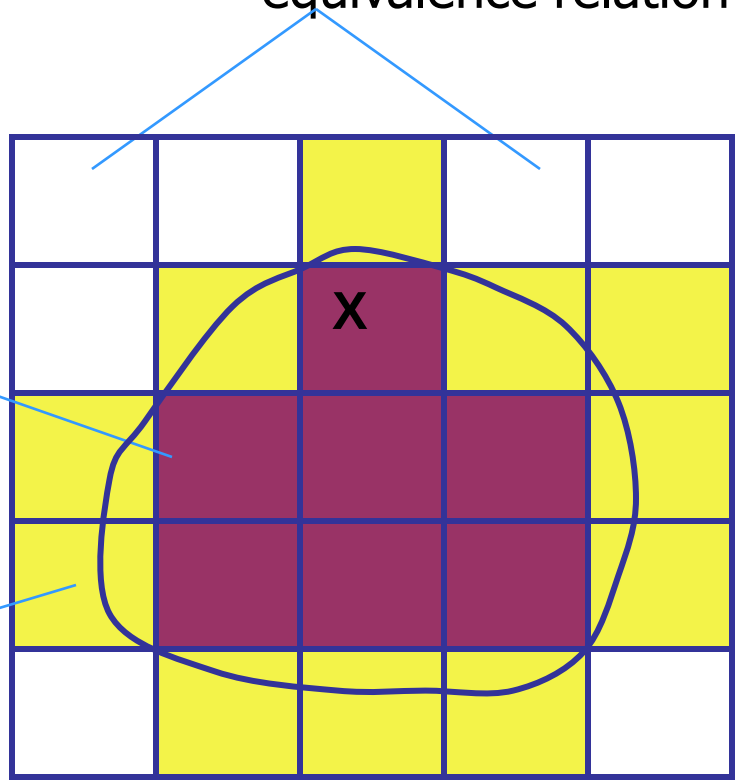
lower approximation X_* is the union of equivalence classes included in X

$$E_*(X) = \{o \in O : [o]_E \subseteq X\}$$

upper approximation X^* is the union of equivalence classes having non empty intersection with X

$$E^*(X) = \{o \in O : [o]_E \cap X \neq \emptyset\}$$

Equivalence classes $[o]_E$ defined by an equivalence relation E



Fuzzy Set Approaches

IF (year_employed >= 2) $\wedge$ (income >= $50K) THEN credit = "approved"

A customer with at least two years at job and income $49K will not satisfy the rule. With fuzzy logic we can capture the notion that an income of $49K is, to some degree, high, although not as high as an income of $50K

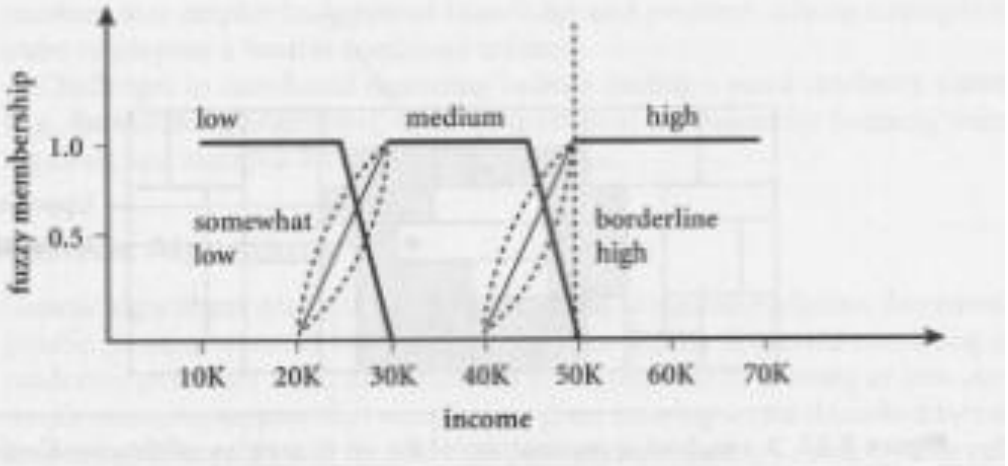


Figure 7.14 Fuzzy values for *income*.